82371FB (PIIX) AND 82371SB (PIIX3)
PCI ISA IDE XCELERATOR

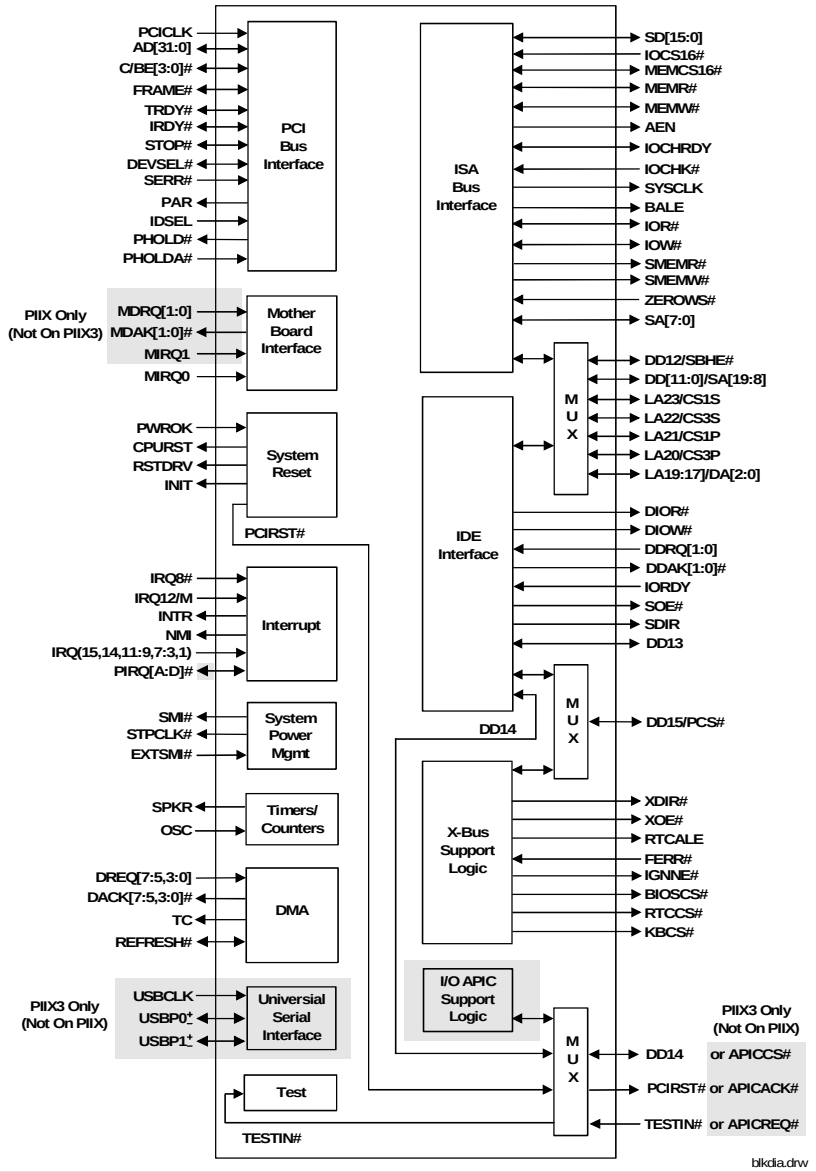
- **Bridge Between the PCI Bus and ISA Bus**
- **PCI and ISA Master/Slave Interface**
  - **PCI from 25–33 MHz**
  - **ISA from 7.5–8.33 MHz**
  - **5 ISA Slots**
- **Fast IDE Interface**
  - **Supports PIO and Bus Master IDE**
  - **Supports up to Mode 4 Timings**
  - **Transfer Rates to 22 MB/Sec**
  - **8 x 32-Bit Buffer for Bus Master IDE PCI Burst Transfers**
  - **Separate Master/Slave IDE Mode Support (PIIX3)**
- **Plug-n-Play Port for Motherboard Devices**
  - **2 Steerable DMA Channels (PIIX Only)**
  - **Fast DMA with 4-Byte Buffer (PIIX Only)**
  - **2 Steerable Interrupts Lines on the PIIX and 1 Steerable Interrupt Line on the PIIX3**
  - **1 Programmable Chip Select**
- **Steerable PCI Interrupts for PCI Device Plug-n-Play**
- **PCI Specification Revision 2.1 Compliant (PIIX3)**
- **Functionality of One 82C54 Timer**
  - **System Timer; Refresh Request; Speaker Tone Output**
- **Two 82C59 Interrupt Controller Functions**
  - **14 Interrupts Supported**
  - **Independently Programmable for Edge/Level Sensitivity**
- **Enhanced DMA Functions**
  - **Two 8237 DMA Controllers**
  - **Fast Type F DMA**
  - **Compatible DMA Transfers**
  - **7 Independently Programmable Channels**
- **X-Bus Peripheral Support**
  - **Chip Select Decode**
  - **Controls Lower X-Bus Data Byte Transceiver**
- **I/O Advanced Programmable Interrupt Controller (IOAPIC) Support (PIIX3)**
- **Universal Serial Bus (USB) Host Controller (PIIX3)**
  - **Compatible with Universal Host Controller Interface (UHCI)**
  - **Contains Root Hub with 2 USB Ports**
- **System Power Management (Intel SMM Support)**
  - **Programmable System Management Interrupt (SMI)—Hardware Events, Software Events, EXTSMI#**
  - **Programmable CPU Clock Control (STPCLK#)**
  - **Fast On/Off Mode**
- **Non-Maskable Interrupts (NMI)**
  - **PCI System Error Reporting**
- **NAND Tree for Board-Level ATE Testing**
- **208-Pin QFP**

The 82371FB (PIIX) and 82371SB (PIIX3) PCI ISA IDE Xcelerators are multi-function PCI devices implementing a PCI-to-ISA bridge function and an PCI IDE function. In addition, the PIIX3 implements a Universal Serial Bus host/hub function. As a PCI-to-ISA bridge, the PIIX/PIIX3 integrates many common I/O functions found in ISA-based PC systems—a seven-channel DMA controller, two 82C59 interrupt controllers, an 8254 timer/counter, and power management support. In addition to compatible transfers, each DMA channel supports type F transfers. Chip select decoding is provided for BIOS, real time clock, and keyboard controller. Edge/Level interrupts and interrupt steering are supported for PCI plug and play compatibility. The PIIX/PIIX3 supports two IDE connectors for up to four IDE devices providing an interface for IDE hard disks and CD ROMs. The PIIX/PIIX3 provides motherboard plug and play compatibility. PIIX implements two steerable DMA channels (including type F transfers) and up to two steerable interrupt lines. PIIX3 implements one steerable interrupt line. The interrupt lines can be routed to any of the available ISA interrupts. Both PIIX/PIIX3 implement a programmable chip select.

PIIX3 contains a Universal Serial Bus (USB) Host Controller that is UHCI compatible. The Host Controller's root hub has two programmable USB ports. PIIX3 also provides support for an external IOAPIC.

This document describes the PIIX3 Component. Unshaded areas describe the 82371FB PIIX. Shaded areas, like this one, describe the PIIX3 operations that differ from the 82371FB PIIX.

April 1997 Order Number: 290550-002



Note:
1. IOAPIC signals are multiplexed with signals from the System Reset, Test, and IDE Interface blocks
2. PIRQD# is an input on the PIIX and bi-directional on PIIX3.

PIIX/PIIX3 Simplified Block Diagram

CONTENTS

PAGE

REVISION HISTORY

Revision Date	Version	Description
May 1996	-001	Initial Release
April 1997	-002	Included information from Specification Update

1.0. SIGNAL DESCRIPTION

This section contains a detailed description of each signal. The signals are arranged in functional groups according to their interface.

Note that the '#' symbol at the end of a signal name indicates that the active, or asserted state occurs when the signal is at a low voltage level. When '#' is not present after the signal name, the signal is asserted when at the high voltage level.

The terms assertion and negation are used extensively. This is done to avoid confusion when working with a mixture of 'active-low' and 'active-high' signals. The term **assert**, or **assertion** indicates that a signal is active, independent of whether that level is represented by a high or low voltage. The term **negate**, or **negation** indicates that a signal is inactive.

Note that certain signal pins provide two separate functions. At the system level, these pins drive other signals with different functions through external buffers or transceivers. These pins have two different signal names depending on the function. These signal names have been noted in the signal description tables, with the signal whose function is being described in **bold** font. (For example, LA23/**CS1S** is in the section describing CS1S and **LA23**/CS1S is in the section describing LA23).

The following notations are used to describe the signal type.

I *Input* is a standard input-only signal.
O *Totem Pole Output* is a standard active driver.
I/O *Input/Output* is a bi-directional, tri-state signal.
od *Open Drain* allows multiple devices to share as a wire-OR.
st Schmitt Trigger input.
t/s *Tri-State* is a bi-directional, tri-state input/output pin.
s/t/s *Sustained Tri-state* is an active low tri-state signal owned and driven by one and only one agent at a time. The agent that drives a s/t/s pin low must drive it high for at least one clock before letting it float. A new agent can not start driving a s/t/s signal any sooner than one clock after the previous owner tri-states it. An external pull-up is required to sustain the inactive state until another agent drives it and must be provided by the central resource.

1.1. PCI Interface Signals

Signal Name	Type	Description
PCICLK	I	**PCI CLOCK:** PCICLK provides timing for all transactions on the PCI Bus. All other PCI signals are sampled on the rising edge of PCICLK, and all timing parameters are defined with respect to this edge. PCI frequencies of 25–33 MHz are supported.
AD[31:0]	I/O	**PCI ADDRESS/DATA:** The standard PCI address and data lines. The address is driven with FRAME# assertion and data is driven or received in following clocks
C/BE[3:0]#	I/O	**BUS COMMAND AND BYTE ENABLES:** The command is driven with FRAME# assertion. Byte enables corresponding to supplied or requested data is driven on following clocks.
FRAME#	I/O (s/t/s)	**FRAME:** Assertion indicates the address phase of a PCI transfer. Negation indicates that one more data transfer is desired by the cycle initiator.

Signal Name	Type	Description
TRDY#	I/O (s/t/s)	**TARGET READY:** Asserted when the target is ready for a data transfer.
IRDY#	I/O (s/t/s)	**INITIATOR READY:** Asserted when the initiator is ready for a data transfer.
STOP#	I/O (s/t/s)	**STOP:** Asserted by the target to request the master to stop the current transaction.
IDSEL	I	**INITIALIZATION DEVICE SELECT:** IDSEL is used as a chip select during configuration read and write transactions.
DEVSEL#	I/O (s/t/s)	**DEVICE SELECT:** The PIIX/PIIX3 asserts DEVSEL# to claim a PCI transaction through positive or subtractive decoding.
PAR	O	**CALCULATED PARITY SIGNAL:** PAR is "even" parity and is calculated on 36 bits—AD[31:0] plus C/BE[3:0]#.
SERR#	I	**SYSTEM ERROR:** SERR# can be pulsed active by any PCI device that detects a system error condition. Upon sampling SERR# active, the PIIX/PIIX3 can be programmed to generate a non-maskable interrupt (NMI) to the CPU.
PHOLD#	O	**PCI HOLD:** The PIIX/PIIX3 asserts this signal to request the PCI Bus. The PIIX3 implements the passive release mechanism by toggling PHOLD# inactive for one PCICLK.
PHLDA#	I	**PCI HOLD ACKNOWLEDGE:** This signal is asserted to grant the PCI bus to the PIIX/PIIX3.

1.2. Motherboard I/O Device Interface Signals

Signal Name	Type	Description
MDRQ[1:0] (PIIX Only)	I	**MOTHERBOARD DEVICE DMA REQUEST:** These signals can be connected internally to any of DREQ[3:0,7:5]. Each pair of request/ acknowledge signals is controlled by a separate register. Each signal can be configured as steerable interrupts for motherboard devices.
MDAK[1:0]# (PIIX Only)	O	**MOTHERBOARD DEVICE DMA ACKNOWLEDGE:** These signals can be connected internally to any of DACK[3:0,7:5]. Each pair of request/ acknowledge signals is controlled by a separate register. Each signal can be configured as steerable interrupts for motherboard devices.

Signal Name	Type	Description
MIRQ0/IRQ0 (PIIX3 Only) MIRQ[1:0] (PIIX Only)	I/O I	**MOTHERBOARD DEVICE INTERRUPT REQUEST**: The MIRQx signals can be internally connected to interrupts IRQ[15,14,12:9,7:3]. Each MIRQx line has a separate Route Control Register. If MIRQx and PIRQx# are steered to the same ISA interrupt, the device connected to the MIRQx should produce active high, level interrupts. The MIRQ0/IRQ0 signal has two functions (for PIIX3 only), depending on the programming of the IRQ0 Enable bit (MIRQ0 Register). In the systems that include the PIIX3 and IOAPIC, the MIRQ0/IRQ0 pin will function as the IRQ0 output and should be connected to the INTIN2 input of the IOAPIC. The interrupt from the Secondary IDE Channel should be connected to the IRQ15 input on PIIX3 and to the INTIN15 input on the IOAPIC. In the systems that include the PIIX3 only, the interrupt from the Secondary IDE Channel should be connected to the MIRQ0/IRQ0 input. If an MIRQ line is steered to a given IRQ input to the internal 8259, the corresponding ISA IRQ is masked, unless the Route Control register is programmed to allow the interrupts to be shared. This should only be done if the device connected to the MIRQ line and the device connected to the ISA IRQ line both produce active high, level interrupts. MIRQ0 can be configured as an output to connect the internal IRQ0 signal to an external IO-APIC.

1.3. IDE Interface Signals

Signal Name	Type	Description
DD[15:0]/ PCS#, SBHE#, SA[19:8] APICCS# (PIIX3)	**I/O** O I/O I/O O	**DISK DATA**: These signals directly drive the corresponding signals on up to two IDE connectors (primary and secondary). In addition, these signals are buffered (using 2xALS245's on the motherboard) to produce the SA[19:8], PCS#, and SBHE# signals (see separate descriptions). For the PIIX3, DD14 is buffered to produce APICCS#
DIOR#	O	**DISK I/O READ**: This signal directly drives the corresponding signal on up to two IDE connectors (primary and secondary).
DIOW#	O	**DISK I/O WRITE**: This signal directly drives the corresponding signal on up to two IDE connectors (primary and secondary).
DDRQ[1:0]	I	**DISK DMA REQUEST**: These input signals are directly driven from the DRQ signals on the primary (DDRQ0) and secondary (DDRQ1) IDE connectors. They are used in conjunction with the PCI Bus master IDE function and are not associated with any ISA-Compatible DMA channel.
DDAK[1:0]#	O	**DISK DMA ACKNOWLEDGE**: These signals directly drive the DAK# signals on the primary (DDAK0#) and secondary (DDAK1#) IDE connectors. These signals are used in conjunction with the PCI Bus master IDE function and are not associated with any ISA-Compatible DMA channel.

Signal Name	Type	Description
IORDY	I	**IO CHANNEL READY**: This input signal is directly driven by the corresponding signal on up to two IDE connectors (primary and secondary).
SOE#	O	**SYSTEM ADDRESS TRANSCEIVER OUTPUT ENABLE**: This signal controls the output enables of the ’245 transceivers that interface the DD[15:0] signals to the SA[19:8], SBHE#, PCS# and APICCS# (PIIX3 only) signals.
SDIR	O	**SYSTEM ADDRESS TRANSCEIVER DIRECTION**: This signal controls the direction of the ’245 transceivers that interface the DD[15:0] signals to the SA[19:8], SBHE#, PCIS, and APICCS# (PIIX3 only), signals. Default condition is high (transmit). When an ISA Bus master is granted use of the bus, the transceivers are turned around to drive the ISA address [19:8] on DD[15:3]. The address can then be latched by the PIIX/PIIX3. In this case, the SDIR signal is low (receive). The SOE# and SDIR signals taken together as a group can assume one of three states: **SOE# SDIR State** 0 1 PCI to ISA transaction 1 1 PCI to IDE 0 0 ISA Bus master

Signals Buffered from LA[23:17]

These signals are buffered from the LA[23:17] lines by an ALS244 tri-state buffer. The output enable of this buffer is tied asserted. These signals are set up with respect to the IDE command strobes (DIOR# and IOW#) and are valid throughout I/O transactions targeting the ATA register block(s).

Signal Name	Type	Description
LA23/ **CS1S**	I/O	**CHIP SELECT:** CS1S is for the ATA command register block and corresponds to the inverted CS1FX# on the secondary IDE connector. CS1S is inverted externally (see PCI Local Bus IDE section).
LA22/ **CS3S**	I/O	**CHIP SELECT:** CS3S is for the ATA control register block and corresponds to the inverted CS3FX# on the secondary IDE connector. CS3S is inverted externally (see PCI Local Bus IDE section).
LA21/ **CS1P**	I/O	**CHIP SELECT:** CS1P is for the ATA command register block and corresponds to the inverted CS1FX# on the primary IDE connector. CS1P is inverted externally (see PCI Local Bus IDE section).
LA20/ **CS3P**	I/O	**CHIP SELECT:** CS3P is for the ATA control register block and corresponds to the inverted CS3FX# on the primary IDE connector. CS3P is inverted externally (see PCI Local Bus IDE section).
LA[19:17] **DA[2:0]**	I/O	**DISK ADDRESS:** DA[2:0] are used to indicate which byte in either the ATA command block or control block is being addressed.

1.4. ISA Interface Signals

Signal Name	Type	Description
BALE	O	**BUS ADDRESS LATCH ENABLE:** BALE is an active high signal asserted by the PIIX/PIIX3 to indicate that the address (SA[19:0], LA[23:17]) and SBHE# signal lines are valid.
AEN	O	**ADDRESS ENABLE:** AEN is asserted during DMA cycles to prevent I/O slaves from misinterpreting DMA cycles as valid I/O cycles. This signal is also driven high during PIIX/PIIX3 initiated refresh cycles. For the PIIX, when TC is sampled low on the assertion of PWORK (External DMA mode), the PIIX tri-states this signal.
SYSCLK	O	**ISA SYSTEM CLOCK:** SYSCLK is the reference clock for the ISA Bus and drives the bus directly. SYSCLK is generated by dividing PCICLK by 3 or 4. The SYSCLK frequencies supported are 7.5 MHz and 8.33 MHz. SYSCLK is a divided down version of PCICLK. **Hardware Strapping Option** SYSCLK is tri-stated when PWROK is negated. The value of SYSCLK is sampled on the assertion of PWROK: If sampled high, the ISA clock divisor is 3 (for 25 MHz PCI). If sampled low, the divisor is 4 (for 30 and 33 MHz PCI).
IOCHRDY	I/O	**I/O CHANNEL READY:** Resources on the ISA Bus negate IOCHRDY to indicate that additional time (wait states) is required to complete the cycle. This signal is normally high on the ISA Bus. IOCHRDY is an input when the PIIX/PIIX3 owns the ISA Bus and the CPU or a PCI agent is accessing an ISA slave or during DMA transfers. IOCHRDY is output when an external ISA Bus Master owns the ISA Bus and is accessing DRAM or a PIIX/PIIX3 register.
IOCS16#	I	**16-BIT I/O CHIP SELECT:** This signal is driven by I/O devices on the ISA Bus to indicate that they support 16-bit I/O bus cycles.
IOCHK#	I	**I/O CHANNEL CHECK:** IOCHK# can be driven by any resource on the ISA Bus. When asserted, it indicates that a parity or an uncorrectable error has occurred for a device or memory on the ISA Bus. If enabled, a NMI is generated to the CPU.
IOR#	I/O	**I/O READ:** IOR# is the command to an ISA I/O slave device that the slave may drive data on to the ISA data bus (SD[15:0]).
IOW#	I/O	**I/O WRITE:** IOW# is the command to an ISA I/O slave device that the slave may latch data from the ISA data bus (SD[15:0]).
LA[23:17]/ CS1S CS3S CS1P CS3P DA[2:0]	**I/O**/ O O O O O	**UNLATCHED ADDRESS:** The LA[23:17] address lines are bi-directional. These address lines allow accesses to physical memory on the ISA Bus up to 16 Mbytes. The LA[23:17] are also used to drive the IDE interface chip selects and address lines via an external ALS244 buffer. See the IDE Interface signal descriptions.

Signal Name	Type	Description
SA[7:0], **SA[19:8]/** DD[11:0]	I/O **I/O** I/O	**SYSTEM ADDRESS BUS:** These bi-directional address lines define the selection with the granularity of one byte within the one-Mbyte section of memory defined by the LA[23:17] address lines. The address lines SA[19:17] that are coincident with LA[19:17] are defined to have the same values as LA[19:17] for all memory cycles. For I/O accesses, only SA[15:0] are used.
SBHE#/ DD12	**I/O** I/O	**SYSTEM BYTE HIGH ENABLE:** SBHE# indicates, when asserted, that a byte is being transferred on the upper byte (SD[15:8]) of the data bus. SBHE# is negated during refresh cycles.
MEMCS16#	od	**MEMORY CHIP SELECT 16:** MEMCS16# is a decode of LA[23:17] without any qualification of the command signal lines. ISA slaves that are 16-bit memory devices drive this signal low. The PIIX/PIIX3 drives this signal low during ISA master to DRAM Cycles.
MEMR#	I/O	**MEMORY READ:** MEMR# is the command to a memory slave that it may drive data onto the ISA data bus. This signal is also driven by the PIIX/PIIX3 during refresh cycles.
MEMW#	I/O	**MEMORY WRITE:** MEMW# is the command to a memory slave that it may latch data from the ISA data bus.
SMEMR#	O	**STANDARD MEMORY READ:** The PIIX/PIIX3 asserts SMEMR# to request an ISA memory slave to drive data onto the data lines. If the access is below 1 Mbyte (00000000–000FFFFFh) during DMA compatible, PIIX/PIIX3 master, or ISA master cycles, the PIIX/PIIX3 asserts SMEMR#. SMEMR# is a delayed version of MEMR#.
SMEMW#	O	**STANDARD MEMORY WRITE:** The PIIX/PIIX3 asserts SMEMW# to request an ISA memory slave to accept data from the data lines. If the access is below 1 Mbyte (00000000–000FFFFFh) during DMA compatible, PIIX/PIIX3 master, or ISA master cycles, the PIIX/PIIX3 asserts SMEMW#. SMEMW# is a delayed version of MEMW#.
ZEROWS#	I	**ZERO WAIT-STATES:** An ISA slave asserts ZEROWS# after its address and command signals have been decoded to indicate that the current cycle can be shortened. A 16-bit ISA memory cycle can be reduced to two SYSCLKs. An 8-bit memory or I/O cycle can be reduced to three SYSCLKs. ZEROWS# has no effect during 16-bit I/O cycles.
SD[15:0]	I/O	**SYSTEM DATA:** SD[15:0] provide the 16-bit data path for devices residing on the ISA Bus. SD[15:8] correspond to the high order byte and SD[7:0] correspond to the low order byte. SD[15:0] are undefined during refresh.

1.5. DMA Signals

Signal Name	Type	Description
DREQ [7:5,3:0]	I	**DMA REQUEST:** The DREQ lines are used to request DMA service from the PIIX/PIIX3 's DMA controller or for a 16-bit master to gain control of the ISA expansion bus. The active level (high or low) is programmed via the DMA Command Register. The request must remain active until the appropriate DACKx# signal is asserted.
DACK [7:5,3:0]#	O	**DMA ACKNOWLEDGE:** The DACK output lines indicate that a request for DMA service has been granted by the PIIX/PIIX3 or that a 16-bit master has been granted the bus. The active level (high or low) is programmed via the DMA Command Register. These lines should be used to decode the DMA slave device with the IOR# or IOW# line to indicate selection. If used to signal acceptance of a bus master request, this signal indicates when it is legal to assert MASTER#. For the PIIX, when TC is sampled low on the assertion of PWORK (External DMA mode), the PIIX tri-states these signals. This mode is not available on PIIX3.
TC	O	**TERMINAL COUNT:** The PIIX/PIIX3 asserts TC to DMA slaves as a terminal count indicator. When all the DMA channels are not in use, TC is negated (low). **Hardware Strapping Option (PIIX Only)** This strapping option selects between the internal ISA DMA mode and External DMA mode. When TC is sampled high on the assertion of PWROK (ISA DMA mode), the PIIX drives the AEN, TC, and DACK#[7:5, 3:0] normally. When TC is sampled low on the assertion of PWROK (External DMA mode), the PIIX tri-states the AEN, TC, and DACK[7:5, 3:0]# signals, and also forwards PCI masters I/O accesses to location 0000h to ISA. TC has an internal pull-up resistor. To tie TC low, an external 1 KΩ pull-down resistor should be used. For the PIIX3, this signal should not be pulled down.
REFRESH#	I/O	**REFRESH:** As an output, REFRESH# indicates when a refresh cycle is in progress. It should be used to enable the SA[7:0] address to the row address inputs of all banks of dynamic memory on the ISA Bus. Thus, when MEMR# is asserted, the entire expansion bus dynamic memory is refreshed. Memory slaves must not drive any data onto the bus during refresh. As an output, this signal is driven directly onto the ISA Bus. This signal is an output only when the PIIX/PIIX3 DMA refresh controller is a master on the bus responding to an internally generated request for refresh. As an input, REFRESH# is driven by 16-bit ISA Bus masters to initiate refresh cycles.

1.6. Timer/Counter Signals

Signal Name	Type	Description
SPKR	O	**SPEAKER DRIVE:** The SPKR signal is the output of counter 2.
OSC	I	**OSCILLATOR:** OSC is the 14.31818 MHz ISA clock signal. It is used by the internal 8254 Timer.

1.7. Interrupt Controller Signals

Signal Name	Type	Description
IRQ[15,14, 11:9, 7:3,1]	I	**INTERRUPT REQUEST:** The IRQ signals provide both system board components and ISA Bus I/O devices with a mechanism for asynchronously interrupting the CPU. The assertion mode of these inputs depends on the programming of the two ELCR registers. The IRQ14 signal must be used by the Bus Master IDE interface function to signal interrupts on the primary IDE channel.
IRQ8#	I	**INTERRUPT REQUEST EIGHT SIGNAL:** IRQ8# is always an active low edge triggered interrupt input (i.e., this interrupt can not be modified by software). Upon PCIRST#, IRQ8# is placed in active low edge sensitive mode.
IRQ12/M	I	**INTERRUPT REQUEST/MOUSE INTERRUPT:** In addition to providing the standard interrupt function (see IRQ[15,14,11:9,7:3,1] signal description), this pin can be programmed (via X-Bus Chip Select Register) to provide a mouse interrupt function.
PIRQ[D:A]#	I I/O For PIRQD# (PIIX3 only)	**PROGRAMMABLE INTERRUPT REQUEST:** The PIRQx# signals can be shared with interrupts IRQ[15,14,12:9,7:3] as described in the Interrupt Steering section. Each PIRQx# line has a separate Route Control Register. These signals require external pull-up resisters. For the PIIX3, the USB interrupt is output on PIRQD#.
INTR	od	**CPU INTERRUPT:** INTR is driven by the PIIX/PIIX3 to signal the CPU that an interrupt request is pending and needs to be serviced. The interrupt controller must be programmed following PCIRST# to ensure that INTR is at a known state.
NMI	od	**NON-MASKABLE INTERRUPT:** NMI is used to force a non-maskable interrupt to the CPU. The PIIX/PIIX3 generates an NMI when either SERR# or IOCHK# is asserted, depending on how the NMI Status and Control Register is programmed.

1.8. System Power Management (SMM) Signals

Signal Name	Type	Description
SMI#	od	**SYSTEM MANAGEMENT INTERRUPT:** SMI# is an active low synchronous output that is asserted by the PIIX/PIIX3 in response to one of many enabled hardware or software events.
STPCLK#	od	**STOP CLOCK:** STPCLK# is an active low synchronous output that is asserted by the PIIX/PIIX3 in response to one of many hardware or software events. STPCLK# connects directly to the CPU and is synchronous to PCICLK.
EXTSMI#	I	**EXTERNAL SYSTEM MANAGEMENT INTERRUPT:** EXTSMI# is a falling edge triggered input to the PIIX/PIIX3 indicating that an external device is requesting the system to enter SMM mode. This signal contains a weak internal pullup.

1.9. X-Bus Signals

Signal Name	Type	Description
XDIR#	O	**X-BUS DIRECTION:** XDIR# is tied directly to the direction control of a 74F245 that buffers the X-Bus data (XD[7:0]). XDIR# is asserted for all I/O read cycles, regardless if the accesses are to a PIIX/PIIX3 supported device. XDIR# is only asserted for memory cycles if BIOS sapce (PIIX and PIIX3) or APIC space (PIIX3 only) has been decoded. For PCI master inititated read cycles, XDIR# is asserted from the falling edge of either IOR# or MEMR# (from MEMR# only if BIOS space (PIIX and PIIX3) or APIC (PIIX3 only) space has been decoded), depending on the cycle type. For ISA master-initiated read cycles, XDIR# is asserted from the falling edge of either IOR# or MEMR# (from MEMR# only if BIOS space has been decoded), depending on the cycle type. When the rising edge of IOR# or MEMR# occurs, the PIIX/PIIX3 negates XDIR#. For DMA read cycles from the X-Bus, XDIR# is asserted from DACKx# falling and negated from DACKx# rising. At all other times, XDIR# is negated.
XOE#	O	**X-BUS OUTPUT ENABLE:** XOE# is tied directly to the output enable of a 74F245 that buffers the X-Bus data (XD[7:0]) from the system data bus (SD[7:0]). XOE# is asserted when a PIIX/PIIX3 supported X-Bus device is decoded, and the devices decode is enabled in the X-Bus Chip Select Enable Register (XBCS Register). XOE# is asserted from the falling edge of the ISA commands (IOR#, IOW#, MEMR#, or MEMW#) for PCI Master and ISA master-initiated cycles. XOE# is negated from the rising edge of the ISA command signals for CPU and PCI Master-initiated cycles and the SA[16:0] and LA[23:17] address for ISA master-initiated cycles. XOE# is not generated during any access to an X-Bus peripheral in which its decode space has been disabled.
DD15/ **PCS#**	O	**PROGRAMMABLE CHIP SELECT:** PCS# is asserted for ISA I/O cycles that are generated by PCI masters and subtractively decoded to ISA, if the access hits the address range programmed into the PCSC Register. The X-Bus buffer signals are enabled when the chip select is asserted (i.e., it is assumed that the peripheral that is selected via this pin resides on the X-Bus).
BIOSCS#	O	**BIOS CHIP SELECT:** BIOSCS# is asserted during read or write accesses to BIOS. BIOSCS# is driven combinatorially from the ISA addresses SA[16:0] and LA [23:17], except during DMA. During DMA cycles, BIOSCS# is not generated.
KBCS#	O	**KEYBOARD CONTROLLER CHIP SELECT:** KBCS# is asserted during I/O read or write accesses to KBC locations 60h and 64h. This signal is driven combinatorially from the ISA addresses SA[16:0] and LA [23:17]. For DMA cycles, KBCS# is never asserted.
RTCCS#	O	**REAL TIME CLOCK CHIP SELECT:** RTCCS# is asserted during read or write accesses to RTC location 71h. RTCCS# can be tied to a pair of external OR gates to generate the real time clock read and write command signals.
RTCALE	O	**REAL TIME CLOCK ADDRESS LATCH:** RTCALE is used to latch the appropriate memory address into the RTC. A write to port 70h with the appropriate RTC memory address that will be written to or read from, causes RTCALE to be asserted. RTCALE is asserted based on IOW# falling and remains asserted for two SYSCLKs.

Signal Name	Type	Description
FERR#	I	**NUMERIC COPROCESSOR ERROR:** This signal is tied to the coprocessor error signal on the CPU. IGNNE# is only used if the PIIX/PIIX3 coprocessor error reporting function is enabled in the XBCSA Register. If FERR# is asserted, the PIIX/PIIX3 generates an internal IRQ13 to its interrupt controller unit. The PIIX/PIIX3 then asserts the INTR output to the CPU. FERR# is also used to gate the IGNNE# signal to ensure that IGNNE# is not asserted to the CPU unless FERR# is active. FERR# has a weak internal pull-up used to ensure a high level when the coprocessor error function is disabled.
IGNNE#	od	**IGNORE ERROR:** This signal is connected to the ignore error pin on the CPU. IGNNE# is only used if the PIIX/PIIX3 coprocessor error reporting function is enabled in the XBCSA Register. If FERR# is asserted, indicating a coprocessor error, a write to the Coprocessor Error Register (F0h) causes the IGNNE# to be asserted. IGNNE# remains asserted until FERR# is negated. If FERR# is not asserted when the Coprocessor Error Register is written, the IGNNE# signal is not asserted.

1.10. APIC Bus Signals (PIIX3 Only)

Signal Name	Type	Description
DD14/ **APICCS#**	I/O O	**APIC CHIP SELECT (PIIX3 only).** This active low output signal is asserted when the APIC Chip Select is enabled and a PCI originated cycle is positively decoded within the programmed IOAPIC address space. The default addresses of the IOAPIC are Memory FEC0_0000h and FEC0_0010h. System Design Note: The DD[14]/APICCS# signal is demuxed externally with a 245 transceiver. The output of the transceiver drives the IOAPIC's CS# signal. At certain times the transceiver floats its outputs, therefore a pullup resistor on the output of the tranceiver is required to keep this signal negated.
TESTIN#/ **APICREQ#**	I	**APIC REQUEST (PIIX3 only).** This signal has two functions, depending on the programming of the APIC Chip Select bit (XBCS Register). See the Test SIgnal Description for the TESTIN# function. APICREQ# is asserted by an external APIC device prior to sending an interrupt over the APIC serial bus. When the PIIX3 samples this pin active it flushes its F-type DMA buffers pointing towards PCI. Once the buffers are flushed, the PIIX3 asserts APICACK# to inform the external APIC that it can proceed to send the APIC interrupt. APICREQ# must be synchronous to PCICLK.
PCIRST#/ **APICACK#**	O O	**APIC ACKNOWLEDGE (PIIX3 only)**. This signal has two functions, depending on the programming of the APIC Chip Select bit (XBCS Register). See the System Reset Signal Description for the PCIRST# function. The PIIX3 asserts APICACK# after its internal buffers are flushed in response to the APICREQ# signal. When the IOAPIC samples this signal asserted it knows that the PIIX3's buffers are flushed and that it can proceed to send the APIC interrupt. The signal is driven from the rising edge of PCICLK and is negated while PCIRST# is asserted.

1.11. Universal Serial Bus Signals (PIIX3 Only)

Signal Name	Type	Description
USBCLK	I	**UNIVERSAL SERIAL BUS CLOCK.** This signal clocks the universial serial bus clock.
USBP0+ USBP0-	I/O	**UNIVERSAL SERIAL BUS PORT 0.** These signals are the differential data pair for Serial Port 0.
USBP1+ USBP1-	I/O	**UNIVERSIAL SERIAL BUS PORT 1.** These signals are the differential data pair for Serial Port 1.

1.12. System Reset Signals

Signal Name	Type	Description
PWROK	I	**POWER OK:** When asserted, PWROK is an indication to the PIIX/PIIX3 that power and PCICLK have been stable for at least 1 ms. PWROK can be driven asynchronously. When PWROK is negated, the PIIX/PIIX3 asserts CPURST, PCIRST# and RSTDRV. When PWROK is asserted, the PIIX/PIIX3 negates CPURST, PCIRST#, and RSTDRV.
CPURST	od	**CPU RESET:** The PIIX/PIIX3 asserts CPURST to reset the CPU. The PIIX/PIIX3 asserts CPURST during power-up and when a hard reset sequence is initiated through the RC register. CPURST is driven synchronously to the rising edge of PCICLK. If a hard reset is initiated through the RC register, the PIIX/PIIX3 resets it's internal registers to the default state.
PCIRST#/ APICACK# (PIIX3 Only)	O O	**PCI RESET:** This signal has two functions, depending on the programming of the APIC Chip Select bit (XBCS Register). See the APIC SIgnal Description for the APICACK# function. The PIIX/PIIX3 asserts PCIRST# to reset devices that reside on the PCI Bus. The PIIX/PIIX3 asserts PCIRST# during power-up and when a hard reset sequence is initiated through the RC register. PCIRST# is driven inactive a minimum of 1 ms after PWROK is driven active. PCIRST# is driven active for a minimum of 1ms when initiated through the RC register. PCIRST# is driven asynchronously relative to PCICLK.
INIT	OD	**INITIALIZATION:** The PIIX/PIIX3 asserts INIT if it detects a shut down special cycle on the PCI Bus or if a soft reset is initiated via the RC Register.
RSTDRV	O	**RESET DRIVE:** The PIIX/PIIX3 asserts this signal during a hard reset and during power-up to reset ISA Bus devices. RSTDRV is also asserted for a minimum of 1 ms if a hard reset has been programmed in the RC Register.

1.13. Test Signals

Signal Name	Type	Description
TESTIN#/ APICREQ# (PIIX3 Only)	I I	**TEST INPUT:** This signal has two functions, depending on the programming of the APIC Chip Select bit (XBCS Register). See the APIC SIgnal Description for the APICREQ# function. The TESTIN# signal is used in conjunction with the IRQ signals to select the various test modes of the PIIX/PIIX3. This input contains an internal pull up resistor. After a hard reset, this pin functions as a TESTIN# signal. An external weak pull-up resistor (4.7k to 20k ohms) is required to 5V.

1.14. Power and Ground Signals

Signal Name	Type	Description
VCC		**Power (5 Volts):** This pin is connected to the 5 volt power supply.
VCC3 (PIIX3)		**Power (3.3 Volts):** This pin is connected to the 3.3 volt power supply. Note that, if the the Universal Serial Bus function is not used, this pin can be connected to the 5 volt power supply.
GND		**Ground:** This pin is connected to the ground plane.

1.15. Signal State During Reset

Table 1 shows the state of all PIIX/PIIX3 output and bi-directional signals during a hard reset. A hard reset is initiated when PWROK is asserted or by programming a hard reset through the RC Register.

Table 1. Output and I/O Signal States During Hard Reset

Signal	State
AD[31:0]	Low (PIIX)
	Tri-State (PIIX3)
C/BE[3:0]#	Low (PIIX)
	Tri-State (PIIX3)
FRAME#	Tri-state
TRDY#	Tri-state
IRDY#	Tri-state
STOP#	Tri-state
DEVSEL#	Tri-state
PAR	Input
PHOLD#	High (PIIX)
	Tri-state (PIIX3)
MDAK[1:0]# (PIIX)	High
DD[15,13:0]/ PCS#,SBHE#, SA[19:8], DD14 (PIIX)	Tri-state
DD14 (PIIX)	Tri-state
DD14/APICCS# (PIIX3)	High
SA[7:0]	Undefined
DIOR#	High
DIOW#	High
DDAK[1:0]#	High
SOE#	High
SDIR	High

Signal	State
LA23/CS1S	Undefined
LA22/CS3S	Undefined
LA21/CS1P	Undefined
LA20/CS3P	Undefined
LA[19:17]/ DA[2:0]	Undefined
BALE	Low
AEN	Depends on strapping option
SYSCLK	Strapping Option
IOCHRDY	Tri-state
IOR#	High
IOW#	High
MEMCS16#	Open drain
MEMR#	Tri-state
MEMW#	Tri-state
SMEMR#	High
SMEMW#	High
SD[15:0]	Tri-state
DACK[7:5,3:0]#	Depends on strapping option
TC	Strapping Option
REFRESH#	Tri-state
SPKR	Low
INTR	Open drain

Signal	State
PIRQD# (PIIX3)	Tri-state
NMI	Open drain
SMI#	Open drain
STPCLK#	Open drain
XDIR#	High
XOE#	High
BIOSCS#	Undefined (PIIX)
	High (PIIX3)
KBCS#	Undefined (PIIX)
	High (PIIX3)
RTCCS#	Undefined (PIIX)
	High (PIIX3)
RTCALE	Low
FERR#	Open drain
IGNNE#	Open drain
CPURST	Open drain
USBP0+ USBP0-	Tri-state
USBP1+ USBP1-	Tri-state
PCIRST# (PIIX)	Low
PCIRST#/ APICACK# (PIIX3)	
INIT	Low
RSTDRV	High

2.0. REGISTER DESCRIPTION

The 82371FB PIIX internal registers are organized into five groups—PCI Configuration Registers (function 0), PCI Configuration Registers (function 1), ISA-Compatible Registers, PCI Bus Master IDE Registers, and System Power Management Registers. These registers are discussed in this section.

The PIIX3 internal registers contain the same register sets as the PIIX plus two additional register sets for the Universal Serial Bus (USB) function—PCI Configuration Registers (function 2) and the USB I/O Registers.

Some of the PIIX/PIIX3 registers contain reserved bits. Software must deal correctly with fields that are reserved. On reads, software must use appropriate masks to extract the defined bits and not rely on reserved bits being any particular value. On writes, software must ensure that the values of reserved bit positions are preserved. That is, the values of reserved bit positions must first be read, merged with the new values for other bit positions and then written back.

In addition to reserved bits within a register, the PIIX/PIIX3 contains address locations in the PCI configuration space that are marked "Reserved". The PIIX/PIIX3 responds to accesses to these address locations by completing the host cycle. Software should not write to reserved PIIX/PIIX3 configuration locations in the device-specific region (above address offset 3Fh).

During a hard reset, the PIIX/PIIX3 sets its internal registers to predetermined **default** states. The default values are indicated in the individual register descriptions.

The following notation is used to describe register access attributes:

RO **Read Only**. If a register is read only, writes have no effect.
WO **Write Only**. If a register is write only, reads have no effect.
R/W **Read/Write**. A register with this attribute can be read and written. Note that individual bits in some read/write registers may be read only.
R/WC **Read/Write Clear**. A register bit with this attribute can be read and written. However, a write of a 1 clears (sets to 0) the corresponding bit and a write of a 0 has no effect.

2.1. Register Access

Table 2, Table 3, and Table 4 show the I/O assignments for the PCI Configuration Registers (function 0, 1, and 2). Table 5 shows the I/O assignments for the ISA Compatible Registers. Table 6 shows the I/O assignments for the Bus Master IDE Interface registers. Table 7 shows the I/O assignments for the USB I/O registers. PCI masters have access to all PIIX/PIIX3 internal registers. In addition, ISA masters have access to some of the ISA-Compatible registers (see Table 5).

PCI Configuration Registers (functions 0, 1, and 2)

The 82371FB PIIX is a multi-function device on the PCI Bus implementing two functions—PCI-to-ISA Bridge (function 0) and IDE Interface (function 1). These functions can be independently configured with two sets of PCI configuration registers in compliance with the PCI Local Bus Specification, Revision 2.0. The two sets of configuration registers are accessed by the CPU through a mechanism defined for multi-functional PCI devices. The PIIX does not assert DEVSEL# for PCI configuration cycles that target functions 2 through 7.

The PIIX3 implements an additional PCI bus function—Universial Serial Bus Interface. The PIIX3 does not assert DEVSEL# for PCI configuration cycles that target functions 3 through 7.

ISA Compatible Registers

The ISA-Compatible registers (e.g., DMA registers, timer/counter registers, X-Bus registers, and NMI registers) are accessed through normal I/O space. Except for the DMA registers, the PIIX/PIIX3 positively decodes accesses to the ISA-Compatible registers. The PIIX/PIIX3 subtractively decodes accesses to all I/O space registers contained within the ISA-Compatible DMA function. This permits another device in the system to implement the compatible DMA function.

PCI master accesses to the ISA-Compatible registers can be 8, 16, 24, or 32 bits. However, the PIIX/PIIX3 only responds to the least significant byte. On writes the other bytes are not loaded and on reads the other bytes have invalid data. The PIIX/PIIX3 responds as an 8-bit ISA I/O slave when accessed by an ISA master. See the PCI Local Bus IDE section for accesses to the IDE register blocks located in the IDE device

In general, accesses from CPU or PCI masters to the internal PIIX/PIIX3 registers are not broadcast to the ISA Bus. Exceptions to this are read/write accesses to 70h and F0h and write accesses to 80h, 84–86h, 88h, 8C–8Eh, 90h, 94–96h, 98h, and 9C–9Eh. These accesses are broadcast to the ISA Bus. Note that aliasing of the 90–9Fh to 80–8Fh can be enabled/disabled via the ISA Controller Recovery Timer Register.

Power Management Registers

There are two power management registers located in normal I/O space. These registers are accessed (by PCI Bus masters) with 8-bit accesses. The other power management registers are located in PCI configuration space for function 0.

PCI Bus Master IDE Registers

The PCI Bus master IDE function uses 16 bytes located in normal I/O space, allocated via the BMIBA register (a PCI base address register). All bus master IDE I/O space registers can be accessed as 8, 16, or 32-bit quantities.

Universal Serial Bus Registers

A set of USB registers provide control and status information for the Host Controller and two USB ports (Table 7). This block of registers is I/O mapped into PCI I/O space and control variouos operations of the USB Host Controller. The registers are relocateable via the USBBASE Register which is located in function 0 PCI configuration space.

Table 2. PCI Configuration Registers—Function 0 (PCI to ISA Bridge)

Configuration Offset	Mnemonic	Register	Register Access
00–01h	VID	Vendor Identification	RO
02–03h	DID	Device Identification	RO
04–05h	PCICMD	PCI Command	R/W
06–07h	PCISTS	PCI Device Status	R/WC
08h	RID	Revision Identification	RO
09–0Bh	CLASSC	Class Code	RO
0C–0Dh	—	Reserved	—
0Eh	HEDT	Header Type	RO

Configuration Offset	Mnemonic	Register	Register Access
0F–4Bh	—	Reserved	—
4Ch	IORT	ISA I/O Controller Recovery Timer	R/W
4Dh	—	Reserved	—
4Eh (PIIX) 4E–4Fh (PIIX3)	XBCS	X-Bus Chip Select Enable	R/W
4F–5Fh (PIIX) 50–5Fh (PIIX3)	—	Reserved	—
60–63h	PIRQRC[A:D]	PCI IRQ Route Control	R/W
64–68h	—	Reserved	—
69h	TOM	Top of Memory	R/W
6A–6Bh	MSTAT	Miscellaneous Status	R/W
6C–6Fh	—	Reserved	—
70h	MBIRQ0	Motherboard IRQ Route Control 0	R/W
71h	MBIRQ1	Motherboard IRQ Route Control 1 (PIIX)	R/W
	—	Reserved (PIIX3)	—
72–75h	—	Reserved	—
76–77h	MBDMA[1:0]	Motherboard DMA Control	R/W
78–79h	PCSC	Programmable Chip Select Control	R/W
7A–7Fh	—	Reserved	—
80h	—	Reserved (PIIX)	—
	APICBASE	APIC Base Address Relocation (PIIX3)	R/W
81h	—	Reserved	—
82h	—	Reserved (PIIX)	—
	DLC	Deterministic Latency Control (PIIX3)	R/W
83–9Fh	—	Reserved	—
A0h	SMICNTL	SMI Control	R/W
A1h	—	Reserved	—
A2–A3h	SMIEN	SMI Enable	R/W
A4–A7h	SEE	System Event Enable	R/W
A8h	FTMR	Fast Off Timer	R/W
A9h	—	Reserved	—
AA–ABh	SMIREQ	SMI Request	R/W

Configuration Offset	Mnemonic	Register	Register Access
ACh	CTLTMR	Clock Scale STPCLK# Low Timer	R/W
ADh	—	Reserved	—
AEh	CTHTMR	Clock Scale STPCLK# High Timer	R/W
AF–FFh	—	Reserved	—

Table 3. PCI Configuration Registers—Function 1 (IDE Interface)

Configuration Offset	Mnemonic	Register	Register Access
00–01h	VID	Vendor Identification	RO
02–03h	DID	Device Identification	RO
04–05h	PCICMD	Command	R/W
06–07h	PCISTS	PCI Device Status	R/WC
08h	RID	Revision Identification	RO
09–0Bh	CLASSC	Class Code	RO
0Ch	—	Reserved	—
0Dh	MLT	Master Latency Timer	R/W
0Eh	HEDT	Header Type	RO
0F–1Fh	—	Reserved	—
20–23h	BMIBA	Bus Master Interface Base Address	R/W
24–3Fh	—	Reserved	—
40–43h	IDETIM	IDE Timing Modes	R/W
44h	—	Reserved (PIIX)	—
	SIDETIM	Slave IDE Timing Modes (PIIX3)	R/W
45–FFh	—	Reserved	—

Table 4. PCI Configuration Registers—Function 2 (Universial Serial Bus Interface) (PIIX3 Only)

Configuration Offset	Mnemonic	Register	Register Access
00–01h	VID	Vendor Identification	RO
02–03h	DID	Device Identification	RO
04–05h	PCICMD	Command	R/W
06–07h	PCISTS	Device Status	R/WC
08h	RID	Revision Identification	RO
09–0Bh	CLASSC	Class Code	RO
0Ch	—	Reserved	—
0Dh	LATTMR	Latency Timer	R/W
0Eh	HEDT	Header Type	RO
0F–19h	—	Reserved	—
20–23h	BASEADD	IO Space Base Address	R/W
24–3Bh	—	Reserved	—
3Ch	IL	Interrupt Line	R/W
3Dh	INTRP	Interrupt Pin	RO
3E–5Fh	—	Reserved.	—
60h	SBRNUM	Serial Bus Release Number	RO
61–69h	—	Reserved	—
6A–6Bh	MSTAT	Miscellaneous Status	R/W
6C–BFh	—	Reserved	—
C0–C1h	LEGSUP	Legacy Support	R/WC
C2–FFh	—	Reserved	—

Table 5. ISA-Compatible and Power Management Registers

Address	Address (bits)				Type	Name	Access
(hex)	FEDC	BA98	7654	3210			
0000h[3]	0000	0000	000x	0000	r/w	DMA1 CH0 Base and Current Address	PCI
0001h[3]	0000	0000	000x	0001	r/w	DMA1 CH0 Base and Current Count	PCI
0002h[3]	0000	0000	000x	0010	r/w	DMA1 CH1 Base and Current Address	PCI
0003h[3]	0000	0000	000x	0011	r/w	DMA1 CH1 Base and Current Count	PCI
0004h[3]	0000	0000	000x	0100	r/w	DMA1 CH2 Base and Current Address	PCI
0005h[3]	0000	0000	000x	0101	r/w	DMA1 CH2 Base and Current Count	PCI
0006h[3]	0000	0000	000x	0110	r/w	DMA1 CH3 Base and Current Address	PCI
0007h[3]	0000	0000	000x	0111	r/w	DMA1 CH3 Base and Current Count	PCI
0008h[3]	0000	0000	000x	1000	r/w	DMA1 Status(r) Command(w) Register	PCI
0009h[3]	0000	0000	000x	1001	wo	DMA1 Write Request	PCI
000Ah[3]	0000	0000	000x	1010	wo	DMA1 Write Single Mask Bit	PCI
000Bh[3]	0000	0000	000x	1011	wo	DMA1 Write Mode	PCI
000Ch[3]	0000	0000	000x	1100	wo	DMA1 Clear Byte Pointer	PCI
000Dh[3]	0000	0000	000x	1101	wo	DMA1 Master Clear	PCI
000Eh[3]	0000	0000	000x	1110	wo	DMA1 Clear Mask	PCI
000Fh[3]	0000	0000	000x	1111	r/w	DMA1 Read/Write All Mask Register Bits	PCI
0020h	0000	0000	001x	xx00	r/w	INT 1 Control	PCI/ISA
0021h	0000	0000	001x	xx01	r/w	INT 1 Mask	PCI/ISA
0040h	0000	0000	010x	0000	r/w	Timer Counter 1 - Counter 0 Count	PCI/ISA
0041h	0000	0000	010x	0001	r/w	Timer Counter 1 - Counter 1 Count	PCI/ISA
0042h	0000	0000	010x	0010	r/w	Timer Counter 1 - Counter 2 Count	PCI/ISA
0043h	0000	0000	010x	0011	wo	Timer Counter 1 Command Mode	PCI/ISA
0060h[1]	0000	0000	0110	0000	r	Reset XBus IRQ12/M and IRQ1	PCI/ISA
0061h	0000	0000	0110	0001	r/w	NMI Status and Control	PCI/ISA

Address (hex)	Address (bits) FEDC	BA98	7654	3210	Type	Name	Access
0070h[1]	0000	0000	0111	0xx0	wo	CMOS RAM Address and NMI Mask Reg	PCI/ISA
0080h[2,3]	0000	0000	100x	0000	r/w	DMA Page (Reserved)	PCI/ISA
0081h[3]	0000	0000	100x	0001	r/w	DMA Channel 2 Page	PCI/ISA
0082h[3]	0000	0000	1000	0010	r/w	DMA Channel 3 Page	PCI/ISA
0083h[3]	0000	0000	100x	0011	r/w	DMA Channel 1 Page	PCI/ISA
0084h[2,3]	0000	0000	100x	0100	r/w	DMA Page (Reserved)	PCI/ISA
0085h[2,3]	0000	0000	100x	0101	r/w	DMA Page (Reserved)	PCI/ISA
0086h[2,3]	0000	0000	100x	0110	r/w	DMA Page (Reserved)	PCI/ISA
0087h[3]	0000	0000	100x	0111	r/w	DMA Channel 0 Page	PCI/ISA
0088h[2,3]	0000	0000	100x	0100	r/w	DMA Page (Reserved)	PCI/ISA
0089h[3]	0000	0000	100x	1001	r/w	DMA Channel 6 Page	PCI/ISA
008Ah[3]	0000	0000	100x	1010	r/w	DMA Channel 7 Page	PCI/ISA
008Bh[3]	0000	0000	100x	1011	r/w	DMA Channel 5 Page	PCI/ISA
008Ch[2,3]	0000	0000	100x	1100	r/w	DMA Page (Reserved)	PCI/ISA
008Dh[2,3]	0000	0000	100x	1101	r/w	DMA Page (Reserved)	PCI/ISA
008Eh[2,3]	0000	0000	100x	1110	r/w	DMA Page (Reserved)	PCI/ISA
008Fh[3]	0000	0000	100x	1111	r/w	DMA low page Register Refresh	PCI/ISA
00A0h	0000	0000	101x	xx00	r/w	INT 2 Control	PCI/ISA
00A1h	0000	0000	101x	xx01	r/w	INT 2 Mask	PCI/ISA
00B2h	0000	0000	1011	0010	r/w	Advanced Power Management Control	PCI
00B3h	0000	0000	1011	0011	r/w	Advanced Power Management Status	PCI
00C0h[3]	0000	0000	1100	000x	r/w	DMA2 CH0 Base and Current Address	PCI
00C2h[3]	0000	0000	1100	001x	r/w	DMA2 CH0 Base and Current Count	PCI
00C4h[3]	0000	0000	1100	010x	r/w	DMA2 CH1 Base and Current Address	PCI
00C6h[3]	0000	0000	1100	011x	r/w	DMA2 CH1 Base and Current Count	PCI
00C8h[3]	0000	0000	1100	100x	r/w	DMA2 CH2 Base and Current Address	PCI

Address (hex)	Address (bits)				Type	Name	Access
	FEDC	BA98	7654	3210			
00CAh[3]	0000	0000	1100	101x	r/w	DMA2 CH2 Base and Current Count	PCI
00CCh[3]	0000	0000	1100	110x	r/w	DMA2 CH3 Base and Current Address	PCI
00CEh[3]	0000	0000	1100	111x	r/w	DMA2 CH3 Base and Current Count	PCI
00D0h[3]	0000	0000	1101	000x	r/w	DMA2 Status(r) Command(w)	PCI
00D2h[3]	0000	0000	1101	001x	wo	DMA2 Write Request	PCI
00D4h[3]	0000	0000	1101	010x	wo	DMA2 Write Single Mask Bit	PCI
00D6h[3]	0000	0000	1101	011x	wo	DMA2 Write Mode	PCI
00D8h[3]	0000	0000	1101	100x	wo	DMA2 Clear Byte Pointer	PCI
00DAh[3]	0000	0000	1101	101x	wo	DMA2 Master Clear	PCI
00DCh[3]	0000	0000	1101	110x	wo	DMA2 Clear Mask	PCI
00DEh[3]	0000	0000	1101	111x	r/w	DMA2 Read/Write All Mask Register Bits	PCI
00F0h[1]	0000	0000	1111	0000	wo	Coprocessor Error	PCI/ISA
04D0h	0000	0100	1101	0000	r/w	INT-1 edge/level control	PCI/ISA
04D1h	0000	0100	1101	0001	r/w	INT-2 edge/level control	PCI/ISA
0CF9h	0000	1100	1111	1001	r/w	Reset Control	PCI

NOTES:

1. Read and write accesses to these locations are always broadcast to the ISA Bus.
2. Write accesses to these locations are broadcast to the ISA Bus. Read Accesses are not.

   PIIX: If programmed in the ISA Controller Recovery Timer register, the PIIX will not alias the 90h-9Fh address range with the following addresses: 80h, 84–86h, 88h, and 8C–8Eh. In this case accesses to the 90h - 9Fh address range for the previously specified addresses are forwarded to the ISA Bus for both reads and writes and are ignored during ISA Master cycles (i.e., They are no-longer considered PIIX registers). Note, that port 92 is always a distinct ISA register and is always forwarded to the ISA Bus.

   PIIX3: If programmed in the ISA Controller Recovery Timer register, the PIIX3 does not alias the entire 90h-9Fh address range. These locations are considered ISA Bus register locations and not PIIX3 registers.
3. ISA-Compatible DMA register I/O space accesses are always subtractively decoded.

Table 6. PCI Bus Master IDE I/O Registers

Offset From Base Address	Mnemonic	Register	Register Access
00h	BMICP	Bus Master IDE Command (primary)	R/W
01h	—	Reserved	—
02h	BMISP	Bus Master IDE Status (primary)	R/WC
03h	—	Reserved	—
04–07h	BMIDTPP	Bus Master IDE Descriptor Table Pointer (primary)	R/W
08h	BMICS	Bus Master IDE Command (secondary)	R/W
09h	—	Reserved	—
0Ah	BMISS	Bus Master IDE Status (secondary)	R/WC
0Bh	—	Reserved	—
0C–0Fh	BMIDTPS	Bus Master IDE Descriptor Table Pointer (secondary)	R/W

NOTE:
The base address is programmable via the BMIBA Register (20–23h; function 1)

Table 7. USB Host/Controller I/O Registers

I/O Address	Mnemonic	Register Description	Register Access
Base + (00–01h)	USBCMD	USB Command	R/W
Base + (02–03h)	USBSTS	USB Status	R/WC
Base + (04–05h)	USBINTR	USB Interrupt Enable	R/W
Base + (06–07h)	FRNUM	Frame Number	R/W**
Base + (08–0Bh)	FRBASEADD	Frame List Base Address	R/W
Base + 0Ch	SOFMOD	Start Of Frame Modify	R/W
Base + (10–11h)	PORTSC1	Port 1 Status/Control	R/WC**
Base + (12–13h)	PORTSC2	Port 2 Status/Control	R/WC**

** NOTE: These registers are WORD writeable only. Byte writes to these registers have unpredictable effects.

2.2. PCI Configuration Registers—PCI To ISA Bridge (Function 0)

2.2.1. VID—VENDOR IDENTIFICATION REGISTER (Function 0)

Address Offset: 00–01h
Default Value: 8086h
Attribute: Read Only

The VID Register contains the vendor identification number. This register, along with the Device Identification Register, uniquely identifies any PCI device. Writes to this register have no effect.

Bit	Description
15:0	**Vendor Identification Number.** This is a 16-bit value assigned to Intel.

2.2.2. DID—DEVICE IDENTIFICATION REGISTER (Function 0)

Address Offset: 02–03h
Default Value: 122Eh (PIIX)
7000h (PIIX3)
Attribute: Read Only

The DID Register contains the device identification number. This register, along with the VID Register, define the PIIX. Writes to this register have no effect.

Bit	Description
15:0	**Device Identification Number.** This is a 16-bit value assigned to the PIIX.

2.2.3. PCICMD—COMMAND REGISTER (Function 0)

Address Offset: 04–05h
Default Value: 0007h
Attribute: Read/Write

This 16-bit register provides basic control over the PIIX's ability to respond to PCI cycles.

Bit	Description
15:10	**Reserved.** Read as 0.
9	**Fast Back-to-Back Enable. (Not Implemented)** This bit is hardwired to 0.
8	**PIIX: Reserved.** Read as 0. **PIIX3: SERR# Enable (SERRE).** 1=Enable. 0=Disable. When enabled (and bit 3=1 in the DLC register), a delayed transaction timeout causes the PIIX3 to assert the SERR# signal. The PCISTS register reports the status of the SERR# signal.
7:5	**Reserved.** Read as 0.
4	**Postable Memory Write Enable. (Not Implemented)** This bit is hardwired to 0.
3	**Special Cycle Enable (SCE)**. 1=Enable, the PIIX/PIIX3 recognizes shutdown special cycle. 0=Disable, the PIIX/PIIX3 ignores all PCI Special Cycles.

Bit	Description
2	**Bus Master Enable (BME). (Not Implemented)** The PIIX/PIIX3 does not support disabling its bus master capability. This bit is hardwired to 1.
1	**Memory Access Enable (MAE). (Not Implemented)** The PIIX/PIIX3 does not support disabling access to main memory. This bit is hardwired to 1.
0	**I/O Space Access Enable (IOSE).** The PIIX/PIIX3 does not support disabling its response to PCI I/O cycles. This bit is hardwired to 1.

2.2.4. PCISTS—PCI DEVICE STATUS REGISTER (Function 0)

Address Offset: 06–07h
Default Value: 0200h
Attribute: Read/Write Clear

The PCISTS Register reports the occurrence of a PCI master-abort by the PIIX/PIIX3 or a PCI target-abort when the PIIX/PIIX3 is a master. The register also indicates the PIIX/PIIX3 DEVSEL# signal timing.

Bit	Description
15	**Detected Parity Error (PERR). (Not Implemented)** Read as 0.
14	**PIIX: Signaled SERR# Status (SERRS). (Not Implemented)** Read as 0. **PIIX3: Signaled SERR# Status (SERRS)—R/WC.** When the PIIX3 asserts the SERR# signal, this bit is set to 1. Software sets this bit to a 0 by writing a 1 to it
13	**Master-Abort Status (MA)—R/WC.** When the PIIX, as a master (for the ISA bridge function), generates a master-abort, MA is set to a 1. Software sets MA to 0 by writing a 1 to this bit location.
12	**Received Target-Abort Status (RTA)—R/WC.** When the PIIX/PIIX3 is a master on the PCI Bus (for the ISA bridge function) and receives a target-abort, this bit is set to a 1. Software sets RTA to 0 by writing a 1 to this bit location.
11	**Signaled Target-Abort Status (STA)—R/WC.** This bit is set when the PIIX/PIIX3 ISA bridge function is targeted with a transaction that the PIIX/PIIX3 terminates with a target abort. Software sets STA to 0 by writing a 1 to this bit location.
10:9	**DEVSEL# Timing Status (DEVT)—RO.** The PIIX/PIIX3 always generates DEVSEL# with medium timing for ISA functions. Thus, DEVT=01. This DEVSEL# timing does not include Configuration cycles.
8	**PERR# Response. (Not Implemented)** Read as 0.
7	**Fast Back to Back—RO.** This bit indicates to the PCI Master that PIIX/PIIX3 as a target is capable of accepting fast back-to-back transactions.
6:0	**Reserved.** Read as 0s.

2.2.5. RID—REVISION IDENTIFICATION REGISTER (Function 0)

Address Offset: 08h
Default Value: Refer to applicable specification update document
Attribute: Read Only

This 8 bit register contains device stepping information. Writes to this register have no effect.

Bit	Description
7:0	**Revision ID Byte.** The register is hardwired to the default value.

2.2.6. CLASSC—CLASS CODE REGISTER (Function 0)

Address Offset: 09–0Bh
Default Value: 060100h
Attribute: Read Only

This register contains the device programming interface information related to the Sub-Class Code and Base Class Code definition for the PIIX3 (function 0). This register also identifies the Base Class Code and the function sub-class in relation to the Base Class Code.

Bit	Description
23:16	**Base Class Code (BASEC).** 06h=Bridge device.
15:8	**Sub-Class Code (SCC)**. 01h=PCI-to-ISA Bridge.
7:0	**Programming Interface (PI).** 00h=hardwired as a PCI-to-ISA Bridge.

2.2.7. HEDT—HEADER TYPE REGISTER (Function 0)

Address Offset: 0Eh
Default Value: 80h
Attribute: Read Only

The HEDT Register identifies the PIIX/PIIX3 as a multi-function device.

Bit	Description
7:0	**Device Type (DEVICET).** 80h=multi-function device.

2.2.8. IORT—ISA I/O RECOVERY TIMER REGISTER (Function 0)

Address Offset: 4Ch
Default Value: 4Dh
Attribute: Read/Write

The I/O recovery mechanism in the PIIX/PIIX3 is used to add additional recovery delay between CPU or PCI master originated 8-bit and 16-bit I/O cycles to the ISA Bus. The PIIX/PIIX3 automatically forces a minimum delay of 3.5 SYSCLKs between back-to-back 8 and 16-bit I/O cycles to the ISA Bus. This delay is measured from the rising edge of the I/O command (IOR# or IOW#) to the falling edge of the next I/O command. If a delay of greater than 3.5 SYSCLKs is required, the ISA I/O Recovery Time Register can be programmed to

increase the delay in increments of SYSCLKs. No additional delay is inserted for back-to-back I/O "sub cycles" generated as a result of byte assembly or disassembly. This register defaults to 8 and 16-bit recovery enabled with one SYSCLK clock added to the standard I/O recovery.

Bit	Description
7	**DMA Reserved Page Register Aliasing Control (DMAAC).** When DMAAC=0, the PIIX/PIIX3 aliases I/O accesses in the 90–9Fh range to the 80–8Fh range. In this case, the PIIX/PIIX3 only forwards write accesses to these locations to the ISA Bus. PIIX: When DMAAC=1, the PIIX disables aliasing for the following registers; 80h, 84–86h, 88h, and 8C–8Eh. When disabled, the PIIX forwards read and write accesses to these registers to the ISA. PIIX3: When DMAAC=1, the PIIX3 disables aliasing for the entire 90–9Fh range (they are considered ISA bus register locations). When disabled, the PIIX3 forwards read ands write accesses to these registers to ISA. Note, that port 92h is always a distinct ISA register in the 90–9Fh range and is always forwarded to the ISA Bus. When DMAAC=1, ISA master accesses to the 90–9Fh range are ignored by the PIIX. Also, when DMAAC=1, the PIIX/PIIX3 does not re-load the power management fast-off-timer with its original value for accesses to the 90–9Fh address range.
6	**8-Bit I/O Recovery Enable.** 1=Enable the recovery time programmed in bits [5:3]. 0=Disable recovery times in bits [5:3] and the recovery timing of 3.5 SYSCLKs is inserted.
5:3	**8-Bit I/O Recovery times.** When bit 6=1, this 3-bit field defines the recovery time for 8-bit I/O. **Bit[5:3] SYSCLK Bit[5:3] SYSCLK** 001 1 101 5 010 2 110 6 011 3 111 7 100 4 000 8
2	**16-Bit I/O Recovery Enable.** 1=Enable, the recovery times programmed in bits [1:0]. 0=Disable, programmable recovery times in bits [1:0] and the recovery timing of 3.5 SYSCLKs is inserted.
1:0	**16-Bit I/O Recovery Times.** When bit 2=1, this 2-bit defines the recovery time for 16-bit I/O. **Bit[1:0] SYSCLK** 01 1 10 2 11 3 00 4

2.2.9. XBCS—X-BUS CHIP SELECT REGISTER (Function 0)

Address Offset:	4Eh (PIIX) 4E–4Fh (PIIX3)
Default Value:	03h
Attribute:	Read/Write

This register enables/disables accesses to the RTC, keyboard controller, IOAPIC (PIIX3 only), and BIOS. Disabling any of these bits prevents the chip select and X-Bus output enable control signal (XOE#) for that device from being generated. This register also provides coprocessor error and mouse functions.

Bit	Description
15:9	**PIIX3: Reserved.**
8	**PIIX3: APIC Chip Select.** When enabled (bit 8=1), APICCS# is asserted for PCI memory accesses to the programmable IOAPIC region. This cycle is forwarded to the ISA bus. The default IOAPIC addresses are memory FEC0_0000h and FEC0_0010h. These can be relocated via the APIC Base Address Relocation Register. When disabled (bit 8=0), the PCI cycle is ignored by PIIX3 and APICCS# and XOE# are not generated. Note that APICCS# is not generated for ISA-originated cycles. This bit also controls the function of two shared signals on the PIIX3 as shown below: **Shared Signal Name** / **Bit 8=0** / **Bit 8=1** TESTIN#/APICREQ# / TESTIN# / APICREQ# PCIRST#/APICACK# / PCIRST# / APICACK#
7	**Extended BIOS Enable.** When bit 7=1 (enabled), PCI master accesses to locations FFF80000–FFFDFFFFh are forwarded to ISA and result in the generation of BIOSCS# and XOE#. When forwarding the additional 384-Kbyte region at the top of 4 Gbytes, the PIIX/PIIX3 allows the PCI address A[23:20] to propagate to the ISA LA[23:20] lines as all 1's, aliasing this 384-Kbyte region to the top of the 16-Mbyte space. To avoid contention, ISA addin memory must not be present in this region (00F80000–00FDFFFFh). When bit 7=0, the PIIX/PIIX3 does not generate BIOSCS# or XOE#.
6	**Lower BIOS Enable.** When bit 6=1 (enabled), PCI master, or ISA master accesses to the lower 64-Kbyte BIOS block (E0000–EFFFFh) at the top of 1 Mbyte, or the aliases at the top of 4 Gbyte (FFFE0000–FFFEFFFFh) result in the generation of BIOSCS# and XOE#. When forwarding the region at the top of 4 Gbytes to the ISA Bus, the ISA LA[23:20] lines are all 1's, aliasing this region to the top of the 16-Mbyte space. To avoid contention, ISA addin memory must not be present in this region (00F80000–00FDFFFFh). When bit 6=0, the PIIX/PIIX3 does not generate BIOSCS# or XOE# during these accesses and does not forward the accesses to ISA.
5	**Coprocessor Error function Enable.** 1=Enable; the FERR# input, when asserted, triggers IRQ13 (internal). FERR# is also used to gate the IGNNE# output.
4	**IRQ12/M Mouse Function Enable.** 1=Mouse function; 0=Standard IRQ12 interrupt function.
3	**Reserved.**
2	**BIOSCS# Write Protect Enable.** 1=Enable (BIOSCS# is asserted for BIOS memory read and write cycles in decoded BIOS region); 0=Disable (BIOSCS# is only asserted for BIOS read cycles).
1	**Keyboard Controller Address Location Enable.** 1=Enable KBCS# and XOE# for address locations 60h and 64h. 0=Disable KBCS#/XOE# for accesses to these locations.
0	**RTC Address Location Enable.** 1=Enable RTCCS#/RTCALE and XOE# for accesses to address locations 70–77h. 0=Disable RTCCS#/RTCALE and XOE# for these accesses.

2.2.10. PIRQRC[A:D]—PIRQx ROUTE CONTROL REGISTERS (Function 0)

Address Offset : 60h (PIRQRCA#)—63h (PIRQRCD#)
Default Value: 80h
Attribute: R/W

These registers control the routing of the PIRQ[A:D]# signals to the IRQ inputs of the interrupt controller. Each PIRQx# can be independently routed to any one of 11 interrupts. All four PIRQx# lines can be routed to the same IRQx input. Note that the IRQ that is selected through bits [3:0] must be set to level sensitive mode in the corresponding ELCR Register. When a PIRQ signal is routed to an interrupt controller IRQ, the PIIX/PIIX3 masks the corresponding IRQ signal.

Bit	Description
7	**Interrupt Routing Enable.** 0=Enable; 1=Disable
6:4	**Reserved.** Read as 0s.
3:0	**Interrupt Routing.** When bit 7=0, this field selects the routing of the PIRQx to one of the interrupt controller interrupt inputs.

Bits[3:0]	IRQ Routing	Bits[3:0]	IRQ Routing	Bits[3:0]	IRQ Routing
0000	Reserved	0110	IRQ6	1011	IRQ11
0001	Reserved	0111	IRQ7	1100	IRQ12
0010	Reserved	1000	Reserved	1101	Reserved
0011	IRQ3	1001	IRQ9	1110	IRQ14
0100	IRQ4	1010	IRQ10	1111	IRQ15
0101	IRQ5				

2.2.11. TOM—TOP OF MEMORY REGISTER (Function 0)

Address Offset: 69h
Default Value: 02h
Attribute: Read/Write

This register enables the forwarding of ISA or DMA memory cycles to the PCI Bus and sets the top of main memory accessible by ISA or DMA devices. In addition, this register controls the forwarding of ISA or DMA accesses to the lower BIOS region (E0000–EFFFFh) and the 512–640-Kbyte main memory region (80000–9FFFFh). The Top Of Memory configuration register must be set by the BIOS.

Bit	Description
7:4	**Top Of Memory.** The top of memory can be assigned in 1-Mbyte increments from 1–16 Mbytes. ISA or DMA accesses within this region, and not in the memory hole region, are forwarded to PCI.

Bits[7:4]	Top of Memory	Bits[7:4]	Top of Memory	Bits[7:4]	Top of Memory
0000	1 Mbyte	0110	7 Mbyte	1011	12 Mbyte
0001	2 Mbyte	0111	8 Mbyte	1100	13 Mbyte
0010	3 Mbyte	1000	9 Mbyte	1101	14 Mbyte
0011	4 Mbyte	1001	10 Mbyte	1110	15 Mbyte
0100	5 Mbyte	1010	11 Mbyte	1111	16 Mbyte
0101	6 Mbyte				

Note that the PIIX/PIIX3 only supports a main memory hole at the top of 16 Mbytes. Thus, If a 1-Mbyte memory hole is created for the Host-to-PCI Bridge DRAM controller between 15 and 16 Mbytes, the PIIX/PIIX3 Top Of Memory should be set at 15 Mbytes.

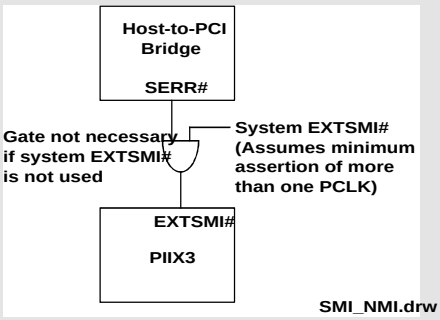
Bit	Description
3	**ISA/DMA Lower BIOS Forwarding Enable.** 1=Enable (forwarded to PCI, if XBCS Register bit 6=0); 0=Disable (contained to ISA). Note that If the XBCS Register bit 6=1, ISA/DMA accesses in this region are always contained to ISA.
2	**PIIX: Reserved** **PIIX3: A,B Segment Forwarding Enable.** 1=Enable (forward to PCI), 0 = Disable (default, contain to ISA). When enabled, this bit allows ISA Master and DMA memory accesses to A0000h–BFFFFh range to be forwarded to PCI. When disabled, these accesses are contained to ISA.
1	**ISA/DMA 512–640-Kbyte Region Forwarding Enable**. 1=Enable (forwarded to PCI); 0=Disable (contained to ISA).
0	**Reserved**

2.2.12. MSTAT—MISCELLANEOUS STATUS REGISTER (Function 0)

Address Offset:	6B–6Ah
Default Value:	Undefined
Attribute:	PIIX: Read/Write (bits [1:0] are read only) PIIX3: Read/Write Clear(bits 1 is read only)

This register provides miscellaneous status and control functions.

Bit	Description
15	**PIIX: Reserved.** **PIIX3: SERR# Generation Due To Delayed Transaction—R/WC.** PIIX3 sets this bit to a 1 when it generates SERR# due to a delayed transaction. Software sets this bit to a 0 by writing a 1 to it.
14:8	**Reserved.**
7	**PIIX: Reserved** **PIIX3: NB Retry Enable (NBRE)--R/W,** 1=Enable, 0 = Disable (default). This bit, when enabled, causes the PIIX3 to retry, without initiating a delayed transaction, CPU initiated, non-LOCK#, PCI cycles. No delayed transactions to PIIX3 may currently be pending and passive release must be active. Delayed Transactions and Passive Release must both be enabled via the DLC Register (function 0, offset 82h). When disabled, the PIIX3 accepts these cycles as normal, which may include retry with initiation of a delayed transaction.

Bit	Description
6	**PIIX: Reserved.** **PIIX3: EXTSMI# Mode Enable (ESMIME) —R/W.** 1=Enable. 0=Disable (default). This bit is used to enable a special SERR# handling protocol between the host-to-PCI bridge and the PIIX3. When ESMIME is enabled, the operating mode of the EXTSMI# signal is determined by the duration of the EXTSMI# signal. If the EXTSMI# signal is asserted for one PCLK, an SERR# is reported in the NMISC Register (address 61h) and an NMI is generated, if enabled in the NMI Registers (address 61h and 70h). If EXTSMI# is asserted for more than one PCLK, the standard mode for handling EXTSMI# is used (i.e., same as when this bit is set to 0). When disabled (standard mode) and a falling edge is detected on EXTSMI#, an SMI is signaled, if EXTSMI# signaling is enabled. 
5	**Reserved.**
4	**PIIX: Reserved.** **PIIX3: USB Enable (USBE) —R/W.** 1=Enable. 0=Disable (default). When disabled, all USB functionality is disabled. This bit must be set to 1 to access function 2 configuration space. Note that dynamically disabling USB is not supported in PIIX3. This bit is used to enable/disable the USB at boot time. In case software has to disable the USB during run time it has to do the following: Software must turn off the master enable and I/O Decode Enable for function 2 (via the PCICMD Register, function 2) prior to writing this bit to 0.
3	**Reserved.**
2	**PIIX: PCI Header Type Bit Enable—R/W.** This bit controls the Header Type Bit in the PIIX register 0Eh which defines the PIIX as a multifunction device. This bit defaults to 1 (Multifunction device) and should be left in the default state. **PIIX3: Reserved.**
1	**PIIX: Internal ISA DMA or External DMA Mode Status (IEDMAS) —RO.** 0=Normal DMA Operation. This bit reports the strapping option selected on the TC signal (pulled high at reset for a value of 0). **PIIX3: Reserved,**

Bit	Description
0	**PIIX: ISA Clock Divisor Status—RO.** This bit reports the strapping option on the SYSCLK signal. 1=clock divisor of 3 (PCICLK=25 MHz). 0=Clock divisor of 4 (PCICLK=33 MHz). Note that, for PCICLK=30 MHz, a clock divisor of 4 must be selected and produces a SYSCLK of 7.5 MHz. **PIIX3: ISA Clock Divisor—R/W.** This bit controls the frequency of the SYSCLK signal. 1=clock divisor of 3 (PCICLK=25 MHz). 0=Clock divisor of 4 (PCICLK=33 MHz). Note that, for PCICLK=30 MHz, a clock divisor of 4 must be selected and produces a SYSCLK of 7.5 MHz. The default value of this bit is determined by the strapping option on the SYSCLK signal.

2.2.13. MBIRQ[1:0]—MOTHERBOARD DEVICE IRQ ROUTE CONTROL REGISTERS (Function 0)

Address Offset : 70h—MBIRQ0 (PIIX and PIIX3)
71h—MBIRQ1 (PIIX only)
Default Value: 80h
Attribute: R/W

These registers control the routing of motherboard device interrupts (MIRQ[1:0]) to the internal IRQ inputs of the interrupt controller. Each MIRQx# can be independently routed to any one of 11 interrupts.

Note that, when an MIRQ line and a PIRQ# line are steered to the same ISA interrupt, the device connected to the MIRQ line must be set for active high, level interrupts. In this case, the ISA interrupt will be masked. Bit 6 of that motherboard device IRQ Route Control Register must be programmed to a 0.

Bit	Description
7	**Interrupt Routing Enable.** 0=Enable routing; 1=Disable routing.
6	**MIRQx/IRQx Sharing Enable.** 0=Disable sharing; 1=Enable sharing. When sharing is disabled and bit 7 of this register is 0, the interrupt specified by bits [3:0] is masked. Interrupt sharing should only be enabled when the device connected to the MIRQ line and the device connected to the ISA IRQ line both produce active high, level interrupts.
5	**PIIX: Reserved** **PIIX3: IRQ0 Enable,** 1=Enable (IRQ0 Output), 0 = Disable (default, MIRQ0 input). When enabled the MIRQ0/IRQ0 pin functions as the IRQ0 output and the IDE Interrupt Status flag in Bus Master IDE Status Register is set by the IRQ15 input signal. When disabled the MIRQ0/IRQ0 pin functions as the MIRQ0 input and the IDE Interrupt Status flag in the Bus Master IDE Status Register is set by the MIRQ0 input signal.
4	**Reserved.** Read as zero.
3:0	**Interrupt Routing.** When bit 7=0, this field selects the routing of the MBIRQx to one of the interrupt controller interrupt inputs.

Bits[3:0]	IRQ Routing	Bits[3:0]	IRQ Routing	Bits[3:0]	IRQ Routing
0000	Reserved	0110	IRQ6	1011	IRQ11
0001	Reserved	0111	IRQ7	1100	IRQ12
0010	Reserved	1000	Reserved	1101	Reserved
0011	IRQ3	1001	IRQ9	1110	IRQ14
0100	IRQ4	1010	IRQ10	1111	IRQ15
0101	IRQ5				

2.2.14. MBDMA[1:0]—MOTHERBOARD DEVICE DMA CONTROL REGISTERS (Function 0)

Address Offset : 76h—MBDMA0#; 77h—MBDMA1#
Default Value: 0Ch
Attribute: R/W

For both the PIIX and PIIX3, these registers enable/disable a type F DMA transfer (3 SYSCLK) for a particular DMA channel.

For the PIIX, these registers also control the routing of motherboard device DMA signals (MDRQ[1:0] and MDAK[1:0]) to the DREQ and DACK# signals on the 8237 DMA controller unit.

Bit	Description
7	**Type F and DMA Buffer Enable (FAST).** 1=Enable for the channel selected by bits[2:0]. 0=Disable for the channel selected by bits[2:0].
6:4	**Reserved.**
3	**PIIX: Disable Motherboard Channel (DMC).** When this bit 3=0, the MDRQ/MDAK# pair associated with this channel is routed to the compatable ISA channel determined by the CHNL field (bits[2:0]). When bit 3=1, the ISA DREQ/DACK# pair is used for that channel. When a MDRQ/MDAK# pair is programmed for a given 8237 DMA channel and DMC=0 (bit 3), the corresponding DREQ/DACK# pins are masked for that channel. When DMC=1, the MDRQ/MDAK# signals are masked. If both motherboard DMAs are used, the motherboard DMAs should be programmed to different compatible DMA channels. Programming both motherboard DMAs to the same compatible DMA channel results in unpredictable device operation. **PIIX3: Reserved.**
2:0	**PIIX: Type F and Motherboard DMA Channel Routing (CHNL).** When DMC=0, this field steers the corresponding MDRQ/MDAK# signals to a compatable ISA channel for a motherboard peripheral (and if FAST=1, also enables type F transfers and the 4-byte DMA buffer). When DMC=1 and FAST=1, this field enables type F transfers and the 4-byte DMA buffer for an ISA peripheral on a given channel. **PIIX3: Type F DMA Channel Routing (CHNL).** When FAST=1, this field enables type F transfers and the 4-byte DMA buffer for an ISA peripheral on the selected channel. **Bits[2:0] DMA channel Bits[2:0] DMA channel** 000 0 100 default (disabled) 001 1 101 5 010 2 110 6 011 3 111 7

2.2.15. PCSC—PROGRAMMABLE CHIP SELECT CONTROL REGISTER (Function 0)

Address Offset: 78–79h
Default Value: 0002h
Attribute: Read/Write

This register controls the assertion of the PCS# programmable chip select signal. The PCS# signal is asserted for subtractively decoded I/O cycles generated by PCI masters that fall in the range specified by this register. The address is programmable to any 16-bit I/O space location and the range is programmable to be 4, 8 or 16 bytes. A split range is precluded. The upper sixteen address bits (AD[31:16]) must be zero for the

address to be decoded and the PCS# signal asserted. The PCS# signal is never asserted for ISA bus masters access.

Bit	Description
15:2	**PCS Address (PCSADDR).** This field defines a 16-bit I/O space address (4 byte range) that causes the PCS# signal to assert. Address bits [3:2] may be masked (considered "don't care") by programming bits [1:0] of this register.
1:0	**PCS Address Mask.** When bit 1=1, PCSADDR3 is masked. When bit 0=1, PCSADDR2 is masked. **Bits[1:0] Range** 00 4 bytes (default) 01 8 bytes, contiguous 10 Disabled 11 16 bytes, contiguous

2.2.16. APICBASE—APIC BASE ADDRESS RELOCATION REGISTER (Function 0) (PIIX3 Only)

Address Offset: 80h
Default Value: 00h
Attribute: Read/Write

This register provides the modifier for the APIC base address. APIC is mapped in the memory space at the locations FEC0_xy00h and FEC0_xy10h (x=0-Fh, y=0,4,8,Ch). The value of 'y' is defined by bits [1,0] and the value of 'x' is defined by bits [5:2]. Thus, the relocation register provides 1-Kbyte address granularity (i.e., potentially up to 64 IOAPICs can be uniformly addresses in the memory space). The default value of 00h provides mapping of the IOAPIC unit at the addresses FEC0_0000h and FEC0_0010h.

Bit	Description
7	**Reserved.**
6	**A12 Mask.** This bit determines selects whether APICCS# is generated for one or two IOAPIC address ranges. When bit 6=1, address bit 12 is ignored allowing the APICCS# signal to be generated for two consecutive IOAPIC address ranges. External logic is needed to select individual IOAPICs by combining SA12 and APICCS#. For example, when bit 6=1 (and x and y = 0), APICCS# is generated for addresses FEC0_0000h, FEC0_0010, as well as FEC0_1000h, FEC0_1010. When bit 6=0, APICCS# is generated for one IOAPIC address range.
5:2	**X-Base Address.** Bits[5:2] are compared with PCI address bits AD[15:12], respectively.
1:0	**Y-Base Address.** Bits[1:0] are compared with PCI address bits AD[11:10], respectively.

2.2.17. DLC—DETERMINISTIC LATENCY CONTROL REGISTER (Function 0) (PIIX3 Only)

Address Offset: 82h
Default Value: 00h
Attribute: Read/Write

This register enables/disables the Delayed Transaction and Passive Release functions, respectively. When enabled, these functions make the PIIX3 PCI revision 2.1 compliant. Note that neither delayed completion nor passive release have any effect on bus master IDE and type F DMA transfers.

The 2.1 revision of the PCI specification requires much tighter controls on target and master latency. Targets must respond with TRDY# or STOP# within 16 clocks of FRAME#, and masters must assert IRDY# within 8 PCI clocks for any data phase. PCI cycles to or from ISA typically take longer than this. The PIIX3 provides a programmable ***delayed completion*** mechanism described in the PCI specification to meet the required target latencies.

ISA bridges also support GAT mode, which will now violate the spirit of the PCI specification. The PIIX3 provides a programmable ***passive release*** mechanism to meet the required master latencies. When passive release is enabled in the PIIX3, ISA masters may see long delays in accesses to any PCI memory, including the main DRAM array. The ISA GAT mode is not supported with passive release enabled. ISA masters must honor IOCHRDY.

Bit	Description
7:4	**Reserved.**
3	**SERR# Generation Due To Delayed Transaction Timeout Enable.** 1=Enable. 0=Disable.
2	**USB Passive Release Enable (USBPR).** 1=Enable 0=Disable.
1	**Passive Release Enable.** 1=Enable the Passive Release mechanism encoded on the PHOLD# signal (default). 0=PHOLD# functions identically to the way it did in the PIIX (82371FB).
0	**Delayed Transaction Enable.** 1=Enable the Delayed Transaction mechanism when the PIIX3 is the target of a PCI transaction (default). 0=PIIX3 resonds as a target of a PCI transaction identically to the way the PIIX (82371FB) responded.

2.2.18. SMICNTL—SMI CONTROL REGISTER (Function 0)

Address Offset: A0h
Default Value: 08h
Attribute: Read/Write

The SMICNTL Register provides Fast Off Timer control, STPCLK# enable/disable, and CPU clock scaling. This register also enables/disables the system management interrupt (SMI).

Bit	Description
7:5	**Reserved**
4:3	**Fast Off Timer Freeze (CTMRFRZ).** This field enables/disables the Fast Off Timer and when enabled, selects the timer count granularity as shown below: **Bits[4:3] / Count Granularity (33 MHz PCICLK) / Count Granularity (30 MHz PCICLK) / Count Granularity (25 MHz PCICLK)** 00 / 1 Minute / 1.1 Minute / 1.32 Minute 01 / Disabled (default) / Disabled (default) / Disabled (default) 10 / 1 PCICLK / 1 PCICLK / 1 PCICLK 11 / 1 Msec / 1.1 Msec / 1.32 Msec
2	**STPCLK# Scaling Enable (CSTPCLKSC).** 1=Enable; 0=Disable. When enabled (and bit 1=1), the high and low times for the STPCLK# signal are controlled by the Clock Scaling STPCLK# High Timer and Clock Scaling STPCLK# Low Timer Registers.
1	**STPCLK# Signal Enable (CSTPCLKE).** 1=Enable; 0=Disable. When enabled, an APMC Register read causes STPCLK# to be asserted. When disabled, the STPCLK# signal is disabled and is negated (high). Software can set this bit to 0 by writing a 0 to it.
0	**SMI# Gate (CSMIGATE).** 1=Enable; 0=Disable. When enabled, a system management interrupt condition asserts the SMI# signal. When disabled, the SMI# signal is masked and negated. This bit only affects the SMI# signal and does not affect the detection/recording of SMI events (i.e., This bit does not effect the SMI status bits in the SMIREQ Register). Thus, if an SMI is pending when this bit is set to 1, the SMI# signal is asserted.

2.2.19. SMIEN—SMI ENABLE REGISTER (Function 0)

Address Offset: A2–A3h
Default Value: 0000h
Attribute: Read/Write

This register enables the generation of SMI (asserting the SMI# signal) for the associated hardware events (bits [5:0]), and software events (bit 7). When a hardware event is enabled, the occurrence of a corresponding event results in the assertion of SMI#, if enabled via the SMICNTL Register. The SMI# is asserted independent of the current power state (Power-On or Fast Off). The default for all sources in this register is disabled.

Bit	Description
15:9	**Reserved.**
8	**PIIX: Reserved.** **PIIX3: Legacy USB SMI Enable** 1=Enable USB Legacy logic to generate SMI#. 0=Disable (default).

Bit	Description
7	**APMC Write SMI Enable.** 1=Enable; 0=Disable.
6	**EXTSMI# SMI Enable.** 1=Enable; 0=Disable.
5	**Fast Off Timer SMI Enable.** 1=Enable; 0=Disable. When enabled, the timer generates an SMI when it decrements to zero.
4	**IRQ12 SMI Enable (PS/2 Mouse Interrupt).** 1=Enable; 0=Disable.
3	**IRQ8 SMI Enable (RTC Alarm Interrupt).** 1=Enable; 0=Disable.
2	**IRQ4 SMI Enable (COM2/COM4 Interrupt or Mouse).** 1=Enable; 0=Disable.
1	**IRQ3 SMI Enable (COM1/COM3 Interrupt or Mouse).** 1=Enable; 0=Disable.
0	**IRQ1 SMI Enable (Keyboard Interrupt).** 1=Enable; 0=Disable.

2.2.20. SEE—SYSTEM EVENT ENABLE REGISTER (Function 0)

Address Offset: A4–A7h
Default Value: 00000000h
Attribute: Read/Write

This register enables hardware events as system events and break events for power management control. The default for each system/break event in this register is disabled.

System events: Activity by these events can keep the system from powering down. When a system event is enabled, the corresponding hardware event activity prevents a Fast Off powerdown condition by reloading the Fast Off Timer with its initial count.

Break events: These events can awaken a powered down system. When a break event is enabled, the corresponding hardware event activity powers up the system by negating STPCLK#.

Bit	Description
31	**Fast Off SMI Enable (FSMIEN).** 1=Enable (system and break events); 0=Disable.
30	**INTR Enable (FINTREN).** 1=Enable (break event); 0=Disable. When enabled, INTR is used as a global break event. In this case, any IRQ that is generated causes the system to powerup via the negation of STPCLK#, regardless of the state of bits[15:3,1:0] in this register. For the PIIX3, this function should be disabled if an external IOAPIC is used.
29	**Fast Off NMI Enable (FNMIEN).** 1=Enable (system and break events); 0=Disable.
28	**82371FB PIIX: Reserved.** **PIIX3: Fast Off APIC Enable (FAPICEN).** 1=Enable (break event); 0=Disable.
27:16	**Reserved**
15:3	**Fast Off IRQ[15:3] Enable (FIRQ[15:3]EN).** 1=Enable (system and break events); 0=Disable.
2	**Reserved.**
1:0	**Fast Off IRQ[1:0] Enable (FIRQ[1:0]EN).** 1=Enable (system and break events); 0=Disable.

2.2.21. FTMR—FAST OFF TIMER REGISTER (Function 0)

Address Offset: A8h
Default Value: 0Fh
Attribute: Read/Write

The Fast Off Timer indicates (through an SMI) that the system has been idle for a pre-programmed period of time. When the timer expires, an SMI special cycle is generated. The count time interval is programmable (via the SMICNTL Register). The granularity of the counter is programmable via the SMICNTL Register.

NOTE

Before writing to the FTMR Register, the Fast Off Timer must be stopped via bits [4:3] of the SMICNTL Register.

Bit	Description
7:0	**Fast Off Timer Value.** Bits [7:0] contain one less than the actual count-down value. Thus, if X is programmed into this register, the countdown value is X+1. The X+1 value is loaded into the counter when an enabled system event occurs. When the Fast Off Timer reaches 00h, an SMI is generated and the timer is re-loaded with the X+1 value. When the Fast Off Timer is enabled (via the SMICNTL Register), the timer counts down from this value. A read from the FTMR Register returns the value last written.

2.2.22. SMIREQ—SMI REQUEST REGISTER (Function 0)

Address Offset: AA–ABh
Default Value: 00h
Attribute: Read/Write

The SMIREQ Register contains status bits indicating which enabled event caused an SMI.

NOTE

1. The SMIREQ bits are set, cleared, or read independently of each other and independently of the CSMIGATE bit in the SMICNTL Register.
2. If software attempts to set a status bit to 0 at the same time that the PIIX/PIIX3 is setting it to 1, the bit is set to 1.
3. Each of the SMIREQ bits is set by the PIIX/PIIX3 in response to the activation of the corresponding SMI event. If the SMI event is still active when the corresponding SMIREQ bit is set to 0, the PIIX/PIIX3 does not set the status bit back to a 1 (i.e., there is only one status indication per active SMI event).
4. When an IRQx signal is asserted, the corresponding IRQx status bit is set to a 1. If the IRQx signal is still active when software sets the corresponding status bit to 0, the status bit is not set back to a 1. The IRQx may be negated before software sets the status bit to 0. However, If the status bit is set to 0 at the same time a new IRQx is activated, the status bit remains at 1. This indicates to the SMI handler that a new SMI event has been detected.
5. If an IRQx is set in level mode and shared by two devices, the IRQ should not be enabled as an SMI# event. The PIIX's SMIREQ bits are essentially set with an edge. When the second IRQ occurs on a shared IRQ, there is no second edge and the SMI# will not be generated for the second IRQ.

Bit	Description
15:9	**Reserved**
8	**PIIX: Reserved.** **PIIX3: Legacy USB SMI Status (RLUSB).** This bit is set to 1 to indicate that the USB Legacy Keyboard logic caused an SMI. Software sets this bit to a 0 by writing a 0 to it.
7	**APM SMI Status (RAPMC).** This bit is set to 1 to indicate that a write to the APM Control Register caused an SMI. Software sets this bit to a 0 by writing a 0 to it.
6	**EXTSMI# SMI Status (REXT).** This bit is set to 1 to indicate that EXTSMI# caused an SMI. Software sets this bit to a 0 by writing a 0 to it. For the PIIX3, the EXTSMI# signal can be used to provide a special protocol between the host-to-PCI bridge and the PIIX3 (see MSTAT Register description, 82h, function 0).
5	**Fast Off Timer Expired Status (RFOT).** This bit is set to 1 to indicate that the Fast Off Timer expired and caused an SMI. Software sets this bit to a 0 by writing a 0 to it. Note that the timer re-starts counting one the next clock after it expires.
4	**IRQ12 Request SMI Status (RIRQ12).** This bit is set to 1 to indicate that IRQ12 caused an SMI. Software sets this bit to a 0 by writing a 0 to it.
3	**IRQ8# Request SMI Status (RIRQ8).** This bit is set to 1 to indicate that IRQ8# caused an SMI. Software sets this bit to a 0 by writing a 0 to it.
2	**IRQ4 Request SMI Status (RIRQ4).** This bit is set to 1 to indicate that IRQ4 caused an SMI. Software sets this bit to a 0 by writing a 0 to it.
1	**IRQ3 Request SMI Status (RIRQ3).** This bit is set to 1 to indicate that IRQ3 caused an SMI. Software sets this bit to a 0 by writing a 0 to it.
0	**IRQ1 Request SMI Status (RIRQ1).** This bit is set to 1 to indicate that IRQ1 caused an SMI. Software sets this bit to a 0 by writing a 0 to it.

2.2.23. CTLTMR—CLOCK SCALE STPCLK# LOW TIMER (Function 0)

Address Offset: ACh
Default Value: 00h
Attribute: Read/Write

The value in this register defines the duration of the STPCLK# asserted period when bit 2 in the SMICNTL Register is set to 1. The value in this register is loaded into the STPCLK# Timer when STPCLK# is asserted. The STPCLK# timer is a divide by PCI clock. The base count for a value of 0 is 42 µsec for a 50 MHz time base, 35 µsec for a 60 MHz time base, 32 µsec for a 66 MHz time base. These numbers are determined as follows: # of PCI clocks STPCLK# is asserted (or negated) = 1 + 1056 X (programmed value in register +1).

Bit	Description
7:0	**Clock Scaling STPCLK# Low Timer Value.** Bits [7:0] define the duration of the STPCLK# asserted period during clock throttling.

2.2.24. CTHTMR—CLOCK SCALE STPCLK# HIGH TIMER (Function 0)

Address Offset: AEh
Default Value: 00h
Attribute: Read/Write

The value in this register defines the duration of the STPCLK# negated period when bit 2 in the SMICNTL Register is set to 1. The value in this register is loaded into the STPCLK# Timer when STPCLK# is negated. The STPCLK# timer is a divide by PCI clock. The base count for a value of 0 is 42 µsec for a 50 MHz time base, 35 µsec for a 60 MHz time base, 32 µsec for a 66 MHz time base. These numbers are determined as follows: # of PCI clocks STPCLK# is asserted (or negated) = 1 + 1056 X (programmed value in register +1).

Bit	Description
7:0	**Clock Scaling STPCLK# High Timer Value.** Bits [7:0] define the duration of the STPCLK# negated period during clock throttling.

2.3. PCI Configuration Registers—IDE Interface (Function 1)

The PIIX/PIIX3 are multi-function devices, as indicated by bit 7 of the Header Type Register. The PCI IDE interface function uses Function 1.

2.3.1. VID—Vendor Identification Register (Function 1)

Address Offset: 00–01h
Default Value: 8086h
Attribute: Read Only

The VID Register contains the vendor identification number. This register, along with the Device Identification Register, uniquely identify any PCI device. Writes to this register have no effect.

Bit	Description
15:0	**Vendor Identification Number.** This is a 16-bit value assigned to Intel

2.3.2. DID—DEVICE IDENTIFICATION REGISTER (Function 1)

Address Offset: 02–03h
Default Value: 1230h (PIIX)
7010h (PIIX3)
Attribute: Read Only

The DID Register contains the device identification number. This register, along with the VID Register, define the PIIX. Writes to this register have no effect.

Bit	Description
15:0	**Device Identification Number.** This is a 16-bit value assigned to the PIIX.

2.3.3. PCICMD—COMMAND REGISTER (Function 1)

Address Offset: 04–05h
Default Value: 0000h
Attribute: Read/Write

The PCICMD Register controls access to the I/O space registers.

Bit	Description
15:10	**Reserved.** Read 0.
9	**Fast Back to Back Enable (FBE). (Not Implemented)** This bit is hardwired to 0.
8:5	**Reserved.** Read as 0.
4	**Memory Write and Invalidate Enable (MWI). (Not Implemented)** This bit is hardwired to 0.
3	**Special Cycle Enable (SCE). (Not Implemented)** This bit is hardwired to 0
2	**Bus Master FunctionEnable (BME).** 1=Enable. 0=Disable.
1	**Memory Space Enable (MSE). (Not Implemented)** This bit is hardwired to 1.
0	**I/O Space Enable (IOSE).** This bit controls access to the I/O space registers. When IOSE=1, access to the Legacy IDE ports (both primary and secondary) and the PCI Bus Master IDE I/O Registers is enabled. The Base Address Register for the PCI Bus Master IDE I/O Registers should be programmed before this bit is set to 1.

2.3.4. PCISTS—PCI DEVICE STATUS REGISTER (Function 1)

Address Offset: 06–07h
Default Value: 0280h
Attribute: Read/Write

PCISTS is a 16-bit status register for the IDE interface function. The register also indicates the PIIX's DEVSEL# signal timing.

Bit	Description
15	**Detected Parity Error (PERR). (Not Implemented)** Read as 0.
14	**SERR# Status (SERRS). (Not Implemented)** Read as 0.
13	**Master-Abort Status (MAS)—R/W.** When the Bus Master IDE interface function, as a master, generates a master abort, MA is set to a 1. Software sets MA to 0 by writing a 1 to this bit.
12	**Received Target-Abort Status (RTA)—R/W.** When the Bus Master IDE interface function is a master on the PCI Bus and receives a target abort, this bit is set to a 1. Software sets RTA to 0 by writing a 1 to this bit.
11	**Signaled Target Abort Status (STA)—R/W.** This bit is set when the PIIX/PIIX3 IDE interface function is targeted with a transaction that the PIIX/PIIX3 terminates with a target abort. Software resets STA to 0 by writing a 1 to this bit..
10:9	**DEVSEL# Timing Status (DEVT)—RO.** For the PIIX, DEVT=01 indicating medium timing for DEVSEL# assertion when performing a positive decode. DEVSEL# timing does not include configuration cycles.

Bit	Description
8	**Data Parity Detected (DPD). (Not Implemented)** Read as 0.
7	**Fast Back to back Capable (FBC)—RO.** Hardwired to 1. This bit indicates to the PCI Master that PIIX, as a target, is capable of accepting fast back-to-back transactions.
6:0	**Reserved**. Read as 0's.

2.3.5. RID—REVISION IDENTIFICATION REGISTER (Function 1)

Address Offset: 08h
Default Value: Refer to applicable specification update document
Attribute: Read Only

This 8 bit register contains device stepping information. Writes to this register have no effect.

Bit	Description
7:0	**Revision ID Byte.** The register is hardwired to the default value during manufacturing.

2.3.6. CLASSC—CLASS CODE REGISTER (Function 1)

Address Offset: 09–0Bh
Default Value: 010180h
Attribute: Read Only

This register contains the device programming interface information related to the Sub-Class Code and Base Class Code definition for the PIIX3 (function 1). This register also identifies the Base Class Code and the function sub-class in relation to the Base Class Code.

Bit	Description
23:16	**Base Class Code (BASEC).** 01h=Mass storage device.
15:8	**Sub-Class Code (SCC)**. 01h=IDE controller.
7:0	**Programming Interface (PI).** 80h=Capable of IDE bus master operation.

2.3.7. MLT—MASTER LATENCY TIMER REGISTER (Function 1)

Address Offset: 0Dh
Default Value: 00h
Attribute: Read / Write

MLT controls the amount of time PIIX, as a bus master, can burst data on the PCI Bus. The count value is an 8-bit quantity. However, MLT[3:0] are reserved and 0 when determining the count value. MLT is cleared and suspended when PIIX/PIIX3 is not asserting FRAME#. When PIIX/PIIX3 asserts FRAME#, the counter begins counting. If the PIIX/PIIX3 finishes its transaction before the count expires, the MLT count is ignored. If the count expires before the transaction completes (count = # of clocks programmed in MLT), PIIX/PIIX3 initiates a transaction termination as soon as its PHLDA# is removed. The number of clocks programmed in the MLT represents the guaranteed time slice (measured in PCI clocks) allotted to PIIX. The default value of MLT is 00h or 0 PCI clocks.

Bit	Description
7:4	**Master Latency Timer Count Value.** PIIX-initiated PCI burst cycles can last indefinitely, as long as PHLDA# remains active. However, if PHLDA# is negated after the burst cycle is initiated, PIIX/PIIX3 limits the burst cycle to the number of PCI Bus clocks specified by this field.
3:0	**Reserved**

2.3.8. HEDT—HEADER TYPE REGISTER (Function 1)

Address Offset: 0Eh
Default Value: 00h
Attribute: Read Only

The HEDT Register identifies the PIIX/PIIX3 as a multi-function device.

Bit	Description
7:0	**Device Type (DEVICET).** 00. Multi-function device capability for PIIX/PIIX3 is defined by the HEDT register in Function 0.

2.3.9. BMIBA—BUS MASTER INTERFACE BASE ADDRESS REGISTER (Function 1)

Address Offset: 20–23h
Default Value: 00000001h
Attribute: Read/Write

This register selects the base address of a 16 byte I/O space to provide a software interface to the Bus Master functions. Only 12 bytes are actually used (6 bytes for primary and 6 bytes for secondary).

Bit	Description
31:16	**Reserved.** Hardwired to 0.
15:4	**Bus Master Interface Base Address.** These bits provide the base address for the Bus Master interface registers and correspond to AD[15:4].
3:2	**Reserved.** Hardwired to 0.
1	**Reserved.**
0	**Resource Type Indicator (RTE)—RO.** This bit is hardwired to 1 indicating that the base address field in this register maps to I/O space.

2.3.10. IDETIM—IDE TIMING REGISTER (Function 1)

Address Offset: Primary Channel=40–41h; Secondary Channel=42–43h
Default Value: 0000h
Attribute: Read / Write Only

This register controls the PIIX's IDE interface and selects the timing characteristics of the PCI Local Bus IDE cycle. Note that primary and secondary denotations distiguish between the cables and the 0/1 denotations distiguish between master (0) and slave (1).

Bit	Description
15	**IDE Decode Enable (IDE).** 1=Enable; 0=Disable. When enabled, I/O transactions on PCI targeting the IDE ATA register blocks (command block and control block) are positively decoded on PCI and driven on the IDE interface. When disabled, these accesses are subtractively decoded to ISA.
14	**PIIX: Reserved.** **PIIX3: Slave IDE Timing Register Enable (SITRE).** 1=Enable SIDETIM Register. 0=Disable (default) SIDETIM Register. When enabled, the ISP and RTC values can be programmed uniquely for each master through the fields in this register and these values can be programmed for each slave through the SIDETIM Register. When disabled, the ISP and RTC values programmed in this register apply to both master and slave drives on each channel.
13:12	**IORDY Sample Point (ISP).** This field selects the number of clocks between DIOx# assertion and the first IORDY sample point. **Bits[13:12] — Number Of Clocks** 00 — 5 01 — 4 10 — 3 11 — 2
11:10	**Reserved**
9:8	**Recovery Time (RTC).** This field selects the minimum number of clocks between the last IORDY# sample point and the DIOx# strobe of the next cycle. **Bits[9:8] — Number Of Clocks** 00 — 4 01 — 3 10 — 2 11 — 1
7	**DMA Timing Enable Only (DTE1).** When DTE1=1, fast timing mode is enabled for DMA data transfers for drive 1. Note that PIO transfers to the IDE data port still run in compatible timing.
6	**Prefetch and Posting Enable (PPE1).** When PPE1=1, prefetch and posting to the IDE data port is enabled for drive 1.
5	**IORDY Sample Point Enable Drive Select 1 (IE1).** When IE1=0, IORDY sampling is disabled for Drive 1. The internal IORDY signal is forced asserted guaranteeing that IORDY is sampled asserted at the first sample point as specified by the ISP field in this register. When IE1=1 and the currently selected drive (via a copy of bit 4 of 1x6h) is Drive 0, all accesses to the enabled I/O address range sample IORDY. The IORDY sample point is specified by the ISP field in this register.

Bit	Description
4	**Fast Timing Bank Drive Select 1 (TIME1).** When TIME1=0, accesses to the data port of the enabled I/O address range use the 16-bit compatible timing PCI local bus path. When TIME1=1 and the currently selected drive (via a copy of bit 4 of 1x6h) is Drive 1, accesses to the data port of the enabled I/O address range use the fast timing bank PCI local bus IDE path. Accesses to the data port use fast timing only if bit 7 of this register (DTE1) is zero. Accesses to all non-data ports of the enabled I/O address range use the 8-bit compatible timing PCI local bus path.
3	**DMA Timing Enable Only (DTE0).** When DTE0=1, fast timing mode is enabled for DMA data transfers for drive 0. Note that PIO transfers to the IDE data port still run in compatible timing.
2	**Prefetch and Posting Enable (PPE0).** 1=Enable; 0=Disable. When enabled, prefetch and posting to the IDE data port is enabled for drive 0.
1	**IORDY Sample Point Enable Drive Select 0 (IE0).** When IE0=0, IORDY sampling is disabled for Drive 0. The internal IORDY signal is forced asserted guaranteeing that IORDY is sampled asserted at the first sample point as specified by the ISP field in this register. When IE0=1 and the currently selected drive (via a copy of bit 4 of 1x6h) is Drive 0, all accesses to the enabled I/O address range sample IORDY. The IORDY sample point is specified by the ISP field in this register.
0	**Fast Timing Bank Drive Select 0 (TIME0).** When TIME0=0, accesses to the data port of the enabled I/O address range uses the 16-bit compatible timing PCI local bus path. When TIME0=1 and the currently selected drive (via a copy of bit 4 of 1x6h) is Drive 0, accesses to the data port of the enabled I/O address range use the fast timing bank PCI local bus IDE path. Accesses to the data port use fast timing only if bit 3 of this register (DTE0) is 0. Accesses to all non-data ports of the enabled I/O address range use the 8-bit compatible timing PCI local bus path.

2.3.11. SIDETIM—SLAVE IDE TIMING REGISTER (Function 1) (PIIX3 Only)

Address Offset: 44h
Default Value: 00h
Attribute: Read / Write Only

This register controls the PIIX3's IDE interface and selects the timing characteristics for the slave drives on each IDE channel. This allows for programming of independent operating modes for each IDE agent. This register has no affect unless the SITRE bit is enabled in the IDETIM Register.

Bit	Description
7:6	**Secondary Drive 1 IORDY Sample Point (SISP1).** This field selects the number of clocks between DIOx# assertion and the first IORDY sample point for the slave drive on the secondary channel. **Bits[7:6]** — **Number Of Clocks** 00 — 5 01 — 4 10 — 3 11 — 2

Bit	Description
5:4	**Secondary Drive 1 Recovery Time (SRTC1).** This field selects the minimum number of clocks between the last IORDY# sample point and the DIOx# strobe of the next cycle for the slave drive on the secondary channel. **Bits[5:4] — Number Of Clocks** 00 — 4 01 — 3 10 — 2 11 — 1
3:2	**Primary Drive 1 IORDY Sample Point (PISP1).** This field selects the number of clocks between DIOx# assertion and the first IORDY sample point for the slave drive on the primary channel. **Bits[3:2] — Number Of Clocks** 00 — 5 01 — 4 10 — 3 11 — 2
1:0	**Primary Drive 1 Recovery Time (PRTC1).** This field selects the minimum number of clocks between the last IORDY# sample point and the DIOx# strobe of the next cycle for the slave drive on the primary channel. **Bits[1:0] — Number Of Clocks** 00 — 4 01 — 3 10 — 2 11 — 1

2.4. PCI Configuration Registers—Universal Serial Bus (Function 2) (PIIX3 Only)

2.4.1. VID—VENDOR IDENTIFICATION REGISTER (Function 2) (PIIX3)

Address Offset: 00–01h
Default Value: 8086h
Attribute: Read Only

The VID Register contains the vendor identification number. This register, along with the Device Identification Register, uniquely identifies any PCI device. Writes to this register have no effect.

Bit	Description
15:0	**Vendor Identification Number.** This is a 16-bit value assigned to Intel.

2.4.2. DID—DEVICE IDENTIFICATION REGISTER (Function 2) (PIIX3)

Address Offset: 02–03h
Default Value: 7020h
Attribute: Read Only

The DID Register contains the device identification number. This register, along with the VID Register, define the PIIX. Writes to this register have no effect.

Bit	Description
15:0	**Device Identification Number.** This is a 16-bit value assigned to the PIIX3.

2.4.3. PCICMD—COMMAND REGISTER (Function 2) (PIIX3)

Address Offset: 04–05h
Default Value: 00h
Attribute: Read/Write

This register controls access to the I/O space registers.

Bit	Description
15:10	**Reserved.** Read 0.
9	**Fast Back to Back Enable (FBE). (Not Implemented)** This bit is hardwired to 0.
8:5	**Reserved.** Read as 0.
4	**Memory Write and Invalidate Enable (MWI). (Not Implemented)** This bit is hardwired to 0.
3	**Special Cycle Enable (SCE). (Not Implemented)** This bit is hardwired to 0.
2	**Bus Master Enable (BME).** This bit controls the PIIX3's ability to act as a master on the PCI bus for the host controller transfers. A value of 0 disables the device from generating PCI accesses. A value of 1 allows the device to behave as a USB host controller bus master. This bit must be set to 1 before serial bus transactions can start.
1	**Memory Space Enable (MSE). (Not Implemented)** This bit is hardwired to 0.
0	**I/O Space Enable (IOSE).** 1=Enable. 0=Disable. This bit controls the access to the I/O space registers. If this bit is set, access to the host controller IO registers is enabled. The base register for the I/O registers must be programmed before this bit is set.

2.4.4. DS—DEVICE STATUS REGISTER (Function 2) (PIIX3)

Address Offset: 06–07h
Default Value: 0280h
Attribute: Read/Write

DSR is a 16-bit status register that reports the occurrence of a PCI master-abort by the USB HC module or a PCI target-abort when the Serial Bus module is a master. The register also indicates the USB HC module DEVSEL# signal timing that is hardwired in the USB HC module. The DS fields are shown in the table below.

Bit	Description
15	**Detected Parity Error (PERR). (Not Implemented)** Read as 0.
14	**SERR# Status (SERRS). (Not Implemented)** Read as 0.
13	**Master-Abort Status (MAS)—R/WC:**. When the Serial Bus module generates a master-abort, MA is set to a 1. Software sets MA to 0 by writing a 1 to this bit.
12	**Received Target-Abort Status (RTA)—R/WC.** When the Serial Bus module is a master on the PCI Bus and receives a target-abort, this bit is set to a 1. Software resets RTA to 0 by writing a 1 to this bit.
11	**Signaled Target-Abort Status (STA)—R/WC.** This bit is set when the Serial Bus module function is targeted with a transaction that the Serial Bus module terminates with a target abort. Software resets STA to 0 by writing a 1 to this bit.
10:9	**DEVSEL# Timing Status (DEVT) —RO.** This 2-bit field defines the timing for DEVSEL# assertion. These read only bits indicate the PIIX3's DEVSEL# timing when performing a positive decode. Since the PIIX3 always generate the DEVSEL# with medium timing, DEVT=01. This DEVSEL# timing does not include Configuration cycles.
8	**Data Parity Detected (DPD).** (Not Implemented). Read as 0.
7	**Fast Back to Back Capable (FBC)—RO.** Hardwired to 1. This bit indicates to the PCI Master that Serial Bus module as a target is capable of accepting fast back-to-back transactions.
6:0	**Reserved**. Read as 0's.

2.4.5. RID—REVISION IDENTIFICATION REGISTER (Function 2) (PIIX3)

Address Offset: 08h
Default Value: Refer to applicable specification update document
Attribute: Read Only

This 8-bit register contains device stepping information. Writes to this register have no effect.

Bit	Description
7:0	**Revision ID Byte.** The register is hardwired to the default value during manufacturing.

2.4.6. CLASSC—CLASS CODE REGISTER (Function 2) (PIIX3)

Address Offset: 09–0Bh
Default Value: 010180h
Attribute: Read Only

This register contains the device programming interface information related to the Sub-Class Code and Base Class Code definition for the PIIX3 (function 2). This register also identifies the Base Class Code and the function sub-class in relation to the Base Class Code.

Bit	Description
23:16	**Base Class Code (BASEC).** 0Ch=Universal Serial Bus controller.
15:8	**Sub-Class Code (SCC)**. 03h=Universal Serial Bus Host Controller..
7:0	**Programming Interface (PI).** 00h=Capable of Universal Serial Bus operation.

2.4.7. MLT—MASTER LATENCY TIMER REGISTER (Function 2) (PIIX3)

Address Offset: 0Dh
Default Value: 00h
Attribute: Read/Write

MLT is an 8-bit register that controls the amount of time (in terms of PCI clocks) the USB module can do transactions on the PCI bus. The count value is an 8-bit quantity, however MLT[3:0] are reserved and assumed to be 0 when determining the count value. MLT is used when the USB module becomes the PCI bus master and is cleared and suspended when PIIX3 is not asserting FRAME#. When PIIX3 asserts FRAME#, the counter is enabled and begins counting. If the serial bus module finishes its transaction before count is expired the MLT value is ignored. If the count expires before the transaction completes, PIIX3 initiates a transaction termination as soon as the current transaction is completed.. The number of clocks programmed in the MLT represents the guaranteed time slice (measured in PCI clocks) allotted to PIIX3, after which it must surrender the bus as soon as the current transaction is completed. The default value of MLT is 00h.

Bit	Description
7:4	**Master Latency Counter Value.** PIIX3 initiated PCI cycles (including multiple transactions) can last indefinitely as long as PHLDA# remains active. However, if PHLDA# is negated after a transaction is initiated, PIIX3 limits the duration of the transactions to the number of PCI bus clocks specified by this field.
3:0	**Reserved.**

2.4.8. HEDT—HEADER TYPE REGISTER (Function 2) (PIIX3)

Address Offset: 0Eh
Default Value: 00h
Attribute: Read Only

This register identifies the Serial Bus module as a single function device.

Bit	Description
7:0	**Device Type (DEVICET).** 00. Multi-function device capability for PIIX/PIIX3 is defined by the HEDT register in Function 0.

2.4.9. BASEADD—I/O SPACE BASE ADDRESS (Function 2) (PIIX3)

Address Offset: 20–23h
Default Value: 00h
Attribute: Read/Write

This register contains the base address of the USB I/O Registers.

Bit	Description
31:16	**Reserved.** Hardwired to 0s. Must be written as 0s.
15:5	**Index Register Base Address.** Bits [15:5] correspond to I/O address signals AD [15:5], respectively.
4:1	**Reserved.** Read as 0.
0	**Resource Type Indicator (RTE)—RO.** This bit is hardwired to 1 indicating that the base address field in this register maps to I/O space.

2.4.10. IL—Interrupt Line Register (Function 2) (PIIX3)

Address Offset: 3Ch
Default Value: 00h
Attribute: Read/Write

Software programs this register with interrupt information concerning the Universal Serial Bus.

Bit	Description
7:0	**Interrupt Line.** . The value in this register has no affect on PIIX3 hardware operations.

2.4.11. INTRP—INTERRUPT PIN (Function 2) (PIIX3)

Address Offset: 3Dh
Default Value: 04h
Attribute: Read only

This register indicates which PCI interrupt pin is used for the Universal Serial Bus module interrupt. The USB interrupt is internally ORed to the interrupt controller with the PIRQD# signal.

Bit	Description
7:3	**Reserved.**
2:0	**Serial Bus Module Interrupt Routing.** This field is hardwired to 100b to select PIRQD#.

2.4.12. SBRNUM—SERIAL BUS RELEASE NUMBER (Function 2) (PIIX3)

Address Offset: 60h
Default Value: 00h
Attribute: Read only

This register contains the release of the Universal Serial Bus Specification with which this Universal Serial Bus Host Controller module is compliant.

Bit	Description
7:0	**Serial Bus Specification Release Number.** All other combinations are reserved. **Bits[7:0] Release Number** 00h Pre-release 1.0 10h Release 1.0

2.4.13. MSTAT—MISCELLANEOUS STATUS REGISTER (Function 2) (PIIX3)

Address Offset: 6A–6Bh
Default Value: 01h
Attribute: Read/Write

This 16-bit register provides miscellaneous control and status for the USB function..

Bit	Description
15:1	**Reserved**
0	**USB Clock Selection.** 1=48 MHz. 0=24 MHz.

2.4.14. LEGSUP—LEGACY SUPPORT REGISTER (FUNCTION 2) (PIIX3)

PCI Address Offset: C0–C1h
Default: 2000h
Attribute: Read/Write Clear

This register provides control and status capability for the legacy keyboard and mouse functions.

Bit	Description
15	**End OF A20GATE Pass-through Status (A20PTS)—R/WC.** This bit is set to 1 to indicate that the A20GATE pass-through sequence has ended. Software must use the enable bits to determine the exact cause of an SMI#. Software cleats this bit by writing a 1 to it.
14	**Reserved.**
13	**USB PIRQ Enable (USBPIRQDEN) —R/W.** 1=USB interrupt is routed to PIRQD (default). 0=USB interrupt does not route to PIRQD. This bit prevents the USB controller from generating an interrupt. Note that PIIX3 will typically be configured to generate an SMI using bit 4 of this register. Default to 1 for compatibility with older USB software.
12	**USB IRQ Status (USBIRQS) —RO.** PIIX3 sets this bit to 1 to indicate that the USB IRQ is active. Software must use the enable bits to determine the exact cause of an SMI#. Writing a 1 to this bit will have no effect. Software must clear the IRQ via the USB controller.
11	**Trap By 64h Write Status (TBY64W)—R/WC.** PIIX3 sets this bit to 1 to indicate that a write to port 64h occurred. Software must use the enable bits to determine the exact cause of an SMI#. Software clears this bit by writing a 1 to it.
10	**Trap By 64h Read Status (TBY64R)—R/WC.** PIIX3 sets this bit to 1 to indicate that a read to port 64h occurred. Software must use the enable bits to determine the exact cause of an SMI#. Software clears this bit by writing a 1 to it.
9	**Trap By 60h Write Status (TBY60W)—R/WC.** PIIX3 sets this bit to 1 to indicate that a write to port 60h occurred. Software must use the enable bits to determine the exact cause of an SMI#. Software clears this bit by writing a 1 to it.
8	**Trap By 60h Read Status (TBY60R)—R/WC.** PIIX3 sets this bit to 1 to indicate that a read to port 60h occurred. Software must use the enable bits to determine the exact cause of an SMI#. Software clears this bit by writing a 1 to it.
7	**SMI At End Of Pass-through Enable (SMIEPTE)—R/W.** 1=Enable the generation of an SMI when the A20GATE pass-through sequence has ended. 0=Disable (default). This may be required if an SMI is generated by a USB interrupt in the middle of an A20GATE pass-through sequence and needs to be serviced later.
6	**Pass-through Status (PSS)—RO.** 1=A20GATE pass-through sequence is currently in progress. 0=Not currently executing the A20GATE pass-through sequence (default). This bit indicates that the host controller is executing the A20GATE pass-through sequence. If software needs to reset this bit, it should set Bit 5 to 0 causing the host controller to immediately end the A20GATE pass-through sequence.

Bit	Description
5	**A20Gate Pass-through Enable (A20PTEN)**—**R/W.** 1=Enable A20GATE pass-through sequence. 0=Disable (default). When enabled, the logic will pass-through the following A20GATE command sequence: **Cycle** **Address** **Data** Write 64h D1h (1 or more) (Starts the Sequence) Write 60h xxh Read 64h N/A (0 or more) Write 64h FFh (Standard End of A20GATE Pass-through Sequence) Any deviation in the above sequence causes the host controller to immediately exit the sequence and return to standard operation, performing an I/O trap and generating an SMI# if appropriate enable bits are set. When enabled, SMI# is not generated during the sequence, even if the various enable bits are set. Note that during a pass-through sequence, the above status bits are not set for the I/O accesses that are part of the sequence.
4	**Trap/SMI ON IRQ Enable (USBSMIEN)**—**R/W.** 1=Enable SMI# generation on USB IRQ. 0=Disable (default).
3	**Trap/SMI On 64h Write Enable (64WEN)**—**R/W.** 1=Enable I/O Trap and SMI# generation on port 64h write. 0=Disable (default).
2	**Trap/SMI On 64h Read Enable (64REN)**—**R/W.** 1=Enable I/O Trap and SMI# generation on port 64h read. 0=Disable (default).
1	**Trap/SMI On 60h Write Enable (60WEN)**—**R/W.** 1=Enable I/O Trap and SMI# generation on port 60h write. 0=Disable (default).
0	**Trap/SMI On 60h Read Enable (60REN)**—**R/W.** 1 = Enable I/O Trap and SMI# generation on port 60h read. 0=Disable (default).

2.5. ISA-Compatible Registers

The ISA-Compatible registers contain the DMA, timer/counter, and interrupt registers. This group also contains the X-Bus, coprocessor, NMI, and reset registers.

2.5.1. DMA REGISTERS

The PIIX/PIIX3 contains DMA circuitry that incorporates the functionality of two 82C37 DMA controllers (DMA1 and DMA2). The DMA registers control the operation of the DMA controllers and are all accessible from the Host CPU via the PCI Bus interface. In addition, some of the registers are accessed from the ISA Bus via ISA I/O space. Unless otherwise stated, a CPURST sets each register to its default value.

2.5.1.1. DCOM—DMA Command Register

I/O Address: Channels 0-3—08h; Channels 4-7—0D0h
Default Value: 00h (CPURST or Master Clear)
Attribute: Write Only

This 8-bit register controls the configuration of the DMA. Note that disabling channels 4-7 also disables channels 0-3, since channels 0-3 are cascaded onto channel 4.

Bit	Description
7	**DACK# ACTIVE Level (DACK#[3:0, (7:5)]).** 1=Active high; 0=Active low.
6	**DREQ Sense Assert Level (DREQ[3:0, (7:5)]).** 1=Active Low; 0=Active high.
5	**Reserved.** Must be 0.
4	**DMA Group Arbitration Priority.** 1=Rotating priority; 0=Fixed priority
3	**Reserved**. Must be 0
2	**DMA Channel Group Enable.** 1=Disable; 0 = Enable.
1:0	**Reserved**. Must be 0.

2.5.1.2. DCM—DMA Channel Mode Register

I/O Address: Channels 0-3=0Bh; Channels 4-7=0D6h
Default Value: Bits[7:2]=0; Bits[1:0]=undefined (CPURST or Master Clear)
Attribute: Write Only

Each channel has a 6-bit DMA Channel Mode Register. The Channel Mode Registers provide control over DMA transfer type, transfer mode, address increment/decrement, and autoinitialization.

Bit	Description
7:6	**DMA Transfer Mode.** Each DMA channel can be programmed in one of four different modes: **Bits[7:6] Transfer Mode** 00 Demand mode 01 Single mode 10 Block mode 11 Cascade mode

Bit	Description
5	**Address Increment/Decrement Select.** 0=Increment; 1=Decrement.
4	**Autoinitialize Enable.** 1=Enable; 0=Disable.
3:2	**DMA Transfer Type.** When Bits [7:6]=11, the transfer type bits are irrelevant. **Bits[3:2] Transfer Type** 00 Verify transfer 01 Write transfer 10 Read transfer 11 Illegal
1:0	**DMA Channel Select.** Bits [1:0] select the DMA Channel Mode Register written to by bits [7:2]. **Bits[1:0] Channel** 00 Channel 0 (4) 01 Channel 1 (5) 10 Channel 2 (6) 11 Channel 3 (7)

2.5.1.3. DR—DMA Request Register

I/O Address: Channels 0-3—09h; Channels 4-7—0D2h
Default Value: Bits[1:0]=undefined; Bits[7:2]=0 (CPURST or Master Clear)
Attribute: Write Only

The Request Register is used by software to initiate a DMA request. The DMA responds to the software request as though DREQx is asserted. These requests are non-maskable and subject to prioritization by the priority encoder network. For a software request, the channel must be in Block Mode. The Request Register status for DMA1 and DMA2 is output on bits [7:4] of a Status Register read.

Bit	Description
7:3	**Reserved**. Must be 0
2	**DMA Channel Service Request.** 0=Resets the individual software DMA channel request bit. 1=Sets the request bit. Generation of a TC also sets this bit to 0.
1:0	**DMA Channel Select.** Bits [1:0] select the DMA channel mode register to program with bit 2. **Bits[1:0] Channel** 00 Channel 0 01 Channel 1 (5) 10 Channel 2 (6) 11 Channel 3 (7)

2.5.1.4. Mask Register—Write Single Mask Bit

I/O Address: Channels 0-3—0Ah; Channels 4-7—0D4h
Default Value: Bits[1:0]=undefined; Bit 2=1; Bits[7:3]=0 (CPURST or a Master Clear)
Attribute: Write Only

A channel's mask bit is automatically set when the Current Byte/Word Count Register reaches terminal count (unless the channel is programmed for autoinitialization). Setting the entire register disables all DMA requests until a clear mask register instruction allows them to occur. This instruction format is similar to the format used with the DMA Request Register. Masking DMA channel 4 (DMA controller 2, channel 0) also masks DMA channels [3:0].

Bit	Description
7:3	**Reserved**. Must be 0.
2	**Channel Mask Select.** 1=Disable DREQ for the selected channel. 0=Enable DREQ for the selected channel.
1:0	**DMA Channel Select.** Bits [1:0] select the DMA Channel Mode Register for bit 2. **Bits[1:0] Channel** 00 Channel 0 (4) 01 Channel 1 (5) 10 Channel 2 (6) 11 Channel 3 (7)

2.5.1.5. Mask Register—Write All Mask Bits

I/O Address: Channels 0-3—0Fh; Channels 4-7—0DEh
Default Value: Bit[3:0]=1; Bit[7:4]=0 (CPURST or Master Clear)
Attribute: Read/Write

A channel's mask bit is automatically set to 1 when the Current Byte/Word Count Register reaches terminal count (unless the channel is programmed for autoinitialization). Setting bits [3:0] to 1 disables all DMA requests until a clear mask register instruction enables the requests. Note that, masking DMA channel 4 (DMA controller 2, channel 0), masks DMA channels [3:0]. Also Note that, Masking DMA controller 2 with a write to port 0DEh also masks DREQ assertions from DMA controller 1.

Bit	Description
7:4	**Reserved**. Must be 0.
3:0	**Channel Mask Bits.** 1=Disable the corresponding DREQ(s); 0=Enable the corresponding DREQ(s). **Bit Channel** 0 0 (4) 1 1 (5) 2 2 (6) 3 3 (7)

2.5.1.6. DS—DMA Status Register

I/O Address: Channels 0-3—08h; Channels 4-7—0D0h
Default Value: 00h
Attribute: Read Only

Each DMA controller has a read-only DMA Status Register that indicates which channels have reached terminal count and which channels have a pending DMA request.

Bit	Description
7:4	**Channel Request Status.** When a valid DMA request is pending for a channel (on its DREQ signal line), the corresponding bit is set to 1. When a DMA request is not pending for a particular channel, the corresponding bit is set to 0. The source of the DREQ may be hardware or a software request. Note that channel 4 does not have DREQ or DACK lines, so the response for a read of DMA2 status for channel 4 is irrelevant. **Bit — Channel** 4 — 0 5 — 1 (5) 6 — 2 (6) 7 — 3 (7)
3:0	**Channel Terminal Count Status.** 1=TC is reached; 0=TC is not reached. **Bit — Channel** 0 — 0 1 — 1 (5) 2 — 2 (6) 3 — 3 (7)

2.5.1.7. DMA Base And Current Address Registers (8237 Compatible Segment)

I/O Address: DMA Channel 0—000h; DMA Channel 1—002h; DMA Channel 2—004h; DMA Channel 3—006h; DMA Channel 4—0C0h; DMA Channel 5—0C4h; DMA Channel 6—0C8h; DMA Channel 7—0CCh
Default Value: XXXXh (CPURST or Master Clear)
Attribute: Read/Write

This Register works in conjunction with the Low Page Register. After an autoinitialization, this register retains the original programmed value. Autoinitialize takes place after a TC. The address register is automatically incremented or decremented after each transfer. This register is read/written in successive 8-bit bytes. The programmer must issue the "Clear Byte Pointer Flip-Flop" command to reset the internal byte pointer and correctly align the write prior to programming the Current Address Register. Autoinitialize takes place only after a TC.

Bit	Description
15:0	**Base and Current Address [15:0].** These bits represent address bits [15:0] used when forming the 24-bit address for DMA transfers.

2.5.1.8. DMA Base And Current Byte/Word Count Registers (Compatible Segment)

I/O Address:	DMA Channel 0—001h	DMA Channel 4—0C2h
	DMA Channel 1—003h	DMA Channel 5—0C6h
	DMA Channel 2—005h	DMA Channel 6—0CAh
	DMA Channel 3—007h	DMA Channel 7—0CEh
Default Value:	XXXXh (CPURST or Master Clear)	
Attribute:	Read/Write	

This register determines the number of transfers to be performed. The actual number of transfers is one more than the number programmed in the Current Byte/Word Count Register When the value in the register is decremented from zero to FFFFh, a TC is generated. Autoinitialize can only occur when a TC occurs. If it is not autoinitialized, this register has a count of FFFFh after TC.

For transfers to/from an 8-bit I/O, the Byte/Word count indicates the number of bytes to be transferred. This applies to DMA channels 0-3. For transfers to/from a 16-bit I/O, with shifted address, the Byte/Word count indicates the number of 16-bit words to be transferred. This applies to DMA channels 5-7.

Bit	Description
15:0	**Base and Current Byte/ Word Count.** These bits represent the 16 byte/word count bits used when counting down a DMA transfer.

2.5.1.9. DMA Memory Low Page Registers

I/O Address:	DMA Channel 0—087h	DMA Channel 5—08Bh
	DMA Channel 1—083h	DMA Channel 6—089h
	DMA Channel 2—081h	DMA Channel 7—08Ah
	DMA Channel 3—082h	
Default Value:	XXh (CPURST or Master Clear)	
Attribute:	Read/Write	

This register works in conjunction with the Current Address Register. After an autoinitialization, this register retains the original programmed value. Autoinitialize takes place after a TC.

Bit	Description
7:0	**DMA Low Page [23:16].** These bits represent address bits [23:16] of the 24-bit DMA address

2.5.1.10. DMA Clear Byte Pointer Register

I/O Address: Channels 0-3—00Ch; Channels 4-7—0D8h
Default Value: All bits undefined
Attribute: Write Only

Writing to this register executes the Clear Byte Pointer Command. This command is executed prior to reading/writing a new address or word count to the DMA. The command initializes the byte pointer flip-flop to a known state so that subsequent accesses to register contents address upper and lower bytes in the correct sequence. The Clear Byte Pointer Command (or CPURST or the Master Clear Command) clears the internal latch used to address the upper or lower byte of the 16-bit Address and Word Count Registers.

Bit	Description
7:0	**Clear Byte Pointer.** No specific pattern. Command enabled with a write to the I/O port address.

2.5.1.11. DMC—DMA Master Clear Register

I/O Address: Channel 0-3—00Dh; Channel 4-7—0DAh
Default Value: All bits undefined
Attribute: Write Only

This software instruction has the same effect as the hardware Reset.

Bit	Description
7:0	**Master Clear.** No specific pattern. Command enabled with a write to the I/O port address

2.5.1.12. DCLM—DMA Clear Mask Register

I/O Address: Channel 0-3—00Eh; Channel 4-7—0DCh
Default Value: All bits undefined
Attribute: Write Only

This command clears the mask bits of all four channels, enabling them to accept DMA requests.

Bit	Description
7:0	**Clear Mask Register.** No specific pattern. Command enabled with a write to the I/O port address.

2.5.2. TIMER/COUNTER REGISTER DESCRIPTION

2.5.2.1. TCW—Timer Control Word Register

I/O Address: 043h
Default Value: All bits undefined
Attribute: Write Only

The Timer Control Word Register specifies the counter selection, the operating mode, the counter byte programming order and size of the count value, and whether the counter counts down in a 16-bit or binary-

coded decimal (BCD) format. After writing the control word, a new count can be written at any time. The new value takes effect according to the programmed mode.

Bit	Description
7:6	**Counter Select.** The Read Back Command is selected when bits[7:6] are both 1. **Bit[7:6] Function** 00 Counter 0 select 01 Counter 1 select 10 Counter 2 select 11 Read Back Command
5:4	**Read/Write Select.** The Counter Latch Command is selected when bits[5:4] are both 0. **Bit[5:4] Function** 00 Counter Latch Command 01 R/W Least Significant Byte 10 R/W Most Significant Byte 11 R/W LSB then MSB
3:1	**Counter Mode Selection.** Bits [3:1] select one of six possible counter modes. **Bit[3:1] Mode Function** 000 0 Out signal on end of count (=0) 001 1 Hardware retriggerable one-shot X10 2 Rate generator (divide by n counter) X11 3 Square wave output 100 4 Software triggered strobe 101 5 Hardware triggered strobe
0	**Binary/BCD Countdown Select.** 0=Binary countdown. The largest possible binary count is 2^{16}. 1=Bbinary coded decimal (BCD) count is used. The largest BCD count allowed is 10^4.

Read Back Command

The Read Back Command is used to determine the count value, programmed mode, and current states of the OUT pin and Null count flag of the selected counter or counters. The Read Back Command is written to the Timer Control Word Register which latches the current states of the above mentioned variables. The value of the counter and its status may then be read by I/O access to the counter address. Note that The Timer Counter Register bit definitions are different during the Read Back Command than for a normal Timer Counter Register write.

Bit	Description
7:6	**Read Back Command.** When bits[7:6]=11, the Read Back Command is selected during a write to the Timer Control Word Register. Following the Read Back Command, I/O reads from the selected counter's I/O addresses produce the current latch status, the current latched count, or both if bits 4 and 5 are both 0.
5	**Latch Count of Selected Counters.** When bit 5=0, the current count value of the selected counters will be latched. When bit 5=1, the count will not be latched.
4	**Latch Status of Selected Counters.** When bit 4=0, the status of the selected counters will be latched. When bit 4=1, the status will not be latched. The status byte format is described in Section 4.3.3, Interval Timer Status Byte Format Register.

Bit	Description
3	**Counter 2 Select.** When bit 3=1, Counter 2 is selected for the latch command selected with bits 4 and 5. When bit 3=0, status and/or count will not be latched.
2	**Counter 1 Select.** When bit 2=1, Counter 1 is selected for the latch command selected with bits 4 and 5. When bit 2=0, status and/or count will not be latched.
1	**Counter 0 Select.** When bit 1=1, Counter 0 is selected for the latch command selected with bits 4 and 5. When bit 1=0, status and/or count will not be latched.
0	**Reserved**. Must be 0.

Counter Latch Command

The Counter Latch Command latches the current count value at the time the command is received. If a Counter is latched once and then, some time later, latched again before the count is read, the second Counter Latch Command is ignored. The count read will be the count at the time the first Counter Latch Command was issued. If the counter is programmed for two byte counts, two bytes must be read. The two bytes do not have to be read successively (read, write, or programming operations for other counters may be inserted between the reads). Note that the Timer Counter Register bit definitions are different during the Counter Latch Command than for a normal Timer Counter Register write. Note that, If a counter is programmed to read/write two-byte counts, a program must not transfer control between reading the first and second byte to another routine that also reads from that same counter. Otherwise, an incorrect count will be read.

Bit	Description
7:6	**Counter Selection.** Bits 6 and 7 are used to select the counter for latching. **Bit[7:6]** **Function** 00 latch counter 0 select 01 latch counter 1 select 10 latch counter 2 select 11 Read Back Command select
5:4	**Counter Latch Command.** When bits[5:4]=00, the Counter Latch Command is selected during a write to the Timer Control Word Register. Following the Counter Latch Command, I/O reads from the selected counter's I/O addresses produce the current latched count.
3:0	**Reserved**. Must be 0.

2.5.2.2. Interval Timer Status Byte Format Register

I/O Address: Counter 0—040h; Counter 1—041h; Counter 2—042h
Default Value: Bits[6:0]=X; Bit 7=0
Attribute: Read Only

Each counter's status byte can be read following an Interval Timer Read Back Command. If latch status is chosen (bit 4=0, Read Back Command) as a read back option for a given counter, the next read from the counter's Counter Access Ports Register returns the status byte.

Bit	Description
7	**Counter OUT Pin State.** 1=Pin is 1; 0=Pin is 0.

Bit	Description
6	**Count Register Status.** This bit indicates when the last count written to the Count Register (CR) has been loaded into the counting element (CE). 0=Count has been transferred from CR to CE and is available for reading. 1=Count has not been transferred from CR to CE and is not yet available for reading.
5:4	**Read/Write Selection Status.** Bits[5:4] reflect the read/write selection made through bits[5:4] of the Control Register. **Bit[5:4] Function** 00 Counter Latch Command 01 R/W Least Significant Byte (LSB) 10 R/W Most Significant Byte (MSB) 11 R/W LSB then MSB
3:1	**Mode Selection Status.** Bits[3:1] return the counter mode programming. **Bit[3:1] Mode Selected Bit[3:1] Mode Selected** 000 0 X11 3 001 1 100 4 X10 2 101 5
0	**Countdown Type Status.** 0=Binary countdown; 1=Binary coded decimal (BCD) countdown.

2.5.2.3. Counter Access Ports Register

I/O Address: Counter 0—040h; Counter 1—041h; Counter 2—042h
Default Value: All bits undefined
Attribute: Read/Write

Each of these I/O ports is used for writing count values to the Count Registers; reading the current count value from the counter by either an I/O read, after a counter-latch command, or after a Read Back Command; and reading the status byte following a Read Back Command.

Bit	Description
7:0	**Counter Port bit[x]**. Each counter I/O port address is used to program the 16-bit Count Register. The order of programming, either LSB only, MSB only, or LSB then MSB, is defined with the Interval Counter Control Register. The counter I/O port is also used to read the current count from the Count Register and return counter programming status following a Read Back Command.

2.5.3. INTERRUPT CONTROLLER REGISTERS

The PIIX/PIIX3 contains an ISA-Compatible interrupt controller that incorporates the functionality of two 82C59 interrupt controllers. The interrupt registers control the operation of the interrupt controller.

2.5.3.1. ICW1—Initialization Command Word 1 Register

I/O Address: INT CNTRL-1—020h; INT CNTRL-2—0A0h
Default Value: All bits undefined
Attribute: Write Only

A write to Initialization Command Word 1 starts the interrupt controller initialization sequence. Addresses 020h and 0A0h are referred to as the base addresses of CNTRL-1 and CNTRL-2, respectively. An I/O write to the CNTRL-1 or CNTRL-2 base address with bit 4 equal to 1 is interpreted as ICW1. For PIIX-based ISA systems, three I/O writes to "base address + 1" must follow the ICW1. The first write to "base address + 1" performs ICW2, the second write performs ICW3, and the third write performs ICW4.

ICW1 starts the initialization sequence during which the following automatically occur:

1. The Interrupt Mask register is cleared.
2. IRQ7 input is assigned priority 7.
3. The slave mode address is set to 7.
4. Special Mask Mode is cleared and Status Read is set to IRR.
5. If IC4 was set to 0, then all functions selected by ICW4 are set to 0. However, ICW4 must be programmed in the PIIX/PIIX3 implementation of this interrupt controller, and IC4 must be set to a 1.

Bit	Description
7:5	**ICW/OCW select.** These bits should be 000 when programming the PIIX.
4	**ICW/OCW select.** Bit 4 must be a 1 to select ICW1. After the fixed initialization sequence to ICW1, ICW2, ICW3, and ICW4, the controller base address is used to write to OCW2 and OCW3. Bit 4 is a 0 on writes to these registers. A 1 on this bit at any time will force the interrupt controller to interpret the write as an ICW1. The controller will then expect to see ICW2, ICW3, and ICW4.
3	**Edge/Level Bank Select (LTIM).** This bit is disabled. Its function is replaced by the Edge/Level Triggered Control (ELCR) Registers.
2	**ADI.** Ignored for the PIIX.
1	**Single or Cascade (SNGL).** This bit must be programmed to a 0.
0	**ICW4 Write Required (IC4).** This bit must be set to a 1.

2.5.3.2. ICW2—Initialization Command Word 2 Register

I/O Address: INT CNTRL-1—021h; INT CNTRL-2—0A1h
Default Value: All bits undefined
Attribute: Write Only

ICW2 is used to initialize the interrupt controller with the five most significant bits of the interrupt vector address.

Bit	Description
7:3	**Interrupt Vector Base Address.** Bits [7:3] define the base address in the interrupt vector table for the interrupt routines associated with each interrupt request level input.
2:0	**Interrupt Request Level.** Must be programmed to all 0s.

2.5.3.3. ICW3—Initialization Command Word 3 Register

I/O Address: INT CNTRL-1—021h
Default Value: All bits undefined
Attribute: Write Only

The meaning of ICW3 differs between CNTRL-1 and CNTRL-2. On CNTRL-1, the master controller, ICW3 indicates which CNTRL-1 IRQ line physically connects the INTR output of CNTRL-2 to CNTRL-1.

Bit	Description
7:3	**Reserved.** Must be programmed to all 0s.
2	**Cascaded Mode Enable.** This bit must be programmed to 1 selecting cascade mode.
1:0	**Reserved.** Must be programmed to all 0s.

2.5.3.4. ICW3—Initialization Command Word 3 Register

I/O Address: INT CNTRL-2—0A1h
Default Value: All bits undefined
Attribute: Write Only

On CNTRL-2 (the slave controller), ICW3 is the slave identification code broadcast by CNTRL-1.

Bit	Description
7:3	**Reserved.** Must be programmed to all 0s.
2:0	**Slave Identification Code.** Must be programmed to 010b.

2.5.3.5. ICW4—Initialization Command Word 4 Register

I/O Address: INT CNTRL-1—021h; INT CNTRL-2—0A1h
Default Value: 01h
Attribute: Write Only

Both PIIX/PIIX3 interrupt controllers must have ICW4 programmed as part of their initialization sequence.

Bit	Description
7:5	**Reserved.** Must be programmed to all 0s.
4	**Special Fully Nested Mode (SFNM).** Bit 4, SFNM, should normally be disabled by writing a 0 to this bit. If SFNM=1, the special fully nested mode is programmed.
3	**Buffered mode (BUF).** Must be programmed to 0 selecting non-buffered mode.
2	**Master/Slave in Buffered Mode.** Should always be programmed to 0. Bit not used.
1	**AEOI (Automatic End of Interrupt).** This bit should normally be programmed to 0. This is the normal end of interrupt. If this bit is 1, the automatic end of interrupt mode is programmed.
0	**Microprocessor Mode.** Must be programmed to 1 indicating an Intel Architecture-based system.

2.5.3.6. OCW1—Operational Control Word 1 Register

I/O Address: INT CNTRL-1—021h; INT CNTRL-2—0A1h
Default Value: 00h
Attribute: Read/Write

OCW1 sets and clears the mask bits in the Interrupt Mask Register (IMR). Each interrupt request line may be selectively masked or unmasked any time after initialization. The IMR stores the interrupt line mask bits. The IMR operates on the IRR. Masking of a higher priority input does not affect the interrupt request lines of lower priority. Unlike status reads of the ISR and IRR, for reading the IMR, no OCW3 is needed. The output data bus contains the IMR when an I/O read is active and the I/O address is 021h or 0A1h (OCW1). All writes to OCW1 must occur following the ICW1-ICW4 initialization sequence, since the same I/O ports are used for OCW1, ICW2, ICW3 and ICW4.

Bit	Description
7:0	**Interrupt Request Mask (Mask [7:0])**. When a 1 is written to any bit in this register, the corresponding IRQx line is masked. For example, if bit 4 is set to a 1, then IRQ4 is masked. Interrupt requests on IRQ4 do not set channel 4's interrupt request register (IRR) bit as long is the channel is masked. When a 0 is written to any bit in this register, the corresponding IRQx is unmasked. Note that masking IRQ2 on CNTRL-1 also masks the interrupt requests from CNTRL-2, which is physically cascaded to IRQ2.

2.5.3.7. OCW2—Operational Control Word 2 Register

I/O Address: INT CNTRL-1—020h; INT CNTRL-2—0A0h
Default Value: Bit[4:0]=undefined; Bit[7:5]=001
Attribute: Write Only

OCW2 controls both the Rotate Mode and the End of Interrupt Mode. Following a CPURST or ICW initialization, the controller enters the fully nested mode of operation. Both rotation mode and specific EOI mode are disabled following initialization.

Bit	Description
7:5	**Rotate and EOI Codes.** R, SL, EOI - These three bits control the Rotate and End of Interrupt modes and combinations of the two. A chart of these combinations is listed above under the bit definition. **Bits[7:5] Function Bits[7:5] Function** 001 Non-specific EOI Cmd 000 Rotate in Auto EOI Mode (Clear) 011 Specific EOI Cmd 111 *Rotate on Specific EOI Cmd 101 Rotate on Non-Specific EOI Cmd 110 *Set Priority Cmd 100 Rotate in Auto EOI Mode (Set) 010 No Operation * L0 - L2 Are Used
4:3	**OCW2 Select.** Must be programmed to 00 selecting OCW2.

Bit	Description
2:0	**Interrupt Level Select (L2, L1, L0).** L2, L1, and L0 determine the interrupt level acted upon when the SL bit is active (bit 6). When the SL bit is inactive, bits [2:0] do not have a defined function; programming L2, L1 and L0 to 0 is sufficient in this case. **Bit[2:0] Interrupt Level** — **Bit[2:0] Interrupt Level** 000 IRQ 0(8) — 100 IRQ 4(12) 001 IRQ 1(9) — 101 IRQ 5(13) 010 IRQ 2(10) — 110 IRQ 6(14) 011 IRQ 3(11) — 111 IRQ 7(15)

2.5.3.8. OCW3—Operational Control Word 3 Register

I/O Address: INT CNTRL-1—020h; INT CNTRL-2—0A0h
Default Value: Bit[6,0]=0; Bit[7,4:2]=Undefined; Bit[5,1]=1
Attribute: Read/Write

OCW3 serves three important functions—Enable Special Mask Mode, Poll Mode control, and IRR/ISR register read control.

Bit	Description
7	**Reserved**. Must be 0.
6	**Special Mask Mode (SMM).** If ESMM=1 and SMM=1, the interrupt controller enters Special Mask Mode. If ESMM=1 and SMM=0, the interrupt controller is in normal mask mode. When ESMM=0, SMM has no effect.
5	**Enable Special Mask Mode (ESMM).** 1=Enable SMM bit; 0=Disable SMM bit.
4:3	**OCW3 Select.** Must be programmed to 01 selecting OCW3.
2	**Poll Mode Command.** 0=Disable Poll Mode Command. When bit 2=1, the next I/O read to the interrupt controller is treated as an interrupt acknowledge cycle indicating highest priority request..
1:0	**Register Read Command.** Bits [1:0] provide control for reading the In-Service Register (ISR) and the Interrupt Request Register (IRR). When bit 1=0, bit 0 doesl not affect the register read selection. When bit 1=1, bit 0 selects the register status returned following an OCW3 read. If bit 0=0, the IRR will be read. If bit 0=1, the ISR will be read. Following ICW initialization, the default OCW3 port address read will be "read IRR". To retain the current selection (read ISR or read IRR), always write a 0 to bit 1 when programming this register. The selected register can be read repeatedly without reprogramming OCW3. To select a new status register, OCW3 must be reprogrammed prior to attempting the read. **Bit[1:0] Function** 00 No Action 01 No Action 10 Read IRQ Register 11 Read IS Register

2.5.3.9. ELCR1—Edge/Level Triggered Register

I/O Address: INT CNTRL-1—4D0h
Default Value: 00h
Attribute: Read/Write

ELCR1 register allows IRQ3 - IRQ7 to be edge or level programmable on an interrupt by interrupt basis. IRQ0, IRQ1 and IRQ2 are not programmable and are always edge sensitive.

Bit	Description
7	**IRQ7 ECL.** 0 = edge triggered mode; 1 = level sensitive mode.
6	**IRQ6 ECL.** 0 = edge triggered mode; 1 = level sensitive mode.
5	**IRQ5 ECL.** 0 = edge triggered mode; 1 = level sensitive mode.
4	**IRQ4 ECL.** 0 = edge triggered mode; 1 = level sensitive mode.
3	**IRQ3 ECL.** 0 = edge triggered mode; 1 = level sensitive mode.
2:0	**Reserved.** Must be 0.

2.5.3.10. ELCR2—Edge/Level Triggered Register

I/O Address: INT CNTRL-2—4D1h
Default Value: 00h
Attribute: Read/Write

ELCR2 register allows IRQ[15,14,12:9] to be edge or level programmable on an interrupt by interrupt basis. Note that, IRQ[13,8#] are not programmable and are always edge sensitive.

Bit	Description
7	**IRQ15 ECL.** 0 = edge triggered mode; 1 = level sensitive mode.
6	**IRQ14 ECL.** 0 = edge triggered mode; 1 = level sensitive mode.
5	**Reserved.** Must be 0.
4	**IRQ12 ECL.** 0 = edge triggered mode; 1 = level sensitive mode.
3	**IRQ11 ECL.** 0 = edge triggered mode; 1 = level sensitive mode.
2	**IRQ10 ECL.** 0 = edge triggered mode; 1 = level sensitive mode.
1	**IRQ9 ECL.** 0 = edge triggered mode; 1 = level sensitive mode.
0	**Reserved.** Must be 0.

2.5.4. X-BUS, COPROCESSOR, and RESET REGISTERS

2.5.4.1. Reset X-Bus IRQ12 And IRQ1 Register

I/O Address: 60h
Default Value: N/A
Attribute: Read only

This register clears the mouse interrupt function and the keyboard interrupt (IRQ1). Reads to this address are monitored by the PIIX. When the mouse interrupt function is enabled (X-Bus Chip Select Register), the mouse interrupt function is provided on the IRQ12/M input signal. In this mode, a mouse interrupt generates an interrupt through IRQ12 to the Host CPU. A read of 60h releases IRQ12. Reads/writes flow through to the ISA Bus.

Bit	Description
7:0	**Reset IRQ12 and IRQ1.** No specific pattern. A read of address 60h executes the command.

2.5.4.2. Coprocessor Error Register

I/O Address: F0h
Default Value: N/A
Attribute: Write only

Writing to this register causes the PIIX/PIIX3 to assert IGNNE#. The PIIX/PIIX3 also negates IRQ13 (internal to the PIIX). Note, that IGNNE# is not asserted unless FERR# is active. Reads/writes flow through to the ISA Bus.

Bit	Description
7:0	No special pattern required**.** A write to address F0h executes the command.

2.5.4.3. RC—Reset Control Register

I/O Address: CF9h
Default Value: 00h
Attribute: Read/Write

Bits 1 and 2 in this register are used by the PIIX/PIIX3 to generate a hard reset or a soft reset. Bit 2 should be cleared when writing the reset type (defined by bit 1) and then bit 2 should be set to initiate the reset. The 0 to 1 transition on bit 2 initiates the reset. For example, to initiate a soft reset via the CF9 Reset Control Register, write 00h then 04h, then read the CF9 register.

Bit	Description
7:3	**Reserved**
2	**Reset CPU (RCPU).** This bit is used to initiate (transitions from 0 to 1) a hard reset (bit 1 in this register is set to 1) or a soft reset to the CPU. During a hard reset, the PIIX/PIIX3 asserts CPURST, PCIRST#, and RSTDRV. The PIIX/PIIX3 initiates a hard reset when this register is programmed for a hard reset or PWROK is asserted. This bit cannot be read as a 1.

Bit	Description
1	**System Reset (SRST).** This bit is used in conjunction with bit 2 in this register to initiate a hard reset. When SRST =1, the PIIX/PIIX3 initiates a hard reset to the CPU when bit 2 in this register transitions from 0 to 1. When SRST=0, the PIIX/PIIX3 initiates a soft reset when bit 2 in this register transitions from 0 to 1.
0	**Reserved**

2.5.5. NMI REGISTERS

The NMI logic incorporates two different 8-bit registers. The CPU reads the NMISC Register to determine the NMI source (bits set to a 1). After the NMI interrupt routine processes the interrupt, software clears the NMI status bits by setting the corresponding enable/disable bit to a 1. The NMI Enable and Real-Time Clock Register can mask the NMI signal and disable/enable all NMI sources.

To ensure that all NMI requests are serviced, the NMI service routine software flow should be as follows:

1. NMI is detected by the processor on the rising edge of the NMI input.
2. The processor will read the status stored in port 061h to determine what sources caused the NMI. The processor may then set to 0 the register bits controlling the sources that it has determined to be active. Between the time the processor reads the NMI sources and sets them to a 0, an NMI may have been generated by another source. The level of NMI will then remain active. This new NMI source will not be recognized by the processor because there was no edge on NMI.
3. The processor must then disable all NMIs by setting bit 7 of port 070H to a 1 and then enable all NMIs by setting bit 7 of port 070H to a 0. This will cause the NMI output to transition low then high if there are any pending NMI sources. The CPU's NMI input logic will then register a new NMI.

2.5.5.1. NMISC—NMI Status And Control Register

I/O Address: 061h
Default Value: 00h
Attribute: Read/Write

This register reports the status of different system components, control the output of the speaker counter (Counter 2), and gate the counter output that drives the SPKR signal.

Bit	Description
7	**SERR# NMI Source Status—RO.** Bit 7 is set if a system board agent (PCI devices or main memory) detects a system board error and pulses the PCI SERR# line. This interrupt source is enabled by setting bit 2 to 0. To reset the interrupt, set bit 2 to 0 and then set it to 1. When writing to port 061h, bit 7 must be 0.
6	**IOCHK# NMI Source Status—RO.** Bit 6 is set if an expansion board asserts IOCHK# on the ISA Bus. This interrupt source is enabled by setting bit 3 to 0. To reset the interrupt, set bit 3 to 0 and then set it to 1. When writing to port 061h, bit 6 must be a 0.
5	**Timer Counter 2 OUT Status—RO.** The Counter 2 OUT signal state is reflected in bit 5. The value on this bit following a read is the current state of the Counter 2 OUT signal. Counter 2 must be programmed following a CPURST for this bit to have a determinate value. When writing to port 061h, bit 5 must be a 0.

Bit	Description
4	**Refresh Cycle Toggle—RO.** The Refresh Cycle Toggle signal toggles from either 0 to 1 or 1 to 0 following every refresh cycle. When writing to port 061h, bit 4 must be a 0.
3	**IOCHK# NMI Enable—R/W.** 1=Clear and disable; 0=Enable IOCHK# NMIs.
2	**PCI SERR# Enable—R/W.** 1=Clear and Disable; 0=Enable. For the PIIX3, the SERR# signal can be for a special protocol between the host-to-PCI bridge and the PIIX3 (see MSTAT Register description, 6B–6Ah, function 0).
1	**Speaker Data Enable—R/W.** 0=SPKR output is 0; 1= the SPKR output is the Counter 2 OUT signal value.
0	**Timer Counter 2 Enable—R/W.** 0=Disable; 1=Enable.

2.5.5.2. NMI Enable and Real-Time Clock Address Register

I/O Address: 070h
Default Value: Bit[6:0]=undefined; Bit 7=1
Attribute: Write Only

This port is shared with the real-time clock. Do not modify the contents of this register without considering the effects on the state of the other bits. Reads and writes to this register address flow through to the ISA Bus.

Bit	Description
7	**NMI Enable.** 1=Disable; 0=Enable.
6:0	**Real Time Clock Address.** Used by the Real Time Clock on the Base I/O component to address memory locations. Not used for NMI enabling/disabling.

2.6. System Power Management Registers

This section describes two power management registers—APMS and APMC Registers. These registers are located in normal I/O space and must be accessed (via the PCI Bus) with 8-bit accesses.

2.6.1. APMC—ADVANCED POWER MANAGEMENT CONTROL PORT

I/O Address: 0B2h
Default Value: 00h
Attribute: Read/Write

This register passes data (APM Commands) between the OS and the SMI handler. In addition, writes can generate an SMI and reads can cause STPCLK# to be asserted. The PIIX/PIIX3 operation is not effected by the data in this register.

Bit	Description
7:0	**APM Control Port (APMC).** Writes to this register store data in the APMC Register and reads return the last data written. In addition, writes generate an SMI, if bit 7 of the SMIEN Register and bit 0 of the SMICNTL Register are both is set to 1. Reads cause the STPCLK# signal to be asserted, if bit 1 of the SMICNTL Register is set to 1. Reads do not generate an SMI.

2.6.2. APMS—ADVANCED POWER MANAGEMENT STATUS PORT

I/O Address: 0B3h
Default Value: 00h
Attribute: Read/Write

This register passes status information between the OS and the SMI handler. The PIIX/PIIX3 operation is not effected by the data in this register.

Bit	Description
7:0	**APM Status Port (APMS).** Writes store data in this register and reads return the last data written.

2.7. PCI BUS Master IDE Registers

The PCI Bus master IDE function uses 16 bytes of I/O space, allocated via the BMIBA register (A PCI Base Address register). All bus master IDE I/O space registers can be accessed as byte, word, or Dword quantities. The description of the 16 bytes of I/O registers follows:

2.7.1. BMICOM—BUS MASTER IDE COMMAND REGISTER

Address Offset: Primary Channel—Base + 00h; Secondary Channel—Base + 08h
Default Value: 00h
Attribute: Read / Write

This register enables/disables bus master capability for the IDE function and provides direction control for the IDE DMA transfers. This register also provides bits that software uses to indicate DMA capability of the IDE device.

Bit	Description
7:4	**Reserved**
3	**Bus Master Read/Write Control (RWCON).** 0=Reads; 1=Writes. This bit must NOT be changed when the bus master function is active.
2:1	**Reserved.**
0	**Start/Stop Bus Master (SSBM).** 1=Start; 0=Stop. When this bit is set to 1, bus master operation starts. The controller transfers data between the IDE device and memory only when this bit is set. Master operation can be stopped by writing a 0 to this bit. This results in all state information being lost (i.e., master mode operation cannot be stopped and then resumed). If this bit is set to 0 while bus master operation is still active (i.e., Bit 0=1 in the Bus Master IDE Status Register for that IDE channel) and the drive has not yet finished its data transfer (bit 2=0 in the channel's Bus Master IDE Status Register), the bus master command is aborted and data transferred from the drive may be discarded before being written to system memory. This bit is intended to be set to 0 after the data transfer is completed, as indicated by either bit 0 or bit 2 being set in the IDE Channel's Bus Master IDE Status Register.

2.7.2. BMISTA—BUS MASTER IDE STATUS REGISTER

Address Offset: Primary Channel—Base + 02h; Secondary Channel—Base + 0Ah
Default Value: 00h
Attribute: Read/Write Clear

This register provides status information about the IDE device and state of the IDE DMA transfer. Table 8 describes IDE Interrupt Status and Bus Master IDE Active bit states after a DMA transfer has been started.

Bit	Description
7	**Reserved.** This bit is hardwired to 0.
6	**Drive 1 DMA Capable (DMA1CAP)—R/W.** 1=Drive 1 is capable of DMA transfers. This bit is a software controlled status bit that indicates IDE DMA device capability and does not affect hardware operation.
5	**Drive 0 DMA Capable (DMA0CAP)—R/W.** 1=Drive 0 is capable of DMA transfers. This bit is a software controlled status bit that indicates IDE DMA device capability and does not affect hardware operation.
4:3	**Reserved.**
2	**IDE Interrupt Status—R/WC.** This bit, when set to a 1, indicates when an IDE device has asserted its interrupt line. When bit 2=1, 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. Software sets this bit to a 0 by writing a 1 to it. IRQ14 is used for the primary channel and MIRQ0 is used for the secondary channel. IRQ15 is used for the Secondary Channel when IRQ0 bit, bit5 in the MIRQ0 register, is set. If the interrupt status bit is set to a 0 by writing a 1 to this bit while the interrupt line (IRQ14 or MIRQ0) is still at the active level, this bit remains 0 until another assertion edge is detected on the interrupt line.
1	**IDE DMA Error—R/WC.** This bit is set to 1 when the PIIX/PIIX3 encounters a target abort or master abort while transferring data on the PCI Bus. Software sets this bit to a 0 by writing a 1 to it.
0	**Bus Master IDE Active (BMIDEA)—RO.** The PIIX/PIIX3 sets this bit to 1 when bit 0 in the BMICOM Register is set to 1. The PIIX/PIIX3 sets this bit to 0 when the last transfer for a region is performed (where EOT for that region is set in the region descriptor). The PIIX/PIIX3 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. When this bit is read as a zero, all data transferred from the drive during the previous bus master command is visible in system memory, unless the bus master command was aborted.

Table 8. Interrupt/Activity Status Combinations

Bit 2	Bit 0	Description:
0	1	**DMA transfer is in progress.** No interrupt has been generated by the IDE device.
1	0	**The IDE device generated an interrupt and the Physical Region Descriptors exhausted.** This is normal completion where the size of the physical memory regions is equal to the IDE device transfer size.
1	1	**The IDE device generated an interrupt.** The controller has not reached the end of the physical memory regions. This is a valid completion case when the size of the physical memory regions is larger than the IDE device transfer size.

Bit 2	Bit 0	Description:
0	0	**Error condition.** If the IDE DMA Error bit is 1, there is a problem transferring data to/from memory. Specifics of the error have to be determined using bus-specific information. If the Error bit is 0, the PRD's specified a smaller size than the IDE transfer size.

2.7.3. BMIDTP—BUS MASTER IDE DESCRIPTOR TABLE POINTER REGISTER

Address Offset: Primary Channel—Base + 04h; Secondary Channel—Base + 0Ch
Default Value: 00000000h
Attribute: Read/Write

This register provides the base memory address of the descriptor table. The Descriptor Table must be Dword aligned and not cross a 4-Kbyte boundary in memory.

Bit	Description
31:2	**Descriptor Table Base Address.** Bits [31:2] correspond to A[31:2].
1:0	**Reserved.**

2.8. USB I/O Registers

This section describes the block of USB registers that are located in normal I/O space. The "base" portion of the I/O address is selected via a PCI Configuration register.

2.8.1. USBCMD—USB Command Register

I/O Address: Base+ (00–01h)
Default Value: 0000h
Attribute: Read/Write

The Command Register indicates the command to be executed by the serial bus host/hub controller. Writing to the register causes a command to be executed.

Bit	Description
15:8	**Reserved.**
7	**Max Packet (MAXP).** 1=64 bytes. 0=32 bytes. This bit selects the maximum packet size that can be used for full speed bandwidth reclamation at the end of a frame. This value is used by the Host Controller to determine whether it should initiate another transaction based on the time remaining in the SOF counter. Use of reclamation packets larger than the programmed size will cause a Babble error if executed during the critical window at frame end. The Babble error results in the offending endpoint being stalled. Software is responsible for ensuring that any packet that could be executed under bandwidth reclamation be within this size limit.
6	**Configure Flag (CF).** Software sets this bit as the last action in its process of configuring the Host Controller. This bit has no effect on the hardware. It is provided only as a semaphore service for software.

Bit	Description
5	**Software Debug (SWDBG).** 1=Debug mode. 0=Normal Mode. In SW Debug mode, the Host Controller clears the Run/Stop bit after the completion of each USB transaction. The next transaction is executed when software sets the Run/Stop bit back to 1. The SWDBG bit must only be manipulated when the controller is in the stopped state. This can be determined by checking the HCHalted bit in the USBSTS register.
4	**Force Global Resume (FGR).** 1=Host Controller sends the Global Resume signal on the USB. Software sets this bit to 0 after 20 ms has elapsed to stop sending the Global Resume signal. At that time all USB devices should be ready for bus activity. The Host Controller sets this bit to 1 when a resume event (connect, disconnect, or K-state) is detected while in global suspend mode. Software resets this bit to 0 to end Global Resume signaling. The 1 to 0 transition causes the port to send a low speed EOP signal. This bit remains a 1 until the EOP has completed.
3	**Enter Global Suspend Mode (EGSM).** 1=Host Controller enters the Global Suspend mode. No USB transactions occur during this time. The Host Controller is able to receive resume signals from USB and interrupt the system. Software resets this bit to 0 to come out of Global Suspend mode. Software writes this bit to 0 at the same time that Force Global Resume (bit 4) is written to 0 or after writing bit 4 to 0. Software must also ensure that the Run/Stop bit (bit 0) is cleared prior to setting this bit.
2	**Global Reset (GRESET).** When this bit is set, the Host Controller sends the global reset signal on the serial bus and then resets all its logic, including the internal hub registers. The hub registers are reset to their power on state. This bit is reset by the software after a minimum of 10 ms has elapsed as specified in chapter 7 of the USB Specification. Note: Chip Hardware Reset has the same effect as Global Reset (bit 2), except that the Host Controller does not send the Global Reset on USB.
1	**Host Controller Reset (HCRESET).** When this bit is set to a 1, the Host Controller module resets its internal timers, counters, state machines, etc. to their initial value. Any transaction currently in progress on USB is immediately terminated. This bit is reset by the Host Controller when the reset process is complete. The HCRESET acts on Hub registers are slightly different for Chip Hardware Reset and Global USB Reset. The HCReset affects bits [8,3:0] of the Port Status and Control Register (PORTSC) of each port. HCRESET resets the state machines of the Host Controller including the Connect/Disconnect state machine (one for each port). When the Connect/Disconnect state machine is reset, the output that signals connect/disconnect are negated to 0, effectively signaling a disconnect, even if a device is attached to the port. This virtual disconnect causes the port to be disabled. This disconnect and disabling of the port causes bit 1 (connect status change) and bit 3 (port enable/disable change) of the PORTSC to get set. The disconnect also causes bit 8 of PORTSC to reset. About 64 bit times after HCRESET goes to 0, the connect and low-speed detect will take place and bits 0 and 8 of the PORTSC will change accordingly.
0	**Run/Stop (RS).** 1=Run. 0=Stop. When set to a 1, the Host Controller proceeds with execution of the schedule. The Host Controller continues execution as long as this bit is set. When this bit is set to 0, the Host Controller completes the current transaction on the USB and then halts. The HC Halted bit in the status register indicates when the Host Controller has finished the transaction and has entered the stopped state. The Host Controller clears this bit when the following fatal errors occur: consistency check failure, PCI Bus errors.

Table 9. Run/Stop, Debug Bit Interaction

SWDBG (Bit 5)	Run/Stop (Bit 0)	Operation
0	0	If executing a command, the Host Controller completes the command and then stops. The 1.0 ms frame counter is reset and command list execution resumes from start of frame using the frame list pointer selected by the current value in the FRNUM register. (While Run/Stop=0, the FRNUM register can be reprogrammed).
0	1	Execution of the command list resumes from Start Of Frame using the frame list pointer selected by the current value in the FRNUM register. The Host Controller remains running until the Run/Stop bit is cleared (by Software or Hardware).
1	0	If executing a command, the Host Controller completes the command and then stops and the 1.0 ms frame counter is frozen at its current value. All status are preserved. The Host Controller begins execution of the command list from where it left off when the Run/Stop bit is set.
1	1	Execution of the command list resumes from where the previous execution stopped. The Run/Stop bit is set to 0 by the Host Controller when a TD is being fetched. This causes the Host Controller to stop again after the execution of the TD (single step). When the Host Controller has completed execution, the HC Halted bit in the Status Register is set.

2.8.2. USBSTS—USB Status Register

I/O Address: Base + (02–03h)
Default Value: 0000h
Attribute: Read/Write Clear
size: 16 bits

This register indicates pending interrupts and various states of the Host Controller. The status resulting from a transaction on the serial bus is not indicated in this register. Software sets a bit to 0 in this register by writing a 1 to it.

Bit	Description
15:6	**Reserved**.
5	**HCHalted.** The Host Controller sets this bit to 1 after it has stopped executing as a result of the Run/Stop bit being set to 0, either by software or by the Host Controller hardware (an internal error).
4	**Host Controller Process Error.** The Host Controller sets this bit to 1 when it detects a fatal error and indicates that the Host Controller suffered a consistency check failure while processing a Transfer Descriptor. An example of a consistency check failure would be finding an illegal PID field while processing the packet header portion of the Transfer Descriptor (TD) data structure. When this error occurs, the Host Controller clears the Run/Stop bit in the Command register to prevent further schedule execution. A hardware reset is generated to the system.

Bit	Description
3	**PCI Bus Error.** The Host Controller sets this bit to 1 when a serious error occurs during a PCI access involving the Host Controller module. PCI conditions that set this bit to 1 include PCI Master Abort and PCI Target Abort. When this error occurs, the Host Controller clears the Run/Stop bit in the Command register to prevent further execution of the scheduled TDs. A hardware reset is generated to the system.
2	**Resume Received.** The Host Controller sets this bit to 1 when it receives a “RESUME” signal from a USB device. This is only valid if the Host Controller has been in a suspended state (bit 3 of Command register = 1).
1	**USB Error Interrupt.** The Host Controller sets this bit to 1 when completion of a USB transaction results in an error condition (e.g., error counter underflow). If the Transfer Descriptor (TD) data structure on which the error interrupt occurred also had its IOC bit set, both this bit and Bit 0 are set.
0	**USB Interrupt (USBINT).** The Host Controller sets this bit to 1 when the cause of an interrupt is a completion of a USB transaction whose Transfer Descriptor had its IOC bit set. The Host Controller also sets this bit to 1 when a short packet is detected (actual length field in TD is less than maximum length field in TD), and short packets are enabled in that TD.

2.8.3. USBINTR—USB Interrupt Enable Register

I/O Address: Base + (04–05h)
Default Value: 0000h
Attribute: Read/Write
size: 16 bits

This register enables/disables reporting of the corresponding interrupt to the software. When a bit is set and the corresponding interrupt is active, an interrupt is generated to the host. Fatal errors (Host Controller Processor Error- bit 4, USBSTS Register) cannot be disabled. Interrupt sources that are disabled in this register still appear in the Status Register to allow the software to poll for events.

Bit	Description
15:4	**Reserved**.
3	**Short Packet Interrupt Enable.** 1=Enabled. 0=Disabled.
2	**Interrupt On Complete (IOC) Enable.** 1= Enabled. 0=Disabled.
1	**Resume Enable.** 1= Enabled. 0=Disabled.
0	**Timeout/CRC Enable.** 1= Enabled. 0=Disabled.

2.8.4. FRNUM—Frame Number Register

I/O Address: Base + (06–07h)
Default Value: 0000h
Attribute: Read/Write (Writes must be Word Writes)
Size: 16 bits

Bits [10:0] of this register contain the current frame number and this number is included in the frame SOF packet. This register reflects the count value of the internal frame number counter. Bits [9:0] are used to

select a particular entry in the Frame List during schedule execution. (Frame List is one of the data structures located in system memory that is processed by the Host Controller.)

This register must be written as a word. Byte writes are not supported. This register cannot be written unless the Host Controller is in the STOPPED state as indicated by the HCHalted bit (USBSTS register). A write to this register while the Run/Stop bit (USBCMD register) is set is ignored.

Bit	Description
15:11	**Reserved.**
10:0	**Frame List Current Index/Frame Number.** Bits [10:0] provide the frame number in the SOF Frame. The value in this register increments at the end of each time frame (approximately every 1 ms). In addition, bits [9:0] are used for the Frame List current index and correspond to memory address signals [11:2].

2.8.5. FLBASEADD—Frame List Base Address Register

I/O Address: Base + (08–0Bh)
Default Value: Undefined
Attribute: Read/Write
Size: 32 bits

This 32-bit register contains the beginning address of the Frame List in the system memory. Software loads this register prior to starting the schedule execution by the Host Controller. When written, only the upper 20 bits are used. The lower 12 bits are written as zero (4 Kbyte alignment). The contents of this register are combined with the frame number counter to enable the Host Controller to step through the Frame List in sequence. The two lsb are always 00. This requires DWord alignment for all list entries. This configuration supports 1K (1024) Frame List entries.

Note that Frame List is one of the data structures located in system memory that is processed by the Host Controller during schedule execution.

Bit	Description
31:12	**Base Address.** These bits correspond to memory address signals [31:12], respectively.
11:0	**Reserved.** Must be written as 0s.

2.8.6. Start Of Frame (SOF) Modify Register

I/O Address: Base + (0Ch)
Default Value: 40
Attribute: Read/Write
Size: 8 bits

This one byte register is used to modify the value used in the generation of SOF timing on the USB. Only the 7 least significant bits are used. When a new value is written into the these 7 bits, the SOF timing of the next frame will be adjusted. This feature can be used to adjust out any offset from the frequency synthesizer that generates the clock that drives the SOF counter. This register can also be used to maintain real time synchronization with the rest of the system so that all devices have the same sense of real time. Using this register, the frame length can be adjusted across the full range required by the USB specification.

It's initial programmed value is system dependent based on the accuracy of hardware USB clock and is initialized by system BIOS. It may be reprogrammed by USB system software at any time. Its new value will

take effect from the beginning of the next frame. This register is reset upon a Host Controller Reset or Global Reset. Software must maintain a copy of its value for reprogramming if necessary.

Bit	Description
7	**Reserved.**
6:0	**SOF Timing Value.** Guidelines for the modification of frame time are contained in Chapter 7 of the USB Specification. The SOF counter is initialized to a count of 11936. The SOF cycle time (number of SOF counter clock periods to generate a SOF) is equal to 11936 + value in this field. The default value is decimal 64 which gives a SOF count value of 12000. For a 12 MHz SOF counter clock input, this produces a 1 ms Frame period. The following table indicates what SOF Timing Value to program into this field for a certain frame.

Frame Length (# 12Mhz Clocks) (decimal)	SOF Reg. Value (decimal)
11936	0
11937	1
.	.
.	.
11999	63
12000	64
12001	65
.	.
.	.
12062	126
12063	127

2.8.7. PORTSC—Port Status and Control Register

I/O Address:	Base + (10–11h)—Port 1 Base + (12–13h) —Port 2
Default:	0000h
Access:	Read/Write (WORD writeable only)
Size:	16 bits

After a Power-up and after a RESET, the initial conditions of a port are: No device connected, Port disabled, and the bus line status is 00 (single-ended zero). Note: If a device is attached, the port state will transition to the attached state and system software will process this as with any status change notification. It may take up to 64 USB bit times for the port transition to occur. Note that, if the Host Controller is in Global Suspend mode, then, if any of bits [6,3,1] gets set, the Host Controller will signal a resume. Refer to Chapter 11 of the USB Specification for details on hub operation.

Bit	Description
15:13	**Reserved.** Must written as 0s when writing this register.
12	**Suspend—R/W.** 1=Port in suspend state. 0=Port not in supsend state. This bit should not be written to a 1 if global suspend is active (bit 3=1 in the USBCMD register). Bit 2 and bit 12 of this register define the hub states as follows: **Bits [12,2]** **Hub Port State** x0 Disable 01 Enable 11 Suspend When in suspend state, downstream progagation of data is blocked on this port, except for single-ended 0 resets (global reset and port reset). The blocking occurs at the end of the current transaction if a transaction was in progress when this bit was written to 1. In the suspend state the port will respond to a resume. Note that the bit status does not change until the port is suspended and that there may be a delay in suspending a port if there is a transaction currently in progress on the USB.
11:10	**Reserved.**
9	**Port Reset—R/W**. 1=Port is in Reset. 0=Port is not in Reset. When in the Reset State, the port is disabled and sends the USB Reset signaling. Note that host software must guarantee that the RESET signaling is active for the proper amount of time as specified in the USB Specification.
8	**Low Speed Device Attached—RO.** 1=Low speed device is attached to this port. 0=Full speed device. Writes have no effect.
7	**Reserved—RO.** Always read as 1.
6	**Resume Detect—RO.** 1= Resume detected/driven on port. 0=No resume (K-state) detected/driven on port. Software sets this bit to a 1 to drive resume signalling. The Host Controller sets this bit to a 1 if a J-to-K transition is detected while the port is in the Suspend state. Note that when this bit is 1, a K-state is driven on the port as long as this bit remains 1 and the port is still in suspend state. Writing a 0 (from 1) causes the port to send a low speed EOP. This bit remains a 1 until the EOP has completed.
5:4	**Line Status—RO.** These bits reflect the D+ (bit 4) and D- (bit 5) signals lines' logical levels. These bits are used for fault detect and recovery as well as for USB diagnostics. This field is updated at EOF2 time (See chapter 11 of the USB Specification).
3	**Port Enable/Disable Change—R/WC.** 1=Port enabled/disabled status has changed. 0=No change. For the root hub, this bit gets set only when a port is disabled due to disconnect on the that port or due to the appropriate conditions existing at the EOF2 point (See chapter 11 of the USB Specification). Software clears this bit by writing a 1 to it.
2	**Port Enabled/Disabled—R/W.** 1=Enable. 0=Disable. Ports can be enabled by host software only. Ports can be disabled by either a fault condition (disconnect event or other fault condition) or by host software. Note that the bit status does not change until the port state actually changes and that there may be a delay in disabling a port if there is a transaction currently in progress on the USB.

Bit	Description
1	**Connect Status Change—R/WC.** 1=Change in Current Connect Status. 0=No change. Indicates a change has occurred in the port's Current Connect Status (see bit 0). The hub device sets this bit for any changes to the port device connect status, even if system software has not cleared a connect status change. If, for example, the insertion status changes twice before system software has cleared the changed condition, hub hardware will be "setting" an already-set bit (i.e., the bit will remain set). However, the hub transfers the change bit only once when the Host Controller requests a data transfer to the Status Change endpoint. System software is responsible for determining state change history in such a case. Software sets this bit to 0 by writing a 1 to it.
0	**Current Connect Status—RO.** 1=Device is present on port. 0=No device is present. This value reflects the current state of the port, and may not correspond directly to the event that caused the Insertion Status Change bit (Bit 1) to be set.

Table 10 and Table 11 show Host Controller behavior when a port receives resume, connect, or disconnect signaling and the Host Controller is in global suspend state or not in the global suspend state. A full explanation of hub behavior is given in Chapter 11 of the USB specification. Typically, the PORTSC register associated with the port receiving the signaling reflects the status change appropriate for the type of signaling received. Resume signaling (K-State) is recognized in the PORTSC register only if the port is in selective suspend (PORTSC[bits 2,12]=1,1). Resume is recognized (USBSTS[bit 2]=1), if resume is received on a suspended or enabled port when the Host Controller is in the global suspend state (USBCMD[bit 3]=1).

The host may also initiate a resume on a suspended port or when the Host Controller has been suspended by writing the appropriate resume-detect/force-resume bit to a 1. A global resume is started by writing USBCMD[bit 4] to a 1. A K-State signal is sent on all enabled ports. Any port that needs to send the resume signal and is not enabled must be enabled before the resume is forced. A resume can be forced on a selectively suspended port by writing the corresponding PORTSC[bit 6] to a 1.

Resume signaling is ended by clearing the appropriate suspend and resume bits. This is true for either selective or global resumes, or resumes initiated by signaling at the port or by the Host Controller. For proper single-ended zero termination of the resume signaling, the suspend and resume bits must be simultaneously written to 0 (same write cycle) or the suspend must be written to a 0 after the resume bit is reset. Resume is ended on a suspended Host Controller by writing USBCMD[bits 3,4] to 0. Resume is ended on a suspended port by writing PORTSC[bits 6,12] to 0. If signaling on a suspended port in a globally suspended Host Controller is the source of the resume, the Host Controller suspend and resume bits should be cleared before the port bits are cleared.

Table 10. Behavior During Resume When Host Not In Global Suspend State

Port Status and Signaling Type	Signaled Port Response	Adjacent Port Response		
		Enabled Port	Disabled Port	Suspended Port
Port disabled, resume K-State received	No Effect	No Effect	No Effect	No Effect
Port suspended, Resume K-State received	Resume reflected downstream on signaled port. Resume Detect status bit in PORTSC register is set.	No Effect	No Effect	No Effect
Port enabled, disabled or suspended and disconnect received	PORTSC Connect and Enable status bits are cleared, and Connect Change and Enable/Disable Change status bits are set.	No Effect	No Effect	No Effect

Table 10. Behavior During Resume When Host Not In Global Suspend State

Port Status and Signaling Type	Signaled Port Response	Adjacent Port Response		
		Enabled Port	Disabled Port	Suspended Port
Port disabled and connect received	PORTSC Connect Status and Connect Status Change bits are set.	No Effect	No Effect	No Effect

Table 11. Behavior During Resume when Host is in Global Suspend State

Port Status and Signaling Type	Signaled Port Response	Adjacent Port Response		
		Enabled Port	Disabled Port	Suspended Port
Port enabled, Resume K-State received	Resume reflected downstream on signaled port. Resume Detect Status bit in USBSTS Reg is set.	Signal resume downstream	No Effect	No Effect
Port disabled, resume K-State received	No Effect	No Effect	No Effect	No Effect
Port suspended, Resume K-State received	Resume signal reflected downstream on signaled port. Resume Detect status bit in PORTSC and USBSTS Regs are set.	Signal resume downstream	No Effect	No Effect
Port enabled, disabled or suspended and disconnect received	Resume Detect status bit in USBSTS Reg is set. PORTSC Connect and Enable status bits are cleared and Connect Change and Enable/Disable Change bits are set.	Signal resume downstream	No Effect	No Effect
Port disabled and connect received	Resume Detect status bit in USBSTS Reg is set. PORTSC Connect status bit and Connect Change status bit are set.	Signal resume downstream	No Effect	No Effect

3.0. FUNCTIONAL DESCRIPTION

This section describes each of the major functions on the PIIX/PIIX3 including the memory and I/O address map, DMA controller, interrupt controller, timer/counter, and power management. The PCI, ISA, X-Bus, and IDE interfaces.

3.1. Memory and I/O Address Map

The PIIX/PIIX3 interfaces to two system buses—PCI and ISA Buses. The PIIX/PIIX3 provides positive decode for certain I/O and memory space accesses on these buses as described in this section.

ISA masters and DMA devices have access to PCI memory and some of the internal PIIX/PIIX3 registers as described in the Register Description section. ISA masters and DMA devices do not have access to host or PCI I/O space.

3.1.1. I/O Accesses

The PIIX/PIIX3 positively decodes accesses to the PCI configuration registers (PCI only), power management registers (PCI only), and bus master IDE interface registers (PCI only). The PIIX/PIIX3 subtractively decodes memory accesses to the APIC Registers (PCI only), The PIIX/PIIX3 also positively decodes the ISA-Compatible registers (PCI and ISA), except for the DMA register I/O space which is subtractively decoded. For details concerning accessing these registers, see Register Description section.

The PIIX/PIIX3 also provides positive decode for BIOS, X-Bus, and system event decode for SMM support. In addition, the PIIX/PIIX3 positively decodes PCI Bus accesses to registers located on the IDE device, when enabled. For IDE port accesses, see PCI Local Bus IDE section.

3.1.2. Memory Address Map

For PCI accesses to ISA memory, accesses below 16 Mbytes (including BIOS space) that are not claimed by a PCI device (subtractive decode) are forwarded to ISA. For write accesses that are not claimed by an ISA slave, the cycle completes normally (i.e., 8-bit, 6 SYSCLK cycle). For read accesses that are not claimed by an ISA slave, the PIIX/PIIX3 returns data corresponding to the state of the ISA Bus and completes the cycle normally (i.e., 8-bit, 6 SYSCLK cycle).

For ISA/DMA accesses to main memory, all accesses to memory locations 0–512 Kbytes (512–640 Kbytes, if enabled), or accesses above 1 Mbyte and below the top of memory are forwarded to the PCI Bus (Table 12). The Top of Memory is equal to the value programmed in the Top of Memory Register (bits [7:3]). The PIIX3 can also enable ISA/DMA accesses to the VGA frame buffer space from 640-768 Kbytes (A0000-BFFFFh). All remaining ISA originated accesses are confined to the ISA Bus.

Table 12. DMA and ISA Master Accesses to Main Memory

Memory Space	Response
Top of main memory to 128 Mbytes	Confine to ISA
1 Mbyte to top of main memory	Forward to main memory[1]
1 Mbyte minus 128 Kbytes to 1 Mbyte minus 64 Kbytes	Confine to ISA[2]
640 Kbytes to 1 Mbyte minus 128 Kbytes	Confine to ISA
512–640 Kbytes	Confine to ISA[3]
0–512 Kbytes	Forward to PCI

NOTES:
1. Except accesses to programmed memory hole.
2. Forward to main memory if bit 6=0 in the XBCS Register and bit 3=1 in the TOM Register.
3. Forward to main memory if bit 1=0 in the TOM Register.

3.1.3. BIOS MEMORY

The PIIX/PIIX3 supports 512 Kbytes of BIOS space. This includes the normal 128-Kbyte space plus an additional 384-Kbyte BIOS space (known as the extended BIOS area). The XBCS Register provides BIOS space access control. Access to the lower 64-Kbyte block of the 128-Kbyte space and the extended BIOS space can be individually enabled/disabled. In addition, write protection can be programmed for the entire BIOS space.

PCI Access to BIOS Memory

The 128-Kbyte BIOS memory space is located at 000E0000–000FFFFFh (top of 1 Mbyte) and is aliased at FFFE0000h (top of 4 Gbytes). This 128-Kbyte byte block is split into two 64-Kbyte blocks. Accesses to the top 64 Kbytes (000F0000–000FFFFFh) are forwarded to the ISA Bus and BIOSCS# is always generated. Accesses to the bottom 64 Kbytes (000E0000–000EFFFFh) are forwarded to the ISA Bus and BIOSCS# is only generated when this BIOS region is enabled. 1.If this BIOS region is enabled (bit 6=1 in the XBCS Register), accesses to the aliased region at the top of 4 Gbytes (FFFE0000h - FFFEFFFFh) are forwarded to ISA and BIOSCS# generated. If disabaled, these accesses are not forwarded to ISA and BIOSCS# is not generated.

The additional 384-Kbyte region resides at FFF80000–FFFDFFFFh. If this BIOS region is enabled (bit 7=1 in the XBCS Register), these accesses (FFF80000h–FFFDFFFFh) are forwarded to ISA and BIOSCS# generated. If disabled, these accesses are not forwarded to ISA and BIOSCS# not generated.

ISA Access to BIOS Memory

The PIIX/PIIX3 confines all ISA-initiated BIOS accesses to the top 64 Kbytes of the 128-Kbyte region (F0000–FFFFFh) to the ISA Bus, even if BIOS is shadowed in main memory. Accesses to the bottom 64 Kbytes of the 128-Kbyte BIOS region (E0000–EFFFFh) are confined to the ISA Bus, when this region is enabled. When the BIOS region is disabled, accesses are forwarded to main memory.

Accesses to the top 64-Kbyte BIOS region always generates BIOSCS#. Accesses to the bottom 64-Kbyte BIOS region generate BIOSCS#, when this region is enabled.

3.2. PCI Interface

The PIIX/PIIX3 incorporates a fully PCI Bus compatible master and slave interface. As a PCI master, the PIIX/PIIX3 runs cycles on behalf of DMA, ISA masters, or a bus master IDE. As a PCI slave, the PIIX/PIIX3 accepts cycles initiated by PCI masters targeted for the PIIX's internal register set or the ISA bus. The PIIX/PIIX3 directly supports the PCI interface running at either 25 MHz, 30 MHz, or 33 MHz.

3.2.1. TRANSACTION TERMINATION

The PIIX/PIIX3 supports the standard PCI cycle terminations as described in the PCI Local Bus specification.

PIIX/PIIX3 As Master—Master-Initiated Termination: The PIIX/PIIX3 supports three forms of master-initiated termination: 1.) Normal termination of a completed transaction, 2.) Normal termination of an incomplete transaction due to timeout (applies to line buffer operations-IDE Bus Master, 3.) Abnormal termination due to the slave not responding to the transaction (Abort).

PIIX/PIIX3 As a Master—Response to Target-Initiated Termination: As a master, the PIIX/PIIX3 responds in one of three ways to a target-termination— Target-Abort, Retry, or Disconnect.

PIIX/PIIX3 As a Target—Target-Initiated Termination: The PIIX/PIIX3 supports three forms of Target-initiated Termination — Disconnect, Retry, Target Abort.

3.2.2. PARITY SUPPORT

As a master, the PIIX/PIIX3 generates address parity for read/write cycles and data parity when the PIIX/PIIX3 is providing the data. As a slave, the PIIX/PIIX3 generates data parity for read cycles. The PIIX/PIIX3 does not check parity and does not generate SERR#. However, the PIIX/PIIX3 does generate an NMI when another PCI device asserts SERR# (if enabled).

PAR is the calculated parity signal. PAR is even parity and is calculated on 36 bits—AD[31:0] signals plus C/BE[3:0]#.PAR is always calculated on 36 bits, regardless of the valid byte enables. PAR is only guaranteed to be valid one PCI clock after the corresponding address or data phase.

3.2.3. PCI ARBITRATION

The PIIX/PIIX3 requests the use of the PCI Bus on behalf of ISA devices (bus masters and DMA) and IDE DMA slave devices using the PHOLD# and PHLDA# signals. These signals connect to the Host-to-PCI Bridge where the PCI arbiter is located.

ISA devices (Bus Master or DMA) assert DREQ to gain access to the ISA Bus. In response, The PIIX/PIIX3 asserts PHOLD#. The PIIX/PIIX3 keeps DACK negated until the PIIX/PIIX3 has ownership of the PCI Bus and Memory. The PCI arbiter asserts PHLDA# to the PIIX/PIIX3 when the above conditions are met. The PIIX/PIIX3 gives ownership of the ISA Bus (PCI and Memory) to the ISA device after sampling PHLDA# asserted.

Arbitration

The PIIX/PIIX3 requests the use of PCI bus on behalf of ISA devices, IDE Masters and USBHC channels. The PIIX/PIIX3 arbitrates for the PCI bus through the PHOLD# and PHLDA# signals. The ISA DMA/Master channels, the IDE bus master channels and USBHC channels are arbitrated fairly as a group (fairness between three groups).

Multiple Transactions on PCI Bus (PIIX3 Only)

The USB module utilizes the arbitration advantage available through the PHOLD#/PHLDA# to do multiple transactions on the PCI bus once it has the ownership of the bus and the MLT count has not expired. The USBHC relinquishes the bus ownership as soon the transactions are completed or the MLT counter has expired, whichever happens first. Refer to the DLC register description for information on delayed completion and passive release.

3.3. ISA Interface

The PIIX/PIIX3 incorporates a fully ISA Bus compatible master and slave interface. The PIIX/PIIX3 directly drives five ISA slots without external data buffers. External transceivers are used on the SA[19:8] and SBHE# signals to permit these signals to be used with the IDE interface (Figure 1). The ISA interface also provides byte swap logic, I/O recovery support, wait state generation, and SYSCLK generation.

The ISA interface supports the following types of cycles:

- PCI master-initiated I/O and memory cycles to the ISA Bus
- DMA compatible cycles between main memory and ISA I/O and between ISA I/O and ISA memory
- Enhanced DMA cycles between PCI memory and ISA I/O (for motherboard devices only)
- ISA refresh cycles initiated by either the PIIX/PIIX3 or an external ISA master
- ISA master-initiated memory cycles to PCI and ISA master-initiated I/O cycles to the internal PIIX/PIIX3 registers, as shown in ISA-Compatible Register table in the Register Description section.

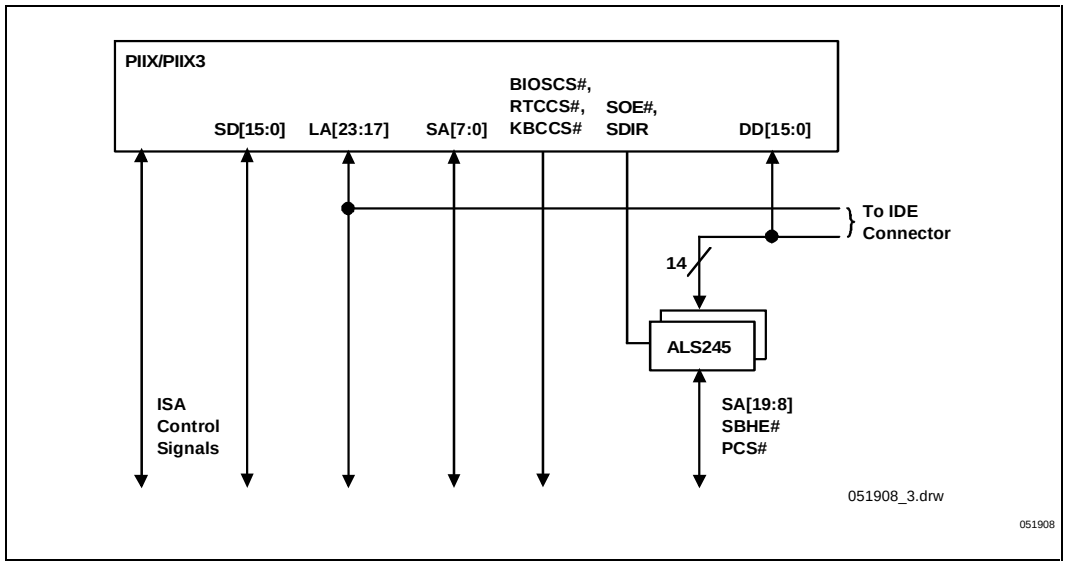


Figure 1. ISA Interface

3.4. DMA Controller

The DMA circuitry incorporates the functionality of two 82C37 DMA controllers with seven independently programmable channels (Channels 0-3 and Channels 5-7). DMA Channel 4 is used to cascade the two controllers and defaults to cascade mode in the DMA Channel Mode (DCM) Register. In addition to accepting requests from DMA slaves, the DMA controller also responds to requests that are initiated by software. Software may initiate a DMA service request by setting any bit in the DMA Channel Request Register to a 1. The DMA controller for Channels 0-3 is referred to as "DMA-1" and the controller for Channels 4-7 is referred to as "DMA-2".

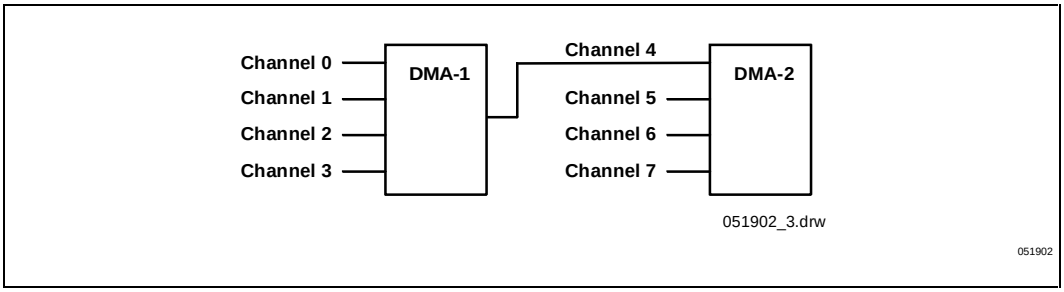


Figure 2. Internal DMA Controller

Each DMA channel is hardwired to the compatible settings for DMA device size; channels [3:0] are hardwired to 8-bit count-by-bytes transfers and channels [7:5] are hardwired to 16-bit count-by-words (address shifted) transfers. The PIIX/PIIX3 provides the timing control and data size translation necessary for the DMA transfer between the memory (ISA or main memory) and the ISA Bus device. ISA Compatible and F type DMA timing are supported. Type F DMA is selected via the MBDMA[1:0] Registers and permits up to two channels to be programmed for type F transfers at the same time.

The PIIX/PIIX3 provides 24-bit addressing in compliance with the ISA-Compatible specification. Each channel includes a 16-bit ISA-Compatible Current Register that contains the 16 least-significant bits of the 24-bit address, an ISA Compatible Page Register that contains the eight next most significant bits of address. The DMA controller also features refresh address generation, and auto-initialization following a DMA termination.

The DMA controller is either in master or slave mode. In master mode, the DMA controller is either servicing a DMA slave's request for DMA cycles or allowing a 16-bit ISA master to use the bus (via a cascaded DREQ signal). In slave mode, the PIIX/PIIX3 monitors both the ISA Bus and PCI, decoding and responding to I/O read and write commands that address its registers.

Note that a DMA device (I/O device) is always on the ISA Bus, but the memory referenced is located on either an ISA Bus device or on PCI. When the PIIX/PIIX3 is running a compatible DMA cycle, it drives the MEMR# or MEMW# strobes if the address is less than 16 Mbytes (000000–FFFFFFh). These memory strobes are generated regardless of whether the cycle is decoded for PCI or ISA memory. The SMEMR# and SMEMW# are generated if the address is less than 1 Mbytes (0000000–00FFFFFh). If the address is greater than 16 Mbytes (1000000–7FFFFFFh), the MEMR# or MEMW# strobe is not generated in order to avoid aliasing problems.

The PIIX/PIIX3 drives the AEN signal asserted (high) during DMA cycles to prevent the I/O devices from misinterpreting the DMA cycle as a valid I/O cycle. The BALE signal is also driven high during DMA cycles. Also, during DMA memory read cycles to the PCI Bus, the PIIX/PIIX3 the data on the ISA Bus is considered random if the PCI cycle is either target aborted or master aborted.

NOTE

1. For type F timing mode DMA transfers, the channel must be programmed with a memory range that will be forwarded to PCI. This means that if BIOS detects that ISA memory is used in the system (i.e., that the top of memory reported to the operating system is higher than the top of memory programmed in the PIIX/PIIX3 Top of Memory register), the BIOS should not enable type F for any channel.
2. For the PIIX in External DMA mode (selected via a strapping option on the TC signal), the PIIX tri-states the AEN, TC, and DACK[7:5, 3:0]# signals, and also forwards PCI master I/O accesses to location 0000h to ISA.

3.4.1. TYPE F TIMING

Type F cycles occur back to back at a minimum repetition rate of 3 SYSCLKs. The type F cycles are always performed using the 4 byte DMA buffer. Type F transfers and the use of the DMA buffer are invoked in the MBDMAx Register. The 4 byte buffer and the type F timings may be used only when the DMA channel is programmed to increment mode (not decrement), and cannot be used when the channel is programmed to operate in block mode (single transfer mode and demand mode are legal). In addition, verify transfers are not supported with type F DMA.

The PIIX device does not support Type FDMA write timings on DMA channels, only AT compatible mode timings should be used for writes. Type FDMA read timings on DMA channels are supported by the PIIX.

3.4.2. ISA REFRESH CYCLES

Refresh cycle requests are generated by two sources—the refresh controller inside the PIIX/PIIX3 component or ISA Bus masters other than the PIIX. In both cases, the PIIX/PIIX3 generates the ISA memory refresh. The PIIX/PIIX3 enables address lines SA[7:0]. Thus, when MEMR# goes active, the entire ISA system memory is refreshed at one time. Memory slaves on the ISA Bus must not drive any data onto the data bus during the refresh cycle. The PIIX/PIIX3 maintains a four deep buffer to record internally generated refresh requests that have not been serviced. Counter 1 in the timer register set should be programmed to provide a request for refresh about every 15µs.

Initiated Refresh Cycle

The PIIX/PIIX3 asserts REFRESH# to indicate a refresh cycle and then drives the address lines SA[7:0] onto the ISA Bus and generates MEMR# and SMEMR#. The PIIX/PIIX3 drives AEN and BALE high for the entire refresh cycle. The memory device may extend this refresh cycle by pulling IOCHRDY low.

ISA Bus refresh cycles are completely decoupled from DRAM Refresh. Transactions driven by PCI masters that target ISA or IDE resources while refresh is active are held off with wait states until the refresh is complete.

ISA Master Initiated Refresh Cycle

If an ISA Bus master holds the ISA Bus longer than 15 µsec, the ISA master must initiate memory refresh cycles. If the ISA Bus master initiates a refresh cycle before it relinquishes the bus, it floats the address lines and control signals and asserts the REFRESH# to the PIIX. The PIIX/PIIX3 drives address lines SA[7:0] and MEMR# onto the ISA Bus. BALE is driven high and AEN is driven low for the entire refresh cycle.

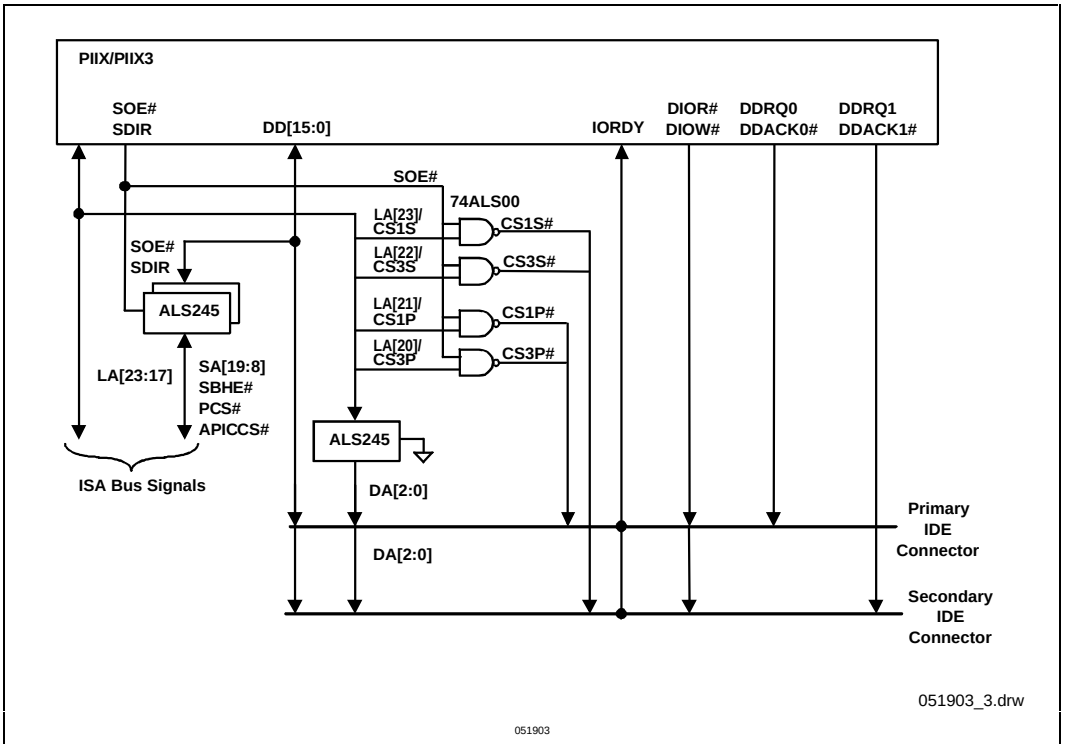
If the ISA Bus master holding the bus does not a generate a refresh request and the PIIX's internal refresh request is not serviced within the normal 15 µs, a refresh queue counter is incremented. The counter records up to four incomplete refresh cycles, which are all executed as soon as PIIX/PIIX3 gets the ISA Bus.

3.5. PCI Local Bus IDE

The PIIX/PIIX3 integrates a high performance interface from PCI to IDE. This interface is capable of accelerated PIO data transfers as well as acting as a PCI Bus master on behalf of an IDE DMA slave device. The PIIX/PIIX3 provides an interface for both primary and secondary IDE connectors (Figure 3).

The IDE data transfer command strobes, DMA request and grant signals, and IORDY signal interface directly to the PIIX. The IDE data lines interface directly to the PIIX, and are buffered to provide part of the ISA address bus as well as the X-Bus chip select signals. The IDE address and chip select signals are multiplexed onto the LA[23:17] lines. The IDE connector signals are driven from the LA[23:17] lines by an ALS244 buffer.

Only PCI masters have access to the IDE port. ISA Bus masters cannot access the IDE I/O port addresses. Memory targeted by the IDE interface acting as a PCI Bus master on behalf of IDE DMA slaves must reside on PCI, usually main memory implemented by the host-to-PCI bridge.



NOTES:

Support for Older Drives: There are cases where the PIIX/PIIX3 asserts both IDE chip selects (CS1x and CS3x). Some older drives may not operate properly when both chip selects are asserted. Because the IDE chip selects are muxed with the ISA LA lines, the 74ALS00 in the figure is used to ensure proper operation of older drives by gating the LA signals with SOE#.

Figure 3. PIIX/PIIX3 IDE Interface

Two connectors (primary and secondary) and two drives per connector (master and slave) are supported as shown in Figure 4.

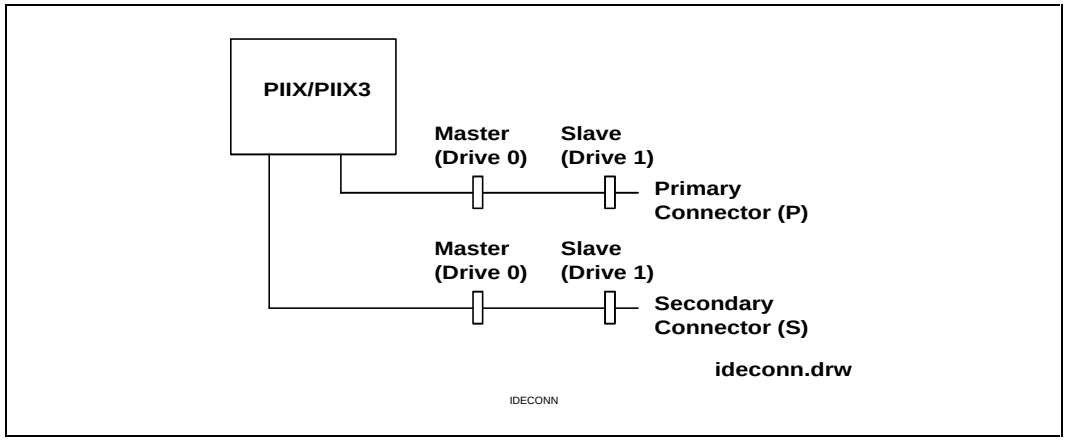


Figure 4. IDE Connector and Drive Nomenclature

3.5.1. ATA REGISTER BLOCK DECODE

The IDE ATA I/O ports are decoded by the PIIX/PIIX3 when enabled in the PCICMD and IDETIM Registers for function 1. (ATA stands for "AT Attachment"—the specification for AT compatible drive interfaces). The actual ATA registers are implemented in the drive itself. An access to the IDE registers results in the assertion of the appropriate chip select for the register. The transaction is then run using compatible timing and using the IDE command strobes (DIOR#, DIOW#).

For each cable (primary and secondary), there are two I/O ranges; the Command block that corresponds to the CS1x# chip select, and the control block that corresponds to the CS3x# chip select. The command block is an 8 byte range while the control block is a 4 byte range. The upper 16 bits of the I/O address are decoded as all 0s.

Primary Command Block Offset:	01F0h
Primary Control Block Offset:	03F4h
Secondary Command Block Offset:	0170h
Secondary Control Block Offset:	0374h

Table 13 specifies the registers as they affect the PIIX/PIIX3 hardware definition.

Table 13. IDE Legacy I/O port definition: COMMAND BLOCK (CS1x# chip select)

IO Offset	Register Function (Read / Write)	Access
00h	Data	R/W
01h	Error/Features	R/W
02h	Sector Count	R/W
03h	Sector Number	R/W
04h	Cylinder Low	R/W
05h	Cylinder High	R/W
06h	Drive/Head	R/W
07h	Status/Command	R/W

The Data Register is accessed as a 16-bit register for PIO transfers (except for ECC bytes). All other registers are accessed as 8-bit quantities.

Table 14. IDE Legacy I/O port definition: CONTROL BLOCK (CS3x# chip select)

IO Offset	Register Function (Read / Write)	Access
00h	Reserved	reserved
01h	Reserved	reserved
02h	Alt Status/Device control	R/W
03h	Forward to ISA (Floppy)	R/W

The PIIX/PIIX3 claims all accesses to these ranges. The byte enables do not have to be externally decoded to assert DEVSEL#. Accesses to byte 3 of the Control Block are forwarded to ISA where the floppy disk controller responds.

Each of the two drives (drive 0 or 1) on a cable implement separate ATA register files. To determine the targeted drive, the PIIX/PIIX3 shadows the value of bit 4 (drive bit) of byte 6 (drive/head register) of the ATA command block (CS1x#) for each of the two IDE connectors (primary and secondary).

3.5.2. ENHANCED TIMING MODES

The PIIX/PIIX3 includes fast timing modes. The fast timing modes can be enabled only for the IDE data ports. All other transactions to the IDE registers are run in single transaction mode with compatible timings.

Up to 2 IDE devices may be attached per IDE connector (drive 0 and drive 1). For each connector in a PIIX system, only one fast timing mode may be specified (via the IDETIM Register). This mode can be applied to drive 0, drive 1, or both. Transactions targeting the other drive will use compatible timing.

For the PIIX3, the IDETIM and SIDETIM Registers permit different timing modes to be programmed for drive 0 and drive 1 of the same connector.

3.5.2.1. Back-To-Back PIO IDE Transactions

IDE data port transaction latency consists of startup latency, cycle latency, and shutdown latency. Cycle latency consists of the I/O strobe assertion length and recovery time. Recovery time is provided so that transactions may occur back-to-back on the IDE interface (without incurring startup and shutdown latency) without violating minimum cycle periods for the IDE interface.

Startup latency is incurred when a PCI master cycle targeting the IDE data port is decoded and the DA[2:0] and CSxx# lines are not set up. Startup latency provides the setup time for the DA[2:0] and CSxx# lines with respect to the read and write strobes (DIOR# and DIOW#).

Shutdown latency is incurred after outstanding scheduled IDE data port transactions (either a non-empty write post buffer or an outstanding read prefetch cycles) have completed and before other transactions can proceed. It provides hold time on the DA[2:0] and CSxx# lines with respect to the read and write strobes (DIOR# and DIOW#).

Cycle latency is the latency incurred by each individual 16-bit IDE data port transfer, and consists of command strobe width and recovery time. The command strobe assertion width is selected by the IDETIM Register and may be set to 2, 3, 4, or 5 PCI clocks. The recovery time is selected by the IDETIM Register and may be set to 1, 2, 3, or 4 PCI clocks.

For the PIIX3, the master drive (drive 1) and slave drive (drive 0) can be programmed to different command strobe assertion widths and recovery times via the IDETIM and SIDETIM Registers.

If IORDY is asserted when the initial sample point is reached, no wait states are added. If IORDY is negated when the initial sample point is reached, additional wait states are added. Since the rising edge of IORDY must be synchronized, at least two additional PCI clocks are added.

NOTE

Bit 2 (16-bit I/O recovery enable) of the ISA Controller Recovery Timer Register does not add wait states to IDE data port read accesses when any of the fast timing modes are enabled.

3.5.2.2. IORDY Masking

The IORDY signal can be forced asserted on a drive by drive basis via the IDETIM Register.

3.5.2.3. PIO 32 Bit IDE Data Port Mode

If the 32-bit IDE data port mode is enabled (via bit 4 and 0 of the IDETIM Register), 32-bit accesses to the IDE data port address (default 01F0h primary, etc.) result in two back to back 16-bit transactions to IDEThe 32-bit data port feature is enabled for all timings, not just enhanced timing.

3.5.3. BUS MASTER FUNCTION

The PIIX/PIIX3 can act as a PCI Bus master on behalf of an IDE slave device. Two PCI Bus master channels are provided—one channel for each IDE connector (primary and secondary). By performing the IDE data transfer as a PCI Bus master, the PIIX/PIIX3 off-loads the CPU and improves system performance in multitasking environments.

Physical Region Descriptor Format

The physical memory region to be transferred is described by a Physical Region Descriptor (PRD). The PRDs are stored in a table in memory. The data transfer proceeds until all regions described by the PRDs in the table have been transferred. Note that the bus master IDE does not support memory for regions or PRDs on ISA.

Each PRD entry is 8 bytes in length. PRDs must be aligned on 64-Kbyte boundaries. The first 4 bytes specify the byte address of a physical memory region. The next two bytes specify the count of the region in bytes (64-Kbyte limit per region). A value of zero in these two bytes indicates 64 Kbytes. Bit 7 of the last byte indicates the end of the table (EOT). Bus master operation terminates when the last descriptor has been retired.

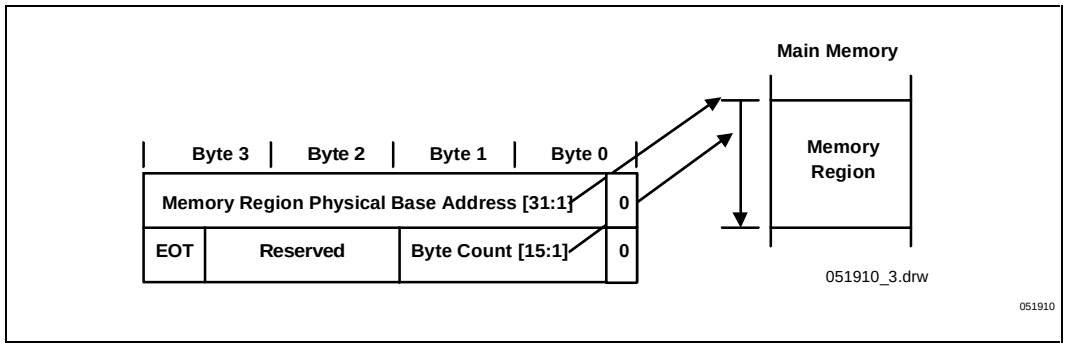


Figure 5. Physical Region Descriptor Table Entry

NOTE

The memory region specified by the descriptor cannot straddle a 64-Kbyte boundary. This means that the byte count can be limited to 64 Kbytes and the incrementer for the current address register need only extend from bit 1 to bit 15. Also, the total sum of the descriptor byte counts must be equal to or greater than the size of the disk transfer request. If greater than the disk transfer request, the driver must terminate the bus master transaction (by setting bit 0 in the Bus Master IDE Command Register to 0) when the drive issues an interrupt to signal transfer completion.

Operation

To initiate a bus master transfer between memory and an IDE DMA slave device, the following steps are required:

1. Software prepares a PRD Table in main memory. Each PRD is 8 bytes long and consists of an address pointer to the starting address and the transfer count of the memory buffer to be transferred. In any given PRD Table, two consecutive PRDs are offset by 8-bytes and are aligned on a 4-byte boundary.
2. Software provides the starting address of the PRD Table by loading the PRD Table Pointer Register . The direction of the data transfer is specified by setting the Read/Write Control bit. Clear the Interrupt bit and Error bit in the Status register.
3. Software issues the appropriate DMA transfer command to the disk device.
4. Engage the bus master function by writing a 1 to the Start bit in the Bus Master IDE Command Register for the appropriate channel. The first entry in the PRD table is fetched by the PIIX. The channel remains masked until the first descriptor is loaded.
5. The controller transfers data to/from memory responding to DMA requests from the IDE device.
6. At the end of the transfer, the IDE device signals an interrupt.
7. In response to the interrupt, software resets the Start/Stop bit in the command register. It then reads the controller status and then the drive status to determine if the transfer completed successfully.

Note that the IRQ14 signal must be used to signal interrupts for the primary channel in bus master mode; MIRQ0 must be used for the secondary channel.

When the last data transfer for a region has been completed on the IDE interface, the next descriptor is fetched from the table. The descriptor contents are loaded into the Current Base and Current Count registers.

The last PRD in a table has the End of List (EOL) bit set. The PCI Bus master data transfers terminates when the physical region described by the last PRD in the table has been completely transferred. The active bit in the BMISx Register is set to 0 and the DDRQx signal is masked.

Line Buffer

A single line buffer exists for the PCI Bus master IDE interface. This buffer is not shared with any other function. The buffer is maintained in either the read state or the write state. Writes are always 4-DWord bursts and invalid DWords have C/BE[3:0]#=F. The size of the buffer is 32 bytes, and is aligned on the cache line boundary. The line buffer allows burst data transfers to proceed at peak transfer rates.

3.6. Universal Serial Bus Host Controller (PIIX3 only)

PIIX3 contains a USB Host Controller (HC). The Host Controller includes the root hub with two USB ports. This permits connection of two USB peripheral devices directly to the PIIX3 without an external hub. If more devices are required, an external hub can be connected to either of the built-in ports. The USB's PCI configuration registers are located in function 2, PCI configuration space. The PIIX3 Host Controller completely supports the standard Universal Host Controller Interface (UHCI) and thus, takes advantage of the standard software drivers written to be compatible with UHCI. Figure 6 Shows a conceptual view of a USB system. UHCI consists of two parts— Host Controller Driver (HCD) and Host Controller (HC). The Host Controller interfaces to the USB system software in the host via the HCD. The HCD software manages the Host Controller operation and is responsible for scheduling the traffic on USB by posting and maintaining transactions in system memory. HCD is part of the system software and is typically provided by the operating system vendor. HCD provides the software layer between the PIIX3's Host Controller and the USBD software layer (also located in the operating system). The UHCI's HCD software interprets requests from the USBD and builds Frame List, Transfer Descriptor, Queue Head, and data buffer data structures for the Host Controller. The data structures are built in system memory and contain all necessary information to provide end-to-end communication between client software in the host and devices on the USB.

The PIIX3's Host Controller moves data between system memory and devices on the USB by processing these data structures and generating the transaction on USB. The Host Controller executes the schedule lists generated by HCD and reports the status of transactions on the USB to HCD. Command execution includes generating serial bus token and/or data packets based on the command and initiating transmission on USB. For commands that require the Host Controller to receive data from the USB device, the Host Controller receives the data and then transfers it to the system memory pointed to by the command. The UHCI's HCD provides sufficient commands and data to keep ahead of the Host Controller execution and analyzes the results as the commands are completed.

The PIIX3 provides support for USB Keyboard and Mouse operation with an operating system environment which does not have the complete USB software drivers loaded. This allows application programs which directly access system I/O space to function with a USB keyboard or mouse. The USB system will emulate a 8042 based keyboard controller.

The PIIX3 has five controller issues regarding USB transactions, each of which is described below.

Multiple USB Babble Interrupts may be observed. 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). This means extra hardware interrupts can be generated to the system. These interrupts are treated as spurious interrupts by the HCD interrupt handler.

A low speed length babble occurring 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. An EOF babble occurs when a device transmits past the frame EOF time point. This condition does not exist if an EOF babble occurs. If this occurs, the USB host controller can not be restarted, disabling the system from performing any further USB transactions. 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 occurrence 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.

Low Speed Length Babble will cause USB Port Disable. If a low speed length babble (see description 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. This means the USB port which is disabled will not receive subsequent packets until the port is reenabled by software. This condition can be corrected using the procedure described above.

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. This means 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. 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.

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. There is no visual effect on functionality of performance. An alternative method of status monitoring in software is necessary for immediate detection. 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 and 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.

More information concerning the PIIX3 USB Host Controller operation can be found in the Universal Host Controller Interface Design Guide, available from Intel Literature Center as document number 297650.

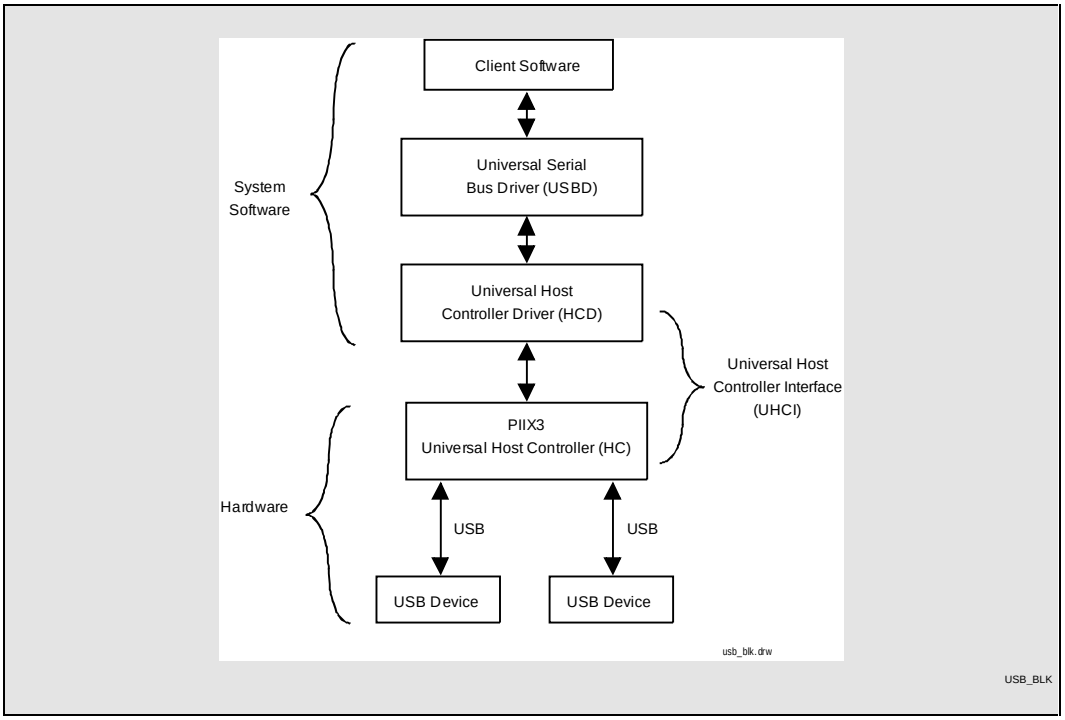


Figure 6. USB System Conceptual View

3.7. Interval Timer

The PIIX/PIIX3 contains three counters that are equivalent to those found in the 82C54 programmable interval timer. The three counters are contained in one PIIX/PIIX3 timer unit, referred to as Timer-1. Each counter output provides a key system function. Counter 0 is connected to interrupt controller IRQ0 and provides a system timer interrupt for a time-of-day, diskette time-out, or other system timing functions. Counter 1 generates a refresh request signal and Counter 2 generates the tone for the speaker. The 14.31818 MHz counters normally use OSC as a clock source.

Counter 0, System Timer

This counter functions as the system timer by controlling the state of IRQ0 and is typically programmed for mode 3 operation. The counter produces a square wave with a period equal to the product of the counter period (838 ns) and the initial count value. The counter loads the initial count value one counter period after software writes the count value to the counter I/O address. The counter initially asserts IRQ0 and decrements the count value by two each counter period. The counter negates IRQ0 when the count value reaches 0. It then reloads the initial count value and again decrements the initial count value by two each counter period. The counter then asserts IRQ0 when the count value reaches 0, reloads the initial count value, and repeats the cycle, alternately asserting and negating IRQ0.

Counter 1, Refresh Request Signal

This counter provides the refresh request signal and is typically programmed for mode 2 operation. The counter negates refresh request for one counter period (838 ns) during each count cycle. The initial count value is loaded one counter period after being written to the counter I/O address. The counter initially asserts refresh request, and negates it for 1 counter period when the count value reaches 1. The counter then asserts refresh request and continues counting from the initial count value.

Counter 2, Speaker Tone

This counter provides the speaker tone and is typically programmed for mode 3 operation. The counter provides a speaker frequency equal to the counter clock frequency (1.193 MHz) divided by the initial count value. The speaker must be enabled by a write to port 061h.

3.8. Interrupt Controller

The PIIX/PIIX3 provides an ISA compatible interrupt controller that incorporates the functionality of two 82C59 interrupt controllers. The two controllers are cascaded so that 13 external and three internal interrupts are possible. The master interrupt controller provides IRQ [7:0] and the slave interrupt controller provides IRQ [15:8] (Figure 7). The three internal interrupts are used for internal functions only and are not available to the user. IRQ2 is used to cascade the two controllers together. IRQ0 is used as a system timer interrupt and is tied to Interval Timer 1, Counter 0. The MIRQ0/IRQ0 pin will function as the IRQ0 output and should be connected to the INTIN2 input of the IOAPIC when IRQ0 enable bit is set in the MIRQ0 register. IRQ13 is connected internally to FERR#. The remaining 13 interrupt lines (IRQ[15,14,12:3,1]) are available for external system interrupts. Edge or level sense selection is programmable on an individual channel by channel basis.

The Interrupt unit also supports interrupt steering. The PIIX/PIIX3 can be programmed to allow the four PCI active low interrupts (PIRQ[D:A]#) to be internally routed to one of 11 interrupts (IRQ[15,14,12:9,7:3]). In addition, the motherboard interrupts (MIRQ[1:0] for PIIX and MIRQ0 for PIIX3) may be routed to any of the 11 interrupts.

The Interrupt Controller consists of two separate 82C59 cores. Interrupt Controller 1 (CNTRL-1) and Interrupt Controller 2 (CNTRL-2) are initialized separately and can be programmed to operate in different modes. The default settings are: 80x86 Mode, Edge Sensitive (IRQ[15:0]) Detection, Normal EOI, Non-Buffered Mode, Special Fully Nested Mode disabled, and Cascade Mode. CNTRL-1 is connected as the Master Interrupt Controller and CNTRL-2 is connected as the Slave Interrupt Controller.

Note that IRQ13 is generated internally (as part of the coprocessor error support) by the PIIX. IRQ12/M is generated internally (as part of the mouse support) when bit-4 in the XBCS Register is set to a 1. When this bit is set to a 0, the standard IRQ12 function is provided and IRQ12 appears externally.

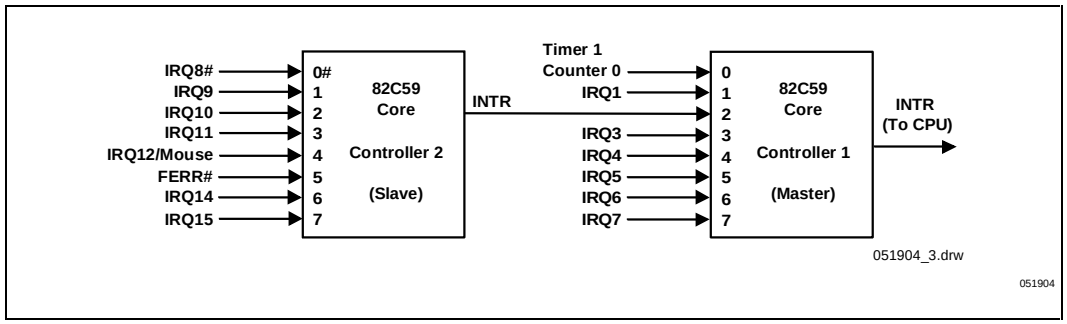


Figure 7. Block Diagram of the Interrupt Controller

3.8.1. PROGRAMMING THE ICWs/OCWs

The Interrupt Controller accepts two types of command words generated by the CPU or bus master:

1. Initialization Command Words (ICWs): Before normal operation can begin, each Interrupt Controller in the system must be initialized. In the 82C59, this is a two to four byte sequence. However, for the PIIX, each controller must be initialized with a four byte sequence. This four byte sequence is required to configure the interrupt controller correctly for the PIIX/PIIX3 implementation. This implementation is ISA-compatible. The four initialization command words are referred to by their acronyms: ICW1, ICW2, ICW3, and ICW4. The base address for each interrupt controller is a fixed location in the I/O memory space, at 0020h for CNTRL-1 and at 00A0h for CNTRL-2.

An I/O write to the CNTRL-1 or CNTRL-2 base address with data bit 4 equal to 1 is interpreted as ICW1. For PIIX-based ISA systems, three I/O writes to "base address + 1" (021h for CNTRL-1 and 0A1h for CNTRL-2) must follow the ICW1. The first write to "base address + 1" (021h/0A1h) performs ICW2, the second write performs ICW3, and the third write performs ICW4.

2. Operation Command Words (OCWs): These are the command words that dynamically reprogram the interrupt controller to operate in various interrupt modes. Any interrupt lines can be masked by writing an OCW1. A 1 written in any bit of this command word masks incoming interrupt requests on the corresponding IRQx line. OCW2 is used to control the rotation of interrupt priorities when operating in the rotating priority mode and to control the End of Interrupt (EOI) function of the controller. OCW3 set up reads of the ISR and IRR, enable/disables the Special Mask Mode (SMM), and sets up the interrupt controller in polled interrupt mode. The OCWs can be written to the Interrupt Controller any time after initialization.

3.8.2. EDGE AND LEVEL TRIGGERED MODE

In ISA systems this mode is programmed using bit 3 in ICW1. With PIIX/PIIX3 this bit is disabled and a new register for edge and level triggered mode selection (per interrupt input) is included. This is the Edge/Level control Registers ELCR1 and ELCR2. The default programming is equivalent to programming the LTIM bit (ICW1 bit 3) to a 0 (all interrupts selected for edge triggered mode). Note, that IRQ0, 1, 2, 8#, and 13 can not be programmed for level sensitive mode and can not be modified by software.

If an ELCR bit = 0, an interrupt request is recognized by a low to high transition on the corresponding IRQx input. The IRQ input can remain high without generating another interrupt.

If an ELCR bit = 1, an interrupt request is recognized by a high level on the corresponding IRQ input and there is no need for an edge detection. The interrupt request must be removed before the EOI command is issued to prevent a second interrupt from occurring.

In both the edge and level triggered modes, the IRQ inputs must remain asserted until the cycle starts (PCI INTA cycle); with FRAME# and IRDY# both asserted. After that, the IRQ may be removed. If the IRQ input goes inactive before this time, a default IRQ7 occurs when the CPU acknowledges the interrupt. This can be a useful safeguard for detecting interrupts caused by spurious noise glitches on the IRQ inputs. To implement this feature, the IRQ7 routine is used for "clean up" simply executing a return instruction, thus ignoring the interrupt. If IRQ7 is needed for other purposes, a default IRQ7 can still be detected by reading the ISR. A normal IRQ7 interrupt sets the corresponding ISR bit. A default IRQ7 does not set this bit. However, If a default IRQ7 routine occurs during a normal IRQ7 routine, the ISR remains set. In this case, it is necessary to keep track of whether or not the IRQ7 routine was previously entered. If another IRQ7 occurs, it is a default.

3.8.3. INTERRUPT STEERING

The PIIX/PIIX3 can be programmed to allow four PCI programmable interrupts (PIRQ[D:A]#) to be internally routed to one of 11 interrupts (IRQ[15,14,12:9,7:3]) using the PIRQx Route Control Register. PCLK is used to synchronize the PIRQx# inputs. The assignment is programmable through the PIRQx Route Control

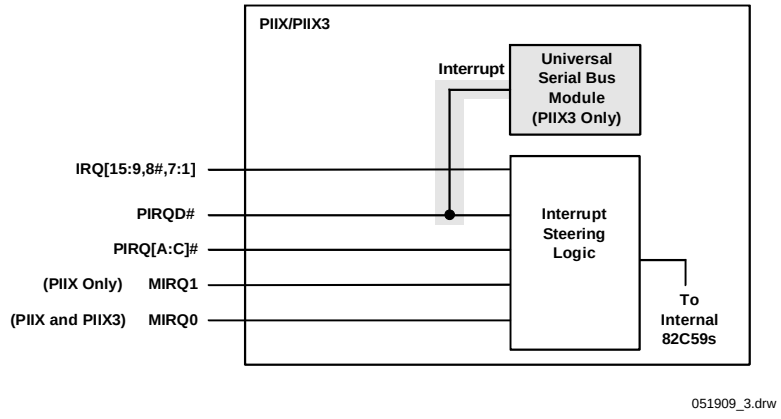
registers. One or more PIRQx# lines can be routed to the same IRQx input. If interrupt steering is not required, the Route Registers can be programmed to disable steering.

For the PIIX3, the Universal Serial Bus (USB) Module interrupt is hardwired to the PIRQD# signal. When the USB is enabled, the USB Host Controller in PIIX3 uses the PIRQD# input to the internal interrupt controller. If an IOAPIC is used, the USB outputs interrupts on PIRQD# to the external IOAPIC. When the USB function is disabled, other PCI devices can use the PIRQD# input.

The PIRQx# lines are defined as active low, level sensitive to allow multiple interrupts on a PCI Board to share a single line across the connector. When a PIRQx# is routed to a specified IRQ line, the software must change the IRQ's corresponding ELCR bit to level sensitive mode. Note, that this means that the selected IRQ can no longer be used by an ISA device, unless that ISA device can respond as an active low level sensitive interrupt.

The 82371FB PIIX supports up to two programmable interrupts (MIRQ[1:0]; intended for use with motherboard devices) to be routed to one of the 11 interupts (IRQ[15,14,12:9,7:3]) using the MBIRQx Route Control Register. The routing is accomplished in the same manner as for the PIRQx# inputs, except that the interrupts are active high. Two MIRQx lines may be routed to the same IRQx input. If interrupt steering is not required, the MBIRQx registers can be programmed to disable routing. Note that only one motherboard interrupt (MIRQ0) is available on PIIX3.

For the 82371FB PIIX , when more than one MIRQ line is routed to an IRQ input, the software must change the IRQ's corresponding ELCR bit to level sensitive mode. Interrupt sharing for motherboard devices must be evaluated for the particular combination of devices under consideration. The IRQ selected bit MBIRQx[3:0] can no longer be used by an ISA device, unless that ISA device can respond as an active high level sensitive interrupt.



Note: MIRQ1 is only availabe on PIIX. MIRQ0 is available on both PIIX and PIIX3.

Figure 8. Interrupt Steering

3.9. Stand-Alone IOAPIC Support (PIIX3)

The PIIX3 supports a stand-alone IOAPIC device on the ISA X-Bus. The PIIX3 provides a chip select signal (APICCS#) for the IOAPIC. It also provides handshake signals to maintain buffer coherency in the IOAPIC environment.

APICCS# is generated when the PCI memory cycle address matches the APIC's programmed address and the APICCS# function is enabled in the XBCS Register. The APIC address can be relocated by programming the APIC Base Address Register (APICBASE).

APICCS# is only generated for PCI originated cycles and is not generated for ISA originated cycles. This PCI cycle is forwarded to the ISA bus. To avoid address aliasing conflicts with other ISA devices, PIIX3 drives SA[19:16] and LA[23:17] to 0 and drives SA[15:0] corresponding to PCI AD[15:2] and C/BE[3:0]#.

When the APICCS# function is enabled, the XOE#/XDIR# signals controlling the X-bus tranceiver and the SOE#/SDIR signals controlling the IDE DD isolation transceiver are also enabled during accesses to the IOAPIC.

The IOAPIC signals (APICCS#, APICREQ#, and APICACK#) are multiplexed with DD14, TESTIN#, and PCIRST#, respectively. Figure 9 shows how these signals are connected in systems with and without the IOAPIC device.

The internal IRQ0 signal can be routed to the external pin MIRQ0 using bit 5 in the MBIRQ Route Control Register 0. This changes MIRQ0 to an output signal and allows the IRQ0 signal to be connected to the external IO-APIC. The secondary IDE device interrupt should then be routed to IRQ15.

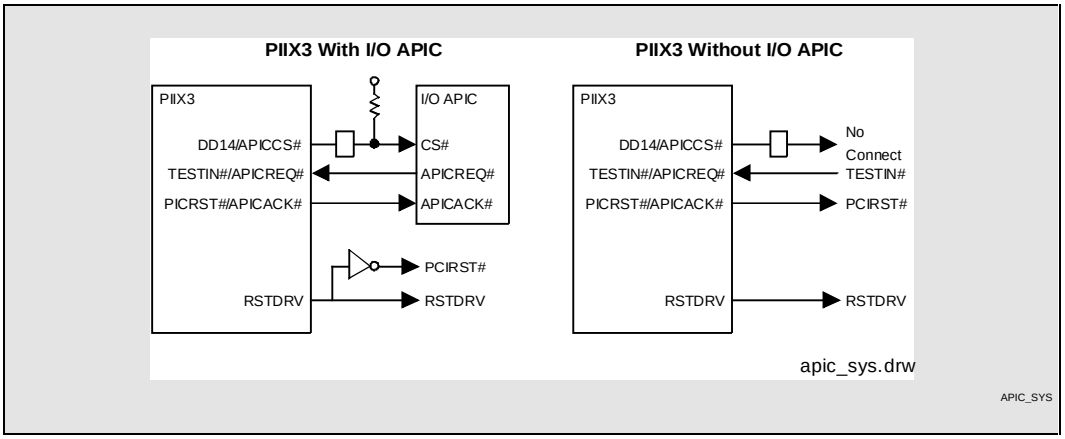


Figure 9. APIC Signal Connections

3.10. 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 which use Local APIC in Virtual Wire mode of operation. This can include Intel 430HX PCIset based 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 Intel 440FX PCIset based 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'. 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 with uses trap gate interrupt handling latency times.
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.

3.11. X-Bus Peripheral Support

The PIIX/PIIX3 provides positive decode (chip selects) and X-Bus buffer control (XDIR# and XOE#) for a real time clock, keyboard controller and BIOS for PCI and ISA initiated cycles. The PIIX/PIIX3 also generates RTCALE (address latch enable) for the RTC. The chip selects are generated combinatorially from the ISA SA(16:0) and LA (23:17) address lines (Note that it is assumed that ISA masters drive SA(19:16) and LA(23:17) low when accessing I/O devices). The PIIX/PIIX3 also provides PS/2 mouse support via the IRQ12/M signal and coprocessor functions (FERR# and IGNNE#). The chip selects and X-Bus buffer control lines can be enabled/disabled via the XBCS Register.

Coprocessor Error Function

This function provides coprocessor error support for the CPU and is enabled via the XBCS Register. FERR# is tied directly to the coprocessor error signal of the CPU. If FERR# is driven active to the PIIX, an internal IRQ13 is generated and an the INTR output from the PIIX/PIIX3 is driven active. When a write to I/O location F0h is detected, the PIIX/PIIX3 negates IRQ13 (internal to the PIIX) and drives IGNNE# active. IGNNE# remains active until FERR# is driven inactive. Note, that IGNNE# is not driven active unless FERR# is active.

Mouse Function

When the mouse interrupt function is enabled (via the XBCS Register), the mouse interrupt function is provided on the IRQ12/M input signal. In this mode, a mouse interrupt generates an interrupt through IRQ12 to the Host CPU. The PIIX/PIIX3 informs the CPU of this interrupt via a INTR. A read of 60h releases IRQ12. If the mouse interrupt function is disabled, a read of address 60h has no effect on IRQ12/M. Reads and writes to this register flow through to the ISA Bus. For additional information, see the IRQ12/M description in the Signal Description.

3.12. Power Management

The PIIX/PIIX3 has extensive power management capability permitting a system to operate in a low power state without being powered down. In a typical desktop personal computer there are two states — Power On and Power-Off. Leaving a system powered on when not in use wastes power. The PIIX/PIIX3 provides a Fast On/Off feature that creates a third state called Fast Off (Figure 10). When in the Fast Off state, the system consumes less power than the Power-On state.

The PIIX's power management architecture is based on three functions—System Management Mode (SMM), Clock Control, and Advanced Power Management (APM). Software (called SMM code) controls the transitions between the Power On state and the Fast Off state. The PIIX/PIIX3 invokes this software by generating an SMI to the CPU (asserting the SMI# signal). A variety of programmable events are provided that can generate an SMI. The SMM code places the system in either the Power On state or the Fast Off state.

A Fast On event is an event that instructs the computer (via an SMI to the CPU) to enter the Power-On state in anticipation of system activity by the user. Fast On events are programmable and include moving the mouse, pressing a key on the keyboard, an external hardware event, an incoming call to a system FAX/Modem, a RTC alarm, or the operating system.

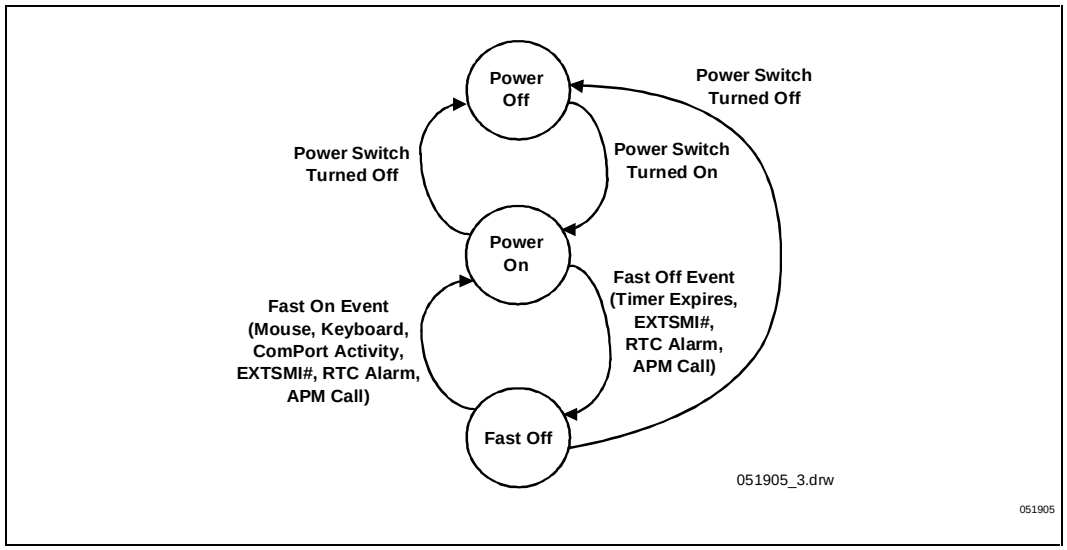


Figure 10. Fast On/Off Flow

3.12.1. SMM MODE

SMM mode is invoked by asserting the SMI# signal to the CPU. The PIIX/PIIX3 provides a variety of programmable events that can generate an SMI. When the CPU receives an SMI, it enters SMM mode and executes SMM code out of SMRAM. Depending on the current state, the SMM code places the system in either the Power On state or the Fast Off state. In the Power On state, the computer system operates normally. In this state one of the four programmable events listed below can trigger an SMI.

1. A global idle timer called the Fast Off timer expires (an indication that the end user has not used the computer for a programmed period of time).
2. The EXTSMI# pin is asserted.
3. The operating system issues an APM call.

3.12.2. SMI SOURCES

The SMI# signal can be asserted by hardware interrupt events, the Fast Off Timer, an external SMI event (EXTSMI#), and software events (via the APMC and APMS Registers). Enable/disable bits (in the SMIEN Register) permit each event to be individually masked from generating an SMI. In addition, the SMI# signal can be globally enabled/disabled in the SMICNTL Register. Status of the individual events causing an SMI is provided in the SMIREQ Register. For detailed information on the SMI control/status registers, refer to the Register Description section.

Hardware Interrupt Events

Hardware events (IRQ[12,8#,4,3,1] and the Fast Off Timer) are enabled/disabled from generating an SMI in the SMIEN Register. When enabled, the occurrence of the corresponding hardware event generates an SMI (asserts the SMI# signal), regardless of the current power state of the system.

Fast Off Timer

The Fast Off Timer is used to indicate (through an SMI) that the system has been idle for a programmed period of time. The timer counts down from a programmed start value and when the count reaches 00h, can generate an SMI. The timer decrement rate is programmable (via the SMICNTL Register) and is re-loaded each time a system event occurs. This counter should not be programmed to 00h. System and break events are described in the SEE Register.

EXTSMI#

The EXTSMI# input pin provides the system designer the capability to invoke SMM with external hardware. For example, the EXTSMI# input could be connected to a "green button" permitting the user to enter the Fast Off state by depressing a button. The EXTSMI# generation of an SMI is enabled/disabled in the SMIEN Register.

For the PIIX3, the EXTSMI# signal can be used to provide a special protocol between the host-to-PCI bridge and the PIIX3 (see MSTAT Register description, 6B-6Ah, function 0).

Software Events

Software events (accessing the APMx Registers) indicate that the OS is passing power management information to the SMI handler. There are two Advanced Power Management (APM) registers — APM Control (APMC) and APM Status (APMS) Registers. These registers permit software to generate an SMI; by writing to the APMC Register. For example, the APMC can be used to pass an APM command between APM OS and BIOS and the APMS Register could be used to pass data between the OS and the SMI handler.

The two APM Registers are located in normal I/O space. The PIIX/PIIX3 subtractively decodes PCI accesses to these registers and forwards the accesses to the ISA Bus. The APM Registers are not accessible by ISA masters. Note that the remaining power management registers are located in PCI configuration space.

3.12.3. CLOCK CONTROL

The CPU can be put in a low power state by asserting the STPCLK# signal. STPCLK# is an interrupt to the CPU. However, for this type of interrupt, the CPU does not generate an interrupt acknowledge cycle. Once the STPCLK# interrupt is executed, the CPU enters the stop grant state. In this state, the CPU's internal clocks are disabled and instruction execution is stopped. The stop grant state is exited when the STPCLK# signal is negated.

Software can assert STPCLK#, if enabled via the SMICNTL Register, by a read of the APMC Register. Note that STPCLK# can also be periodically asserted by using clock scaling as described below.

The PIIX/PIIX3 automatically negates STPCLK# when a break event occurs (if enabled in the SEE Register). Software can negate STPCLK# by disabling STPCLK# in the SMICNTL Register or by a write to the APMC Register.

Clock Scaling (Emulating Clock Division)

Clock scaling permits the PIIX/PIIX3 to periodically place the CPU in a low power state. This emulates clock division. When clock scaling is enabled, the CPU runs at full frequency for a pre-defined time period and then is stopped for a pre-defined time period. The run/stop time interval ratio emulates the clock division effect from a power/performance point of view. However, clock scaling is more effective than dividing the CPU frequency. For example, if the CPU is in the stop grant state and a break event occurs, the CPU clock returns to full frequency. In addition, there is no recovery time latency to start the clock.

Two programmable 8-bit clock scale timer control registers set the STPCLK# high (negate) and low (assert) times — the CTLTMRH and CTLTMRL Registers. The timer is clocked by a 32 usec internal clock. This allows a programmable timer interval for both the STPCLK# high and low times of 0-8 msec.

3.13. Reset Support

The PIIX/PIIX3 integrates the system reset logic for the system. The PIIX/PIIX3 generates CPURST, PCIRST#, and RSTDRV during power up (PWROK) and when a hard reset is initiated through the RC register.

For the PIIX3, the PCIRST# signal is multiplexed with APICACK#. When an external IOAPIC is used, the PCIRST# functionality is provided by externally inverting the RSTDRV signal (see Stand-Alone IOAPIC Support section).

The following PIIX/PIIX3 signals interface directly to the processor:

- CPURST
- INTR
- NMI
- IGNNE#
- SMI#
- STPCLK#

These signals are open drain. Thus, external logic is not required for interfacing with the processors based on 3.3V technology which do no support 5V tolerant input buffers. During power-up these signals are driven low to prevent problems associated with 5V/3.3V power sequencing.

Some PCI devices may drive 3.3V friendly signals directly to 3.3V devices that are not 5V tolerant. If such signals are powered from the 5V supply they must be driven low when PCIRST# is asserted. Some of these signals may need to be driven high before CPURST is negated. PCIRST# is negated 1 ms to 2 ms before CPURST to allow time for this to occur.

3.13.1. HARDWARE STRAPPING OPTIONS

For the PIIX/PIIX3, the SYSCLK signal is used during a hard reset to select the ISA clock divisor (sampled high for divisor of 3 – 25 MHz PCI operation. Sampled low for a divisor of 4 – 33 MHz or 30 MHz PCI operation).

A 10K ohm resistor, jumperable as a pull up or pull down, on SYSCLK, can be used to configure divide-by-3 or divide-by-4 SYSCLK frequency. The logic level sensed on SYSCLK at reset configures the frequency. An alternate solution is using an open drain buffer enabled only during reset to drive the appropriate level on SYSCLK.

For the PIIX, a strapping option on the TC during a hard reset selects between ISA DMA mode and an external DMA mode (sampled high for ISA DMA mode and low for external DMA mode).

4.0. PINOUT AND PACKAGE INFORMATION

4.1. Pinout

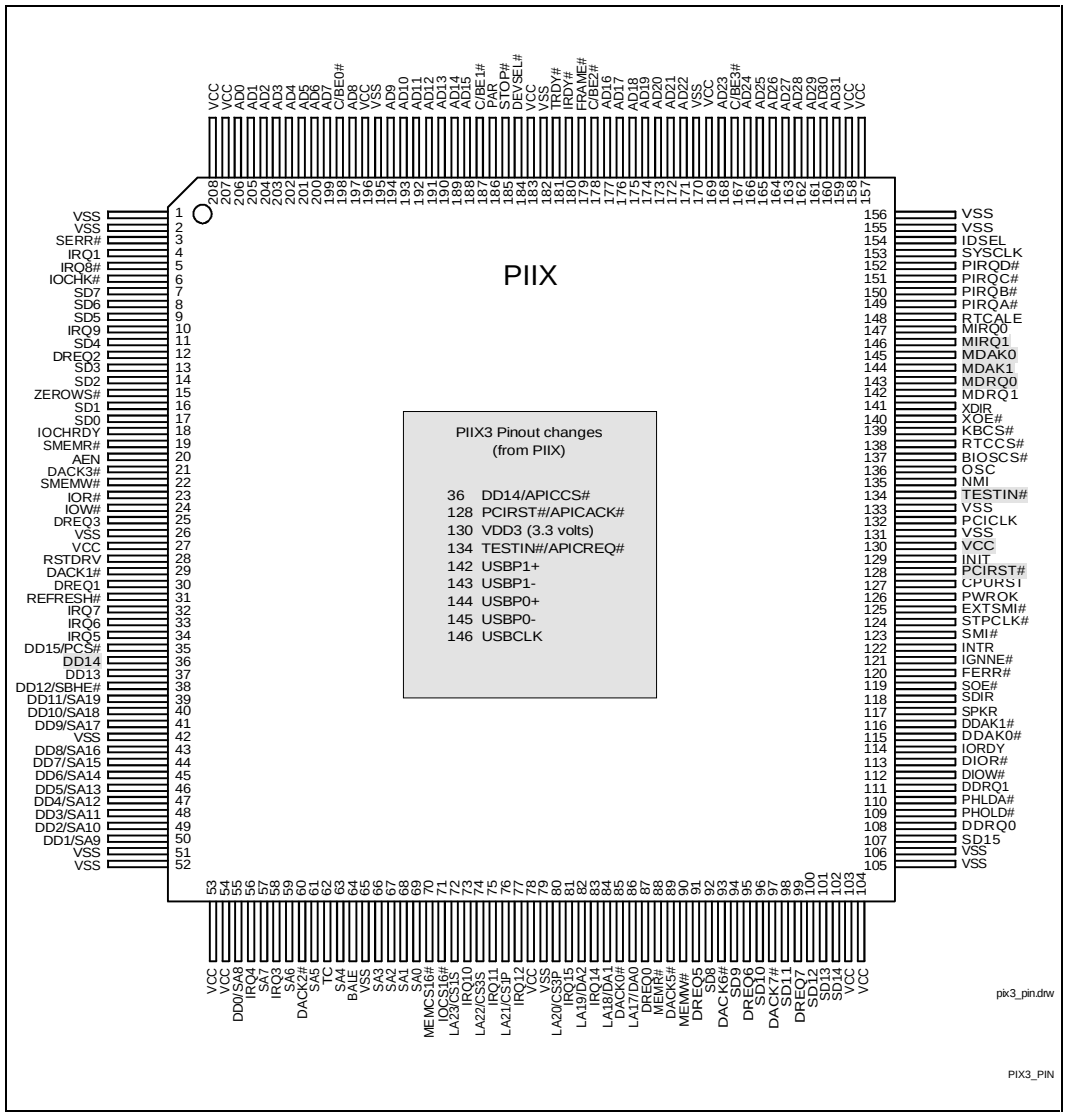


Figure 11. PIIX/PIIX3 Pin Assignment

Table 15. Alphabetical Pin Assignment

Name	Pin #	Type
AD0	206	I/O
AD1	205	I/O
AD2	204	I/O
AD4	202	I/O
AD5	201	I/O
AD6	200	I/O
AD7	199	I/O
AD8	197	I/O
AD9	194	I/O
AD10	193	I/O
AD11	192	I/O
AD12	191	I/O
AD13	190	I/O
AD14	189	I/O
AD15	188	I/O
AD16	177	I/O
AD17	176	I/O
AD18	175	I/O
AD19	174	I/O
AD20	173	I/O
AD21	172	I/O
AD22	171	I/O
AD23	168	I/O
AD24	166	I/O
AD25	165	I/O
AD26	164	I/O
AD27	163	I/O
AD28	162	I/O
AD29	161	I/O
AD3	203	I/O

Name	Pin #	Type
AD30	160	I/O
AD31	159	I/O
AEN	20	O
BALE	64	O
BIOSCS#	137	O
C/BE0#	198	I/O
C/BE1#	187	I/O
C/BE2#	178	I/O
C/BE3#	167	I/O
CPURST	127	O
DACK0#	85	O
DACK1#	29	O
DACK2#	60	O
DACK3#	21	O
DACK5#	89	O
DACK6#	93	O
DACK7#	97	O
DD0/SA8	55	I/O
DD1/SA9	50	I/O
DD2/SA10	49	I/O
DD3/SA11	48	I/O
DD4/SA12	47	I/O
DD5/SA13	46	I/O
DD6/SA14	45	I/O
DD7/SA15	44	I/O
DD8/SA16	43	I/O
DD9/SA17	41	I/O
DD10/SA18	40	I/O
DD11/SA19	39	I/O
DD12/SBHE#	38	I/O

Name	Pin #	Type
DD13	37	I/O
DD14 (PIIX) DD14/APICCS# (PIIX3)	36	I/O
DD15/PCS#	35	I/O
DDAK0#	115	O
DDAK1#	116	O
DDRQ0	108	I
DDRQ1	111	I
DEVSEL#	184	I/O
DIOR#	113	O
DIOW#	112	O
DREQ0	87	I
DREQ1	30	I
DREQ2	12	I
DREQ3	25	I
DREQ5	91	I
DREQ6	95	I
DREQ7	99	I
EXTSMI#	125	I
FERR#	120	I
FRAME#	179	I/O
IDSEL	154	I
IGNNE#	121	O
INIT	129	O
INTR	122	O
IOCHK#	6	I
IOCHRDY	18	I/O
IOCS16#	71	I
IOR#	23	I/O
IORDY	114	I
IOW#	24	I/O

Name	Pin #	Type
IRDY#	180	I/O
IRQ1	4	I
IRQ3	58	I
IRQ4	56	I
IRQ5	34	I
IRQ6	33	I
IRQ7	32	I
IRQ8#	5	I
IRQ9	10	I
IRQ10	73	I
IRQ11	75	I
IRQ12	77	I
IRQ14	83	I
IRQ15	81	I
KBCS#	139	O
LA17/DA0	86	I/O
LA18/DA1	84	I/O
LA19/DA2	82	I/O
LA20/CS3P	80	I/O
LA21/CS1P	76	I/O
LA22/CS3S	74	I/O
LA23/CS1S	72	I/O
MDAK1 (PIIX) USBP0+(PIIX3)	144	O
MDRQ0 (PIIX) USBP1-(PIIX3)	143	I
MDRQ1 (PIIX) USBP1+(PIIX3)	142	I
MEMCS16#	70	I/O
MEMR#	88	I/O
MEMW#	90	I/O
MIDAK0 (PIIX) USBP0-(PIIX3)	145	O

Name	Pin #	Type
MIRQ0/IRQ0	147	O
MIRQ1 (PIIX) USBCLK (PIIX3)	146	I/O
NMI	135	O
OSC	136	I
PAR	186	O
PCICLK	132	I
PCIRST# (PIIX) PICRST#/ APICACK# (PIIX3)	128	O
PHLDA#	110	I
PHOLD#	109	O
PIRQA# (PIIX) PIRQA# (PIIX3)	149 149	I I/O
PIRQB#	150	I
PIRQC#	151	I
PIRQD#	152	I
PWROK	126	I
REFRESH#	31	I/O
RSTDRV	28	O
RTCALE	148	O
RTCCS#	138	O
SA0	69	I/O
SA1	68	I/O
SA2	67	I/O
SA3	66	I/O
SA4	63	I/O
SA5	61	I/O
SA6	59	I/O
SA7	57	I/O
SD0	17	I/O
SD1	16	I/O

Name	Pin #	Type
SD2	14	I/O
SD3	13	I/O
SD4	11	I/O
SD5	9	I/O
SD6	8	I/O
SD7	7	I/O
SD8	92	I/O
SD9	94	I/O
SD10	96	I/O
SD11	98	I/O
SD12	100	I/O
SD13	101	I/O
SD14	102	I/O
SD15	107	I/O
SDIR	118	O
SERR#	3	I
SMEMR#	19	O
SMEMW#	22	O
SMI#	123	O
SOE#	119	O
SPKR	117	O
STOP#	185	I/O
STPCLK#	124	O
SYSCLK	153	O
TC	62	O
TESTIN# (PIIX) TESTIN#/ APICREQ# (PIIX3)	134	I
TRDY#	181	I/O
VCC	27	V
VCC	53	V

Name	Pin #	Type
VCC	54	V
VCC	78	V
VCC	103	V
VCC	104	V
VCC (PIIX) VCC3 (PIIX3)	130	V
VCC	157	V
VCC	158	V
VCC	169	V
VCC	183	V
VCC	196	V
VCC	207	V
VCC	208	V
VSS	1	V
VSS	2	V
VSS	26	V
VSS	42	V

Name	Pin #	Type
VSS	51	V
VSS	52	V
VSS	65	V
VSS	79	V
VSS	105	V
VSS	106	V
VSS	131	V
VSS	133	V
VSS	155	V
VSS	156	V
VSS	170	V
VSS	182	V
VSS	195	V
XDIR	141	I
XOE#	140	O
ZEROWS#	15	I

4.2. PACKAGE DIMENSIONS

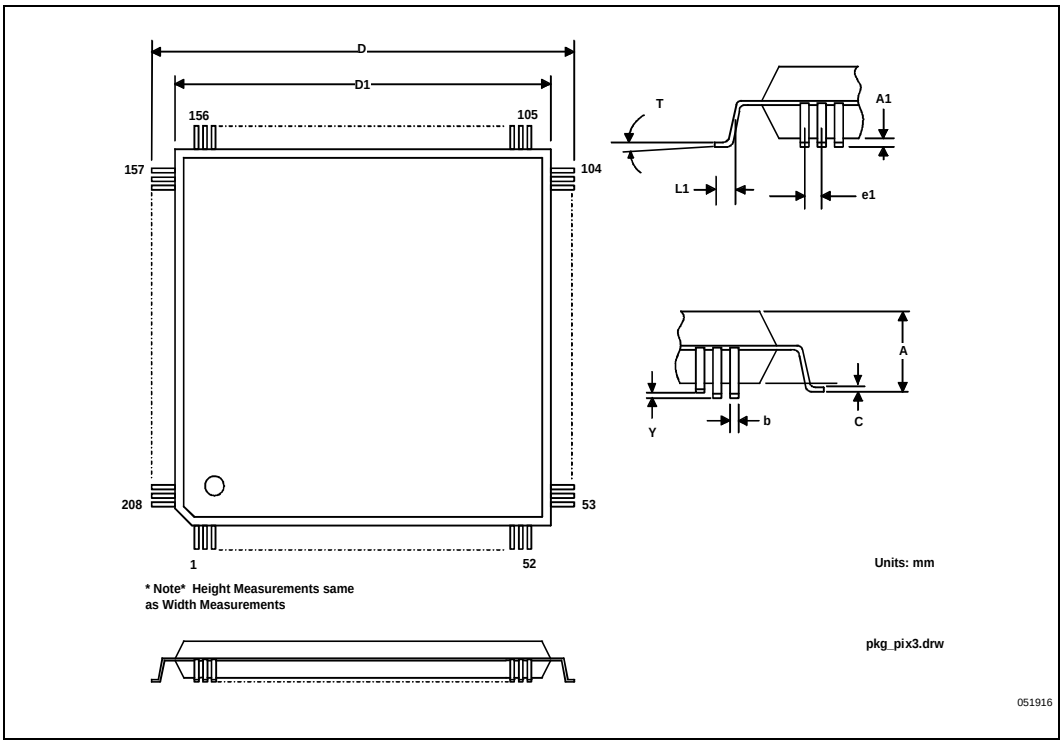


Figure 12. 208 Pin Quad Flat Pack (QFP) Dimensions

Table 16. 208 Pin Quad Flat Pack (QFP) Dimensions

Symbol	Description	Value (mm)
A	Seating Height	4.25 (max)
A1	Stand-off	0.05 (min); 0.40 (max)
b	Lead Width	0.2 ± 0.10
C	Lead Thickness	0.15 +0.1/-0.05
D	Package Length and Width, including pins	30.6 ± 0.4
D1	Package Length and Width, excluding pins	28 ± 0.2
e1	Linear Lead Pitch	0.5 ± 0.1
Y	Lead Coplanarity	0.08 (max)
L1	Foot Length	0.5 ± 0.2
T	Lead Angle	0^{o} - 10^{o}

5.0. TESTABILITY (PIIX/PIIX3)

5.1. Test Mode Description

The test modes are decoded from the IRQ inputs (IRQ 7, 6, 5) and qualified with the TESTIN# pin. Test mode selection is asynchronous. These signals need to remain in their respective state for the duration of the test modes. The test modes are defined as follows.

Table 17. Test Modes

Test Mode	IRQ7	IRQ6	IRQ5	TESTIN#
NAND Tree	0	x	x	0
NAND Tree	x	x	0	0
Tri-state All Outputs	1	1	1	0

5.2. NAND Tree Mode

Tri-states all outputs and bi-directional buffers except for XDIR and DACK1#. Every output buffer except for XDIR and DACK1# is configured as an input in NAND tree mode and included in the NAND chain. The first input of the NAND chain is MDRQ1, and the NAND chain is routed counter-clockwise around the chip (e.g., MDRQ1, MDRQ0, MDAK1#, . . .). DACK1# is an intermediate output, and XDIR is the final output. PCICLK and TESTIN# are the only input pins not included in the NAND chain. Note in the table above there are two possible ways to select NAND tree test mode.

To perform a NAND tree test, all pins included in the NAND tree should be driven to 1 except for the following pins, which use inverting Schmitt trigger inputs and should be driven to 0:

Table 18. Perform NAND Tree Test (Pins Driven To 0)

Pin #	Pin Name
4	IRQ1
5	IRQ8#
6	IOCHK#
10	IRQ9
15	ZEROWS#
32	IRQ7
33	IRQ6
34	IRQ5
56	IRQ4
58	IRQ3
73	IRQ10
75	IRQ11
77	IRQ12

Table 18. Perform NAND Tree Test (Pins Driven To 0)

Pin #	Pin Name
81	IRQ15
83	IRQ14
126	PWROK

Beginning with MDRQ and working counter-clockwise around the chip, each pin can be toggled and a resulting toggle observed on DACK1# or XDIR. The DACK1# output is provided so that the NAND tree test can be divided into two sections.

Table 19. NAND Tree

Pin #	Pin Name	Notes
134	TESTIN#	TESTIN# should be driven to 0 for the duration of the NAND tree test.
32	IRQ7	Test mode select signal.
33	IRQ6	Test mode select signal.
34	IRQ5	Test mode select signal.
142	MDRQ1	
143	MDRQ0	
144	MDK1	
145	MDK0	
146	MIRQ1	
147	MIRQ0	
148	RTCALE	
149	PIRQA#	
150	PIRQB#	
151	PIRQC#	
152	PIRQD#	
153	SYSCLK	
154	IDSEL	
159	AD31	
160	AD30	
161	AD29	
162	AD28	
163	AD27	
164	AD26	
165	AD25	

Table 19. NAND Tree

Pin #	Pin Name	Notes
166	AD24	
167	C/BE3#	
168	AD23	
171	AD22	
172	AD21	
173	AD20	
174	AD19	
175	AD18	
176	AD17	
177	AD16	
178	C/BE2#	
179	FRAME#	
180	IRDY#	
181	TRDY#	
184	DEVSEL#	
185	STOP#	
186	PAR	
187	C/BE1#	
188	AD15	
189	AD14	
190	AD13	
191	AD12	
192	AD11	
193	AD10	
194	AD9	
197	AD8	

Table 19. NAND Tree

Pin #	Pin Name	Notes
198	C/BE0#	
199	AD7	
200	AD6	
201	AD5	
202	AD4	
203	AD3	
204	AD2	
205	AD1	
206	AD0	
3	SERR#	
4	IRQ1	Inverted input signal
5	IRQ8#	Inverted input signal
6	IOCHK#	Inverted input signal
10	IRQ9	Inverted input signal
11	SD4	
12	DREQ2	
13	SD3	
14	SD2	
15	ZEROWS#	Inverted input signal
16	SD1	
17	SD0	
18	IOCHRDY	
19	SMEMR#	
20	AEN	
21	DACK3#	
22	SMEMW#	
23	IOR#	
24	IOW#	
25	DREQ3	
28	RSTDRV	
29	DACK1#	Intermediate NAND-tree output.
30	DREQ1	
31	REFRESH#	
32	IRQ7	Inverted input signal

Table 19. NAND Tree

Pin #	Pin Name	Notes
33	IRQ6	Inverted input signal
34	IRQ5	Inverted input signal
35	DD15	
36	DD14	
37	DD13	
38	DD12	
39	DD11	
40	DD10	
41	DD9	
43	DD8	
44	DD7	
45	DD6	
46	DD5	
47	DD4	
48	DD3	
49	DD2	
50	DD1	
55	DD0	
56	IRQ4	Inverted input signal
57	SA7	
58	IRQ3	Inverted input signal
59	SA6	
60	DACK2#	
61	SA5	
62	TC	
63	SA4	
64	BALE	
66	SA3	
67	SA2	
68	SA1	
69	SA0	
70	MEMCS16#	
71	IOCS16#	
72	LA23	
73	IRQ10	Inverted input signal

Table 19. NAND Tree

Pin #	Pin Name	Notes
74	LA22	
75	IRQ11	Inverted input signal
76	LA21	
77	IRQ12	Inverted input signal
80	LA20	
81	IRQ15	Inverted input signal
82	LA19	
83	IRQ14	Inverted input signal
84	LA18	
85	DACK0#	
86	LA17	
87	DREQ0	
88	MEMR#	
89	DACK5#	
90	MEMW#	
91	DREQ5	
92	SD8	
93	DACK6#	
94	SD9	
95	DREQ6	
96	SD10	
97	DACK7#	
98	SD11	
99	DREQ7	
100	SD12	
101	SD13	
102	SD14	
107	SD15	
108	DDRQ0	
109	PHOLD#	
110	PHLDA#	

Table 19. NAND Tree

Pin #	Pin Name	Notes
111	DDRQ1	
112	DIOW#	
113	DIOR#	
114	IORDY	
115	DDAK0#	
116	DDAK1#	
117	SPKR	
118	SDIR	
119	SOE#	
120	FERR#	
121	IGNNE#	
122	INTR	
123	SMI#	
124	STPCLK#	
125	EXTSMI#	
126	PWROK	Inverted input signal
127	CPURST#	
128	PCIRST#	
129	INIT	
132	PCICLK	Input only, not included in the NAND tree test mode.
135	NMI	
136	OSC	
137	BIOSCS#	
138	RTCCS#	
139	KBCS#	
140	XOE#	
141	XDIR	Final NAND tree output

Figure 13 is a schematic of the NAND tree circuitry.

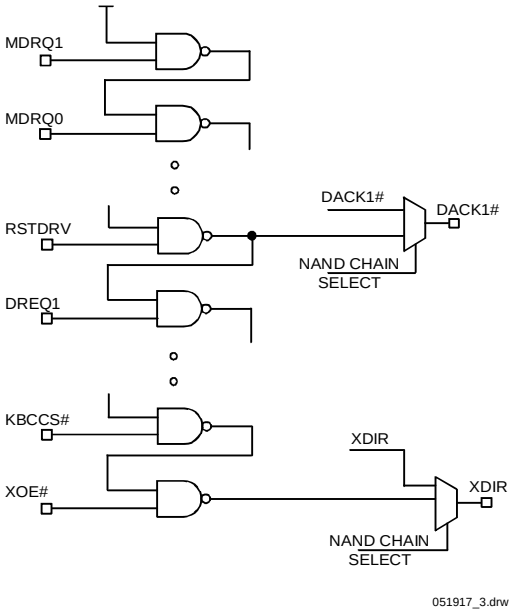


Figure 13. NAND Tree Circuitry

NAND Tree Timing Requirements

Allow 500 ns for the input signals to propagate to the NAND tree outputs (input-to-output propagation delay specification).

5.3. Tri-state Mode

The TESTIN# signal must be 0 and IRQ's 7, 6, and 5 must be 1 to enter the tri-state test mode. When in the tri-state test mode, all outputs and bi-directional pins are tri-stated, including the NAND tree outputs.