82371AB PCI-TO-ISA / IDE XCELERATOR (PIIX4)

- Supported Kits for both Pentium® and Pentium® II Microprocessors
  - 82430TX ISA Kit
  - 82440LX ISA/DP Kit
- Multifunction PCI to ISA Bridge
  - Supports PCI at 30 MHz and 33 MHz
  - Supports PCI Rev 2.1 Specification
  - Supports Full ISA or Extended I/O (EIO) Bus
  - Supports Full Positive Decode or Subtractive Decode of PCI
  - Supports ISA and EIO at 1/4 of PCI Frequency
- Supports both Mobile and Desktop Deep Green Environments
  - 3.3V Operation with 5V Tolerant Buffers
  - Ultra-low Power for Mobile Environments Support
  - Power-On Suspend, Suspend to RAM, Suspend to Disk, and Soft-OFF System States
  - All Registers Readable and Restorable for Proper Resume from 0.V Suspend
- Power Management Logic
  - Global and Local Device Management
  - Suspend and Resume Logic
  - Supports Thermal Alarm
  - Support for External Microcontroller
  - Full Support for Advanced Configuration and Power Interface (ACPI) Revision 1.0 Specification and OS Directed Power Management
- Integrated IDE Controller
  - Independent Timing of up to 4 Drives
  - PIO Mode 4 and Bus Master IDE Transfers up to 14 Mbytes/sec
  - Supports "Ultra DMA/33" Synchronous DMA Mode Transfers up to 33 Mbytes/sec
  - Integrated 16 x 32-bit Buffer for IDE PCI Burst Transfers
  - Supports Glue-less "Swap-Bay" Option with Full Electrical Isolation
- Enhanced DMA Controller
  - Two 82C37 DMA Controllers
  - Supports PCI DMA with 3 PC/PCI Channels and Distributed DMA Protocols (Simultaneously)
  - Fast Type-F DMA for Reduced PCI Bus Usage
- Interrupt Controller Based on Two 82C59
  - 15 Interrupt Support
  - Independently Programmable for Edge/Level Sensitivity
  - Supports Optional I/O APIC
  - Serial Interrupt Input
- Timers Based on 82C54
  - System Timer, Refresh Request, Speaker Tone Output
- USB
  - Two USB 1.0 Ports for Serial Transfers at 12 or 1.5 Mbit/sec
  - Supports Legacy Keyboard and Mouse Software with USB-based Keyboard and Mouse
  - Supports UHCI Design Guide
- SMBus
  - Host Interface Allows CPU to Communicate Via SMBus
  - Slave Interface Allows External SMBus Master to Control Resume Events
- Real-Time Clock
  - 256-byte Battery-Back CMOS SRAM
  - Includes Date Alarm
  - Two 8-byte Lockout Ranges
- Microsoft Win95* Compliant
- 324 mBGA Package

© INTEL CORPORATION 1997
April 1997 Order Number: 290562-001

The 82371AB PCI ISA IDE Xcelerator (PIIX4) is a multi-function PCI device implementing a PCI-to-ISA bridge function, a PCI IDE function, a Universal Serial Bus host/hub function, and an Enhanced Power Management function. As a PCI-to-ISA bridge, PIIX4 integrates many common I/O functions found in ISA-based PC systems—two 82C37 DMA Controllers, two 82C59 Interrupt Controllers, an 82C54 Timer/Counter, and a Real Time Clock. In addition to compatible transfers, each DMA channel supports Type F transfers. PIIX4 also contains full support for both PC/PCI and Distributed DMA protocols implementing PCI-based DMA. The Interrupt Controller has Edge or Level sensitive programmable inputs and fully supports the use of an external I/O Advanced Programmable Interrupt Controller (APIC) and Serial Interrupts. Chip select decoding is provided for BIOS, Real Time Clock, Keyboard Controller, second external microcontroller, as well as two Programmable Chip Selects. PIIX4 provides full Plug and Play compatibility. PIIX4 can be configured as a Subtractive Decode bridge or as a Positive Decode bridge. This allows the use of a subtractive decode PCI-to-PCI bridge such as the Intel 380FB PCIset which implements a PCI/ISA docking station environment.

PIIX4 supports two IDE connectors for up to four IDE devices providing an interface for IDE hard disks and CD ROMs. Up to four IDE devices can be supported in Bus Master mode. PIIX4 contains support for “Ultra DMA/33” synchronous DMA compatible devices.

PIIX4 contains a Universal Serial Bus (USB) Host Controller that is Universal Host Controller Interface (UHCI) compatible. The Host Controller’s root hub has two programmable USB ports.

PIIX4 supports Enhanced Power Management, including full Clock Control, Device Management for up to 14 devices, and Suspend and Resume logic with Power On Suspend, Suspend to RAM or Suspend to Disk. It fully supports Operating System Directed Power Management via the Advanced Configuration and Power Interface (ACPI) specification. PIIX4 integrates both a System Management Bus (SMBus) Host and Slave interface for serial communication with other devices.

Information in this document is provided in conjunction with Intel products. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document or by the sale of Intel products. Except as provided in Intel’s Terms and Conditions of Sale for such products, Intel assumes no liability whatsoever, and Intel disclaims any express or implied warranty, relating to sale and/or use of Intel products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right. Intel products are not intended for use in medical, life saving, or life sustaining applications.

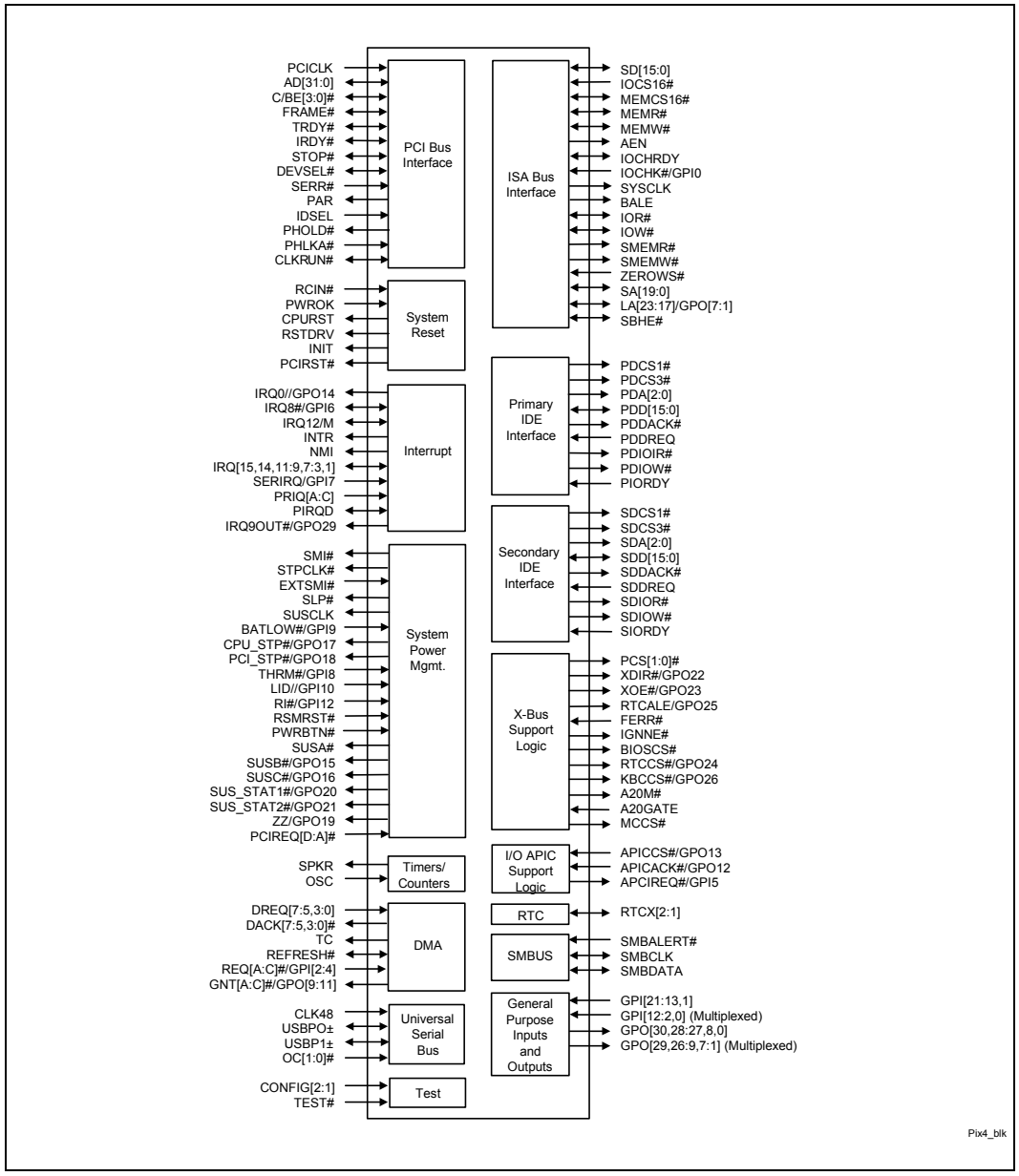
Intel retains the right to make changes to specifications and product descriptions at any time, without notice.

The 82371AB PIIX4 may contain design defects or errors known as errata. Current characterized errata are available on request.

Intel disclaims all liability, including liability for infringement of any proprietary rights, relating to use of information in this specification. Intel does not warrant or represent that such use will not infringe such rights.

I2C is a two-wire communication bus/protocol developed by Philips. SMBus is a subset of the I2C bus/protocol and was developed by Intel. Implementation of the I2C bus/protocol or the SMBus bus/protocol may require licenses from various entities, including Philips Electronics N.V. and North American Philips Corporation.

Third-party brands and names are the property of their respective owners.



Simplified Block Diagram

CONTENTS

PAGE

1.0. ARCHITECTURAL OVERVIEW 12

2.0. SIGNAL DESCRIPTION 15
2.1. PIIX4 Signals 16
2.1.1. PCI Bus Interface 16
2.1.2. ISA Bus Interface 18
2.1.3. X-Bus Interface 21
2.1.4. DMA Signals 23
2.1.5. Interrupt Controller/APIC Signals 24
2.1.6. CPU Interface Signals 26
2.1.7. Clocking Signals 28
2.1.8. IDE Signals 28
2.1.9. Universal Serial Bus Signals 33
2.1.10. Power Management Signals 33
2.1.11. General Purpose Input and Output Signals 35
2.1.12. Other System and Test Signals 39
2.1.13. Power and Ground Pins 39
2.2. Power Planes 40
2.3. Power Sequencing Requirements 41

3.0. REGISTER ADDRESS SPACE 42
3.1. PCI/ISA Bridge Configuration 42
3.1.1. PCI Configuration Registers (Function 0) 43
3.1.2. IO Space Registers 44
3.2. IDE Configuration 47
3.2.1. PCI Configuration Registers (Function 1) 47
3.2.2. IO Space Registers 48
3.3. Universal Serial Bus (USB) Configuration 48
3.3.1. PCI Configuration Registers (Function 2) 48
3.3.2. IO Space Registers 49
3.4. Power Management Configuration 50
3.4.1. IO Space Registers 51

4.0. PCI TO ISA/EIO BRIDGE REGISTER DESCRIPTIONS 53
4.1. PCI to ISA/EIO Bridge PCI Configuration Space Registers (PCI Function 0) 53
4.1.1. VID—Vendor Identification Register (Function 0) 53
4.1.2. DID—Device Identification Register (Function 0) 53
4.1.3. PCICMD—PCI Command Register (Function 0) 54
4.1.4. PCISTS—PCI Device Status Register (Function 0) 55
4.1.5. RID—Revision Identification Register (Function 0) 55

4.1.6. CLASSC—Class Code Register (Function 0) ... 56
4.1.7. HEDT—Header Type Register (Function 0) ... 56
4.1.8. IORT—ISA I/O Recovery Timer Register (Function 0) ... 56
4.1.9. XBCS—X-Bus Chip Select Register (Function 0) ... 57
4.1.10. PIRQRC[A:D]—PIRQX Route Control Registers (Function 0) ... 59
4.1.11. SERIRQC—Serial IRQ Control Register (Function 0) ... 59
4.1.12. TOM—Top of Memory Register (Function 0) ... 60
4.1.13. MSTAT—Miscellaneous Status Register (Function 0) ... 61
4.1.14. MBDMA[1:0]—Motherboard Device DMA Control Registers (Function 0) ... 61
4.1.15. APICBASE—APIC Base Address Relocation Register (Function 0) ... 62
4.1.16. DLC—Deterministic Latency Control Register (Function 0) ... 62
4.1.17. PDMACFG—PCI DMA Configuration Register (Function 0) ... 63
4.1.18. DDMABP—Distributed DMA Slave Base Pointer Registers (Function 0) ... 64
4.1.19. GENCFG—General Configuration Register (Function 0) ... 65
4.1.20. RTCCFG—Real Time Clock Configuration Register (Function 0) ... 67
4.2. PCI to ISA/EIO Bridge IO Space Registers (IO) ... 68
4.2.1. DMA Registers ... 68
4.2.1.1. DCOM—DMA Command Register (IO) ... 68
4.2.1.2. DCM—DMA Channel Mode Register (IO) ... 69
4.2.1.3. DR—DMA Request Register (IO) ... 70
4.2.1.4. WSMB—Write Single Mask Bit (IO) ... 70
4.2.1.5. RWAMB—Read/Write All Mask Bits (IO) ... 71
4.2.1.6. DS—DMA Status Register (IO) ... 71
4.2.1.7. DBADDR—DMA Base and Current Address Registers (IO) ... 72
4.2.1.8. DBCNT—DMA Base and Current Count Registers (IO) ... 72
4.2.1.9. DLPAGE—DMA Low Page Registers (IO) ... 73
4.2.1.10. DCBP—DMA Clear Byte Pointer Register (IO) ... 73
4.2.1.11. DMC—DMA Master Clear Register (IO) ... 73
4.2.1.12. DCLM—DMA Clear Mask Register (IO) ... 74
4.2.2. Interrupt Controller Registers ... 74
4.2.2.1. ICW1—Initialization Command Word 1 Register (IO) ... 74
4.2.2.2. ICW2—Initialization Command Word 2 Register (IO) ... 75
4.2.2.3. ICW3—Initialization Command Word 3 Register (IO) ... 75
4.2.2.4. ICW3—Initialization Command Word 3 Register (IO) ... 76
4.2.2.5. ICW4—Initialization Command Word 4 Register (IO) ... 76
4.2.2.6. OCW1—Operational Control Word 1 Register (IO) ... 77
4.2.2.7. OCW2—Operational Control Word 2 Register (IO) ... 77
4.2.2.8. OCW3— Operational Control Word 3 Register (IO) ... 78
4.2.2.9. ELCR1—Edge/Level Control Register (IO) ... 78
4.2.2.10. ELCR2—Edge/Level Control Register (IO) ... 79
4.2.3. Counter/Timer Registers ... 79
4.2.3.1. TCW—Timer Control Word Register (IO) ... 79
4.2.3.2. TMRSTS—Timer Status Registers (IO) ... 82

4.2.3.3. TMRCNT—Timer Count Registers (IO) 82
4.2.4. NMI Registers 83
4.2.4.1. NMISC—NMI Status and Control Register (IO) 83
4.2.4.2. NMIEN—NMI Enable Register (Shared with Real-Time Clock Index Register) (IO) 84
4.2.5. Real Time Clock Registers 84
4.2.5.1. RTCI—Real-Time Clock Index Register (Shared with NMI Enable Register) (IO) 84
4.2.5.2. RTCD—Real-Time Clock Data Register (IO) 85
4.2.5.3. RTCEI—Real-Time Clock Extended Index Register (IO) 85
4.2.5.4. RTCED—Real-Time Clock Extended Data Register (IO) 85
4.2.6. Advanced Power Management (APM) Registers 86
4.2.6.1. APMC—Advanced Power Management Control Port (IO) 86
4.2.6.2. APMS—Advanced Power Management Status Port (IO) 86
4.2.7. X-Bus, Coprocessor, and Reset Registers 86
4.2.7.1. RIRQ—Reset X-Bus IRQ12/M and IRQ1 Register (IO) 86
4.2.7.2. P92—Port 92 Register (IO) 87
4.2.7.3. CERR—Coprocessor Error Register (IO) 87
4.2.7.4. RC—Reset Control Register (IO) 88

5.0. IDE CONTROLLER REGISTER DESCRIPTIONS (PCI FUNCTION 1) 89
5.1. IDE Controller PCI Configuration Registers (PCI Function 1) 89
5.1.1. VID—Vendor Identification Register (Function 1) 89
5.1.2. DID—Device Identification Register (Function 1) 89
5.1.3. PCICMD—PCI Command Register (Function 1) 89
5.1.4. PCISTS—PCI Device Status Register (Function 1) 90
5.1.5. RID—Revision Identification Register (Function 1) 91
5.1.6. CLASSC—Class Code Register (Function 1) 91
5.1.7. MLT—Master Latency Timer Register (Function 1) 91
5.1.8. HEDT—Header Type Register (Function 1) 92
5.1.9. BMIBA—Bus Master Interface Base Address Register (Function 1) 92
5.1.10. IDETIM—IDE Timing Register (Function 1) 93
5.1.11. SIDETIM—Slave IDE Timing Register (Function 1) 95
5.1.12. UDMACTL—Ultra DMA/33 Control Register (Function 1) 96
5.1.13. UDMATIM—Ultra DMA/33 Timing Register (Function 1) 96
5.2. IDE Controller IO Space Registers 99
5.2.1. BMICX—Bus Master IDE Command Register (IO) 99
5.2.2. BMISX—Bus Master IDE Status Register (IO) 100
5.2.3. BMIDTPX—Bus Master IDE Descriptor Table Pointer Register (IO) 101

6.0. USB HOST CONTROLLER REGISTER DESCRIPTIONS (PCI FUNCTION 2) 102
6.1. USB Host Controller PCI Configuration Registers (PCI Function 2) 102
6.1.1. VID—Vendor Identification Register (Function 2) 102
6.1.2. DID—Device Identification Register (Function 2) 102
6.1.3. PCICMD—PCI Command Register (Function 2) 103
6.1.4. PCISTS—PCI Device Status Register (Function 2) 103

6.1.5. RID—Revision Identification Register (Function 2) 104
6.1.6. CLASSC—Class Code Register (Function 2) 104
6.1.7. MLT—Master Latency Timer Register (Function 2) 105
6.1.8. HEDT—Header Type Register (Function 2) 105
6.1.9. INTLN—Interrupt Line Register (Function 2) 105
6.1.10. INTPN—Interrupt Pin (Function 2) 106
6.1.11. SBRNUM—Serial Bus Release Number (Function 2) 106
6.1.12. LEGSUP—Legacy Support Register (Function 2) 107
6.1.13. USBBA—USB I/O Space Base Address Register (Function 2) 108
6.1.14. MISCSUP—Miscellaneous Support Register (Function 2) 109
6.2. USB Host Controller IO Space Registers 109
6.2.1. USBCMD—USB Command Register (IO) 109
6.2.2. USBSTS—USB Status Register (IO) 111
6.2.3. USBINTR—USB Interrupt Enable Register (IO) 112
6.2.4. FRNUM—Frame Number Register (IO) 113
6.2.5. FLBASEADD—Frame List Base Address Register (IO) 113
6.2.6. SOFMOD—Start Of Frame (SOF) Modify Register (IO) 114
6.2.7. PORTSC—Port Status and Control Register (IO) 115

7.0. POWER MANAGEMENT REGISTER DESCRIPTIONS 117
7.1. Power Management PCI Configuration Registers (PCI Function 3) 117
7.1.1. VID—Vendor Identification Register (Function 3) 117
7.1.2. DID—Device Identification Register (Function 3) 117
7.1.3. PCICMD—PCI Command Register (Function 3) 117
7.1.4. PCISTS—PCI Device Status Register (Function 3) 118
7.1.5. RID—Revision Identification Register (Function 3) 119
7.1.6. CLASSC—Class Code Register (Function 3) 119
7.1.7. HEDT—Header Type Register (Function 3) 119
7.1.8. INTLN—Interrupt Line Register (Function 3) 120
7.1.9. INTPN—Interrupt Pin (Function 3) 120
7.1.10. PMBA—Power Management Base Address (Function 3) 120
7.1.11. CNTA—Count A (Function 3) 121
7.1.12. CNTB—Count B (Function 3) 121
7.1.13. GPICTL—General Purpose Input Control (Function 3) 123
7.1.14. DEVRESD—Device Resource D (Function 3) 123
7.1.15. DEVACTA—Device Activity A (Function 3) 125
7.1.16. DEVACTB—Device Activity B (Function 3) 126
7.1.17. DEVRESA—Device Resource A (Function 3) 127
7.1.18. DEVRESB—Device Resource B (Function 3) 129
7.1.19. DEVRESC—Device Resource C (Function 3) 130
7.1.20. DEVRESE—Device Resource E (Function 3) 131
7.1.21. DEVRESF—Device Resource F (Function 3) 131
7.1.22. DEVRESG—Device Resource G (Function 3) 132

7.1.23. DEVRESH—Device Resource H (Function 3) ... 133
7.1.24. DEVRESI—Device Resource I (Function 3) ... 133
7.1.25. DEVRESJ—Device Resource J (Function 3) ... 133
7.1.26. PMREGMISC—Miscellaneous Power Management (Function 3) ... 134
7.1.27. SMBBA—SMBUS Base Address (Function 3) ... 135
7.1.28. SMBHSTCFG—SMBUS Host Configuration (Function 3) ... 135
7.1.29. SMBSLVC—SMBUS Slave Command (Function 3) ... 135
7.1.30. SMBSHDW1—SMBUS Slave Shadow Port 1 (Function 3) ... 136
7.1.31. SMBSHDW2—SMBUS Slave Shadow Port 2 (Function 3) ... 136
7.1.32. SMBREV—SMBUS Revision Identification (Function 3) ... 136
7.2. Power Management IO Space Registers ... 137
7.2.1. PMSTS—Power Management Status Register (IO) ... 137
7.2.2. PMEN—Power Management Resume Enable Register (IO) ... 138
7.2.3. PMCNTRL—Power Management Control Register (IO) ... 138
7.2.4. PMTMR—Power Management Timer Register (IO) ... 139
7.2.5. GPSTS—General Purpose Status Register (IO) ... 139
7.2.6. GPEN—General Purpose Enable Register (IO) ... 140
7.2.7. PCNTRL—Processor Control Register (IO) ... 141
7.2.8. PLVL2—Processor Level 2 Register (IO) ... 142
7.2.9. PLVL3— Processor Level 3 Register (IO) ... 142
7.2.10. GLBSTS—Global Status Register (IO) ... 143
7.2.11. DEVSTS—Device Status Register (IO) ... 144
7.2.12. GLBEN—Global Enable Register (IO) ... 144
7.2.13. GLBCTL—Global Control Register (IO) ... 145
7.2.14. DEVCTL—Device Control Register (IO) ... 146
7.2.15. GPIREG—General Purpose Input Register (IO) ... 147
7.2.16. GPOREG—General Purpose Output Register (IO) ... 148
7.3. SMBus IO Space Registers ... 148
7.3.1. SMBHSTSTS—SMBus Host Status Register (IO) ... 148
7.3.2. SMBSLVSTS—SMBus Slave Status Register (IO) ... 149
7.3.3. SMBHSTCNT—SMBus Host Control Register (IO) ... 150
7.3.4. SMBHSTCMD—SMBus Host Command Register (IO) ... 150
7.3.5. SMBHSTADD—SMBus Host Address Register (IO) ... 151
7.3.6. SMBHSTDAT0—SMBus Host Data 0 Register (IO) ... 151
7.3.7. SMBHSTDAT1—SMBus Host Data 1 Register (IO) ... 151
7.3.8. SMBBLKDAT—SMBus Block Data Register (IO) ... 152
7.3.9. SMBSLVCNT—SMBus Slave Control Register (IO) ... 152
7.3.10. SMBSHDWCMD—SMBus Shadow Command Register (IO) ... 153
7.3.11. SMBSLVEVT—SMBus Slave Event Register (IO) ... 153
7.3.12. SMBSLVDAT—SMBus Slave Data Register (IO) ... 153

8.0. PCI/ISA BRIDGE FUNCTIONAL DESCRIPTION ... 154
8.1. Memory and IO Address Map ... 154

8.1.1. I/O Accesses 154
8.1.2. Memory Address Map 154
8.1.3. BIOS Memory 155
8.2. PCI Interface 157
8.2.1. Transaction Termination 157
8.2.2. Parity Support 157
8.2.3. PCI Arbitration 157
8.3. ISA/EIO Interface 158
8.4. DMA Controller 158
8.4.1. DMA Transfer Modes 159
8.4.2. DMA Transfer Types 160
8.4.3. DMA Timings 161
8.4.4. DMA Buffer for Type F Transfers 161
8.4.5. DREQ and DACK# Latency Control 161
8.4.6. Channel Priority 162
8.4.7. Register Functionality 162
8.4.8. Address Compatibility Mode 162
8.4.9. Summary of DMA Transfer Sizes 163
8.4.9.1. Address Shifting When Programmed for 16-Bit I/O Count by Words 163
8.4.10. Autoinitialize 163
8.4.11. Software Commands 164
8.4.12. ISA Refresh Cycles 164
8.5. PCI DMA 165
8.5.1. PC/PCI DMA 165
8.5.2. Distributed DMA 168
8.6. Interrupt Controller 171
8.6.1. Programming the Interrupt Controller 172
8.6.2. End-of-Interrupt Operation 173
8.6.3. Modes of Operation 173
8.6.4. Cascade Mode 175
8.6.5. Edge and Level Triggered Mode 175
8.6.6. Interrupt Masks 176
8.6.7. Reading the Interrupt Controller Status 176
8.6.8. Interrupt Steering 177
8.7. Serial Interrupts 178
8.7.1. Protocol 178
8.8. Timer/Counters 180
8.8.1. Programming the Interval Timer 180
8.9. Real Time Clock 183
8.9.1. RTC Registers and RAM 183
8.9.1.1. Control Register A 185
8.9.1.2. Control Register B 186
8.9.1.3. Control Register C 187

8.9.1.4. Control Register D 187
8.9.2. RTC Update Cycle 188
8.9.3. RTC Interrupts 188
8.9.4. Lockable RAM Ranges 188
8.9.5. RTC External Connections 188
8.10. X-Bus Support 188
8.11. Reset Support 189
8.12. Stand-Alone I/O APIC Support 190

9.0. IDE CONTROLLER FUNCTIONAL DESCRIPTION 191
9.1. IDE Signal Configuration 191
9.2. ATA Register Block Decode 192
9.3. PIO IDE Transactions 193
9.4. Bus Master Function 195
9.5. “Ultra DMA/33” Synchronous DMA Operation 197

10.0. USB HOST CONTROLLER FUNCTIONAL DESCRIPTION 199

11.0. POWER MANAGEMENT FUNCTIONAL DESCRIPTION 201
11.1. Power Management Overview 201
11.2. Clock Control 202
11.2.1. Host Clock Control Mechanisms 202
11.2.2. Stop Clock and Deep Sleep State Example Sequence 208
11.2.3. PCI Clock Control 210
11.3. Peripheral Device Management 210
11.3.1. Device Idle Timer 211
11.3.2. Device Trap 212
11.3.3. Peripheral Device Management Sequence 212
11.3.4. Device Location on PCI Bus or ISA Bus 212
11.3.5. Device Specific Details 215
11.3.5.1. Device 0: IDE Primary Drive 0 215
11.3.5.2. Device 1: IDE Primary Drive 1 216
11.3.5.3. Device 2: IDE Secondary Drive 0 216
11.3.5.4. Device 3: IDE Secondary Drive 1 217
11.3.5.5. Device 4: Audio 218
11.3.5.6. Device 5: Floppy Disk Drive 219
11.3.5.7. Device 6: Serial Port A 220
11.3.5.8. Device 7: Serial Port B 221
11.3.5.9. Device 8: LPT (Parallel Port) 222
11.3.5.10. Device 9: Generic I/O Device 0 223
11.3.5.11. Device 10: Generic I/O Device 1 224
11.3.5.12. Device 11: User Interface (Keyboard, Mouse, Video) 225
11.3.5.13. Device 12: Cardbus Slot (or Generic I/O and MEM Device) 226
11.3.5.14. Device 13: Cardbus Slot (or Generic I/O and MEM Device) 227

11.4. Suspend/Resume and Power Plane Control 228
11.4.1. System Suspend 228
11.4.2. System Resume 230
11.4.3. System Suspend and Resume Control Signaling 232
11.4.3.1. Power Supply Timings 232
11.4.3.2. Power Level Active Status Signal Timings 233
11.4.3.3. Power Management Signal Timings (Powered From Suspend Power Well) 234
11.4.3.4. PCI Clock Stop and Start Timing Relationships 235
11.4.3.5. Power Management Signal Timings (Powered From PIIX4 Main Core Well) 236
11.4.3.6. Power Management Signal Timings (Powered From PIIX4 Main Core Well) 238
11.4.3.7. Mechanical Off to On Condition Timings 240
11.4.3.8. On State to Power On Suspend State Timing 242
11.4.3.9. Power On Suspend to On Timing (With a Full System Reset) 244
11.4.3.10. System Transition From Power On Suspend to On (With Only Processor Reset) 246
11.4.3.11. Power On Suspend to On Timing (With No Resets) 248
11.4.3.12. On State to Suspend to RAM State Timing 250
11.4.3.13. Suspend-To-RAM to On Timing (With Full System Reset) 252
11.4.3.14. On State to Suspend to Disk/Soft Off State Timings 254
11.4.3.15. Suspend-To-Disk to On (With Full System Reset) 256
11.4.4. Shadow Registers 258
11.5. System Management 262
11.5.1. SMI Operation 262
11.5.2. SMI# Generation Events 263
11.5.3. Global Standby Timer Operation 265
11.5.4. SMBus Functional Description 266
11.5.4.1. SMBus Host Interface 266
11.5.4.2. SMBus Slave Interface 267
11.6. ACPI Support 268
11.6.1. SCI Generation 268
11.6.2. Power Management Timer 268
11.6.3. Global Lock 269

12.0. PINOUT INFORMATION 270

13.0. PIIX4 PACKAGE INFORMATION 274

14.0. TESTABILITY 277
14.1. Test Mode Description 277
14.2. Tri-state Mode 278
14.3. NAND Tree Mode 278

1.0. ARCHITECTURAL OVERVIEW

PIIX4 is a multi-function PCI device that integrates many system-level functions. Figure 1 shows an example system block diagram using PIIX4.

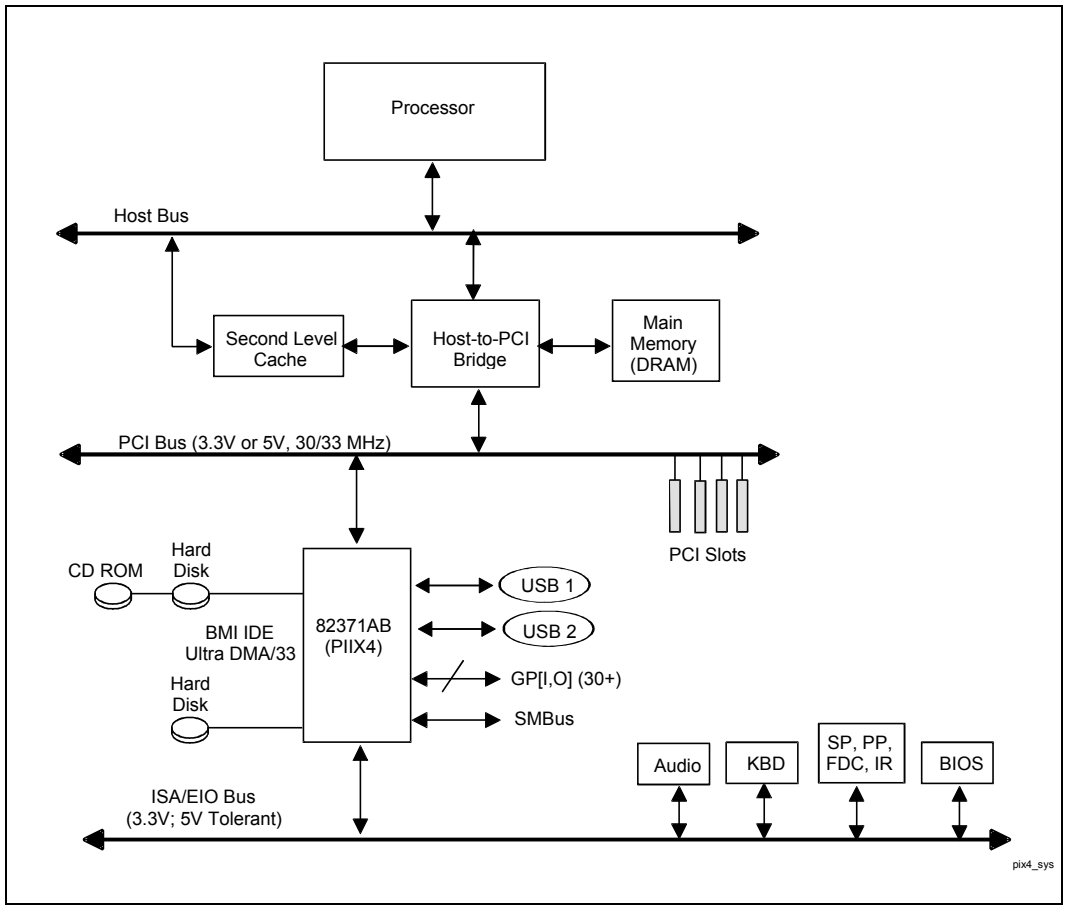


Figure 1. PIIX4 System Block Diagram

PCI to ISA/EIO Bridge

PIIX4 is compatible with the PCI Rev 2.1 specification, as well as the IEEE 996 specification for the ISA (AT) bus. On PCI, PIIX4 operates as a master for various internal modules, such as the USB controller, DMA controller, IDE bus master controller, distributed DMA masters, and on behalf of ISA masters. PIIX4 operates as a slave for its internal registers or for cycles that are passed to the ISA or EIO buses. All internal registers are positively decoded.

PIIX4 can be configured for a full ISA bus or a subset of the ISA bus called the Extended IO (EIO) bus. The use of the EIO bus allows unused signals to be configured as general purpose inputs and outputs. PIIX4 can

directly drive up to five ISA slots without external data or address buffering. It also provides byte-swap logic, I/O recovery support, wait-state generation, and SYSCLK generation. X-Bus chip selects are provided for Keyboard Controller, BIOS, Real Time Clock, a second microcontroller, as well as two programmable chip selects.

PIIX4 can be configured as either a subtractive decode PCI to ISA bridge or as a positive decode bridge. This gives a system designer the option of placing another subtractive decode bridge in the system (e.g., an Intel 380FB Dock Set).

IDE Interface (Bus Master capability and synchronous DMA Mode)

The fast IDE interface supports up to four IDE devices providing an interface for IDE hard disks and CD ROMs. Each IDE device can have independent timings. The IDE interface supports PIO IDE transfers up to 14 Mbytes/sec and Bus Master IDE transfers up to 33 Mbytes/sec. It does not consume any ISA DMA resources. The IDE interface integrates 16x32-bit buffers for optimal transfers.

PIIX4's IDE system contains two independent IDE signal channels. They can be electrically isolated independently, allowing for the implementation of a "glueless" Swap Bay. They can be configured to the standard primary and secondary channels (four devices) or primary drive 0 and primary drive 1 channels (two devices). This allows flexibility in system design and device power management.

Compatibility Modules (DMA Controller, Timer/Counters, Interrupt Controller)

The DMA controller incorporates the logic of two 82C37 DMA controllers, with seven independently programmable channels. Channels [0:3] are hardwired to 8-bit, count-by-byte transfers, and channels [5:7] are hardwired to 16-bit, count-by-word transfers. Any two of the seven DMA channels can be programmed to support fast Type-F transfers. The DMA controller also generates the ISA refresh cycles.

The DMA controller supports two separate methods for handling legacy DMA via the PCI bus. The PC/PCI protocol allows PCI-based peripherals to initiate DMA cycles by encoding requests and grants via three PC/PCI REQ#/GNT# pairs. The second method, Distributed DMA, allows reads and writes to 82C37 registers to be distributed to other PCI devices. The two methods can be enabled concurrently. The serial interrupt scheme typically associated with Distributed DMA is also supported.

The timer/counter block contains three counters that are equivalent in function to those found in one 82C54 programmable interval timer. These three counters are combined to provide the system timer function, refresh request, and speaker tone. The 14.31818-MHz oscillator input provides the clock source for these three counters.

PIIX4 provides an ISA-Compatible interrupt controller that incorporates the functionality of two 82C59 interrupt controllers. The two interrupt controllers are cascaded so that 14 external and two internal interrupts are possible. In addition, PIIX4 supports a serial interrupt scheme. PIIX4 provides full support for the use of an external IO APIC.

All of the registers in these modules can be read and restored. This is required to save and restore system state after power has been removed and restored to the circuit.

Enhanced Universal Serial Bus (USB) Controller

The PIIX4 USB controller provides enhanced support for the Universal Host Controller Interface (UHCI). This includes support that allows legacy software to use a USB-based keyboard and mouse.

RTC

PIIX4 contains a Motorola* MC146818A-compatible real-time clock with 256 bytes of battery-backed RAM. The real-time clock performs two key functions: keeping track of the time of day and storing system data, even when the system is powered down. The RTC operates on a 32.768-kHz crystal and a separate 3V lithium battery that provides up to 7 years of protection.

The RTC also supports two lockable memory ranges. By setting bits in the configuration space, two 8-byte ranges can be locked to read and write accesses. This prevents unauthorized reading of passwords or other system security information.

The RTC also supports a date alarm, that allows for scheduling a wake up event up to 30 days in advance, rather than just 24 hours in advance.

GPIO and Chip Selects

Various general purpose inputs and outputs are provided for custom system design. The number of inputs and outputs varies depending on PIIX4 configuration. Two programmable chip selects are provided which allows the designer to place devices on the X-Bus without the need for external decode logic.

Pentium® and Pentium® II Processor Interface

The PIIX4 CPU interface allows connection to all Pentium and Pentium II processors. The Sleep mode for the Pentium II processors is also supported.

Enhanced Power Management

PIIX4's power management functions include enhanced clock control, local and global monitoring support for 14 individual devices, and various low-power (suspend) states, such as Power-On Suspend, Suspend-to-DRAM, and Suspend-to-Disk. A hardware-based thermal management circuit permits software-independent entrance to low-power states. PIIX4 has dedicated pins to monitor various external events (e.g., interfaces to a notebook lid, suspend/resume button, battery low indicators, etc.). PIIX4 contains full support for the Advanced Configuration and Power Interface (ACPI) Specification.

System Management Bus (SMBus)

PIIX4 contains an SMBus Host interface that allows the CPU to communicate with SMBus slaves and an SMBus Slave interface that allows external masters to activate power management events.

Configurability

PIIX4 provides a wide range of system configuration options. This includes full 16-bit I/O decode on internal modules, dynamic disable on all the internal modules, various peripheral decode options, and many options on system configuration.

2.0. SIGNAL DESCRIPTION

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

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 exclusively. This is done to avoid confusion when working with a mixture of "active low" and "active high" signal. 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.

Certain signals have different functions, depending on the configuration programmed in the PCI configuration space. The signal whose function is being described is in **bold** font. Some of the signals are multiplexed with General Purpose Inputs and Outputs. The default configuration and control bits for each are described in Table 1 and Table 2.

Each output signal description includes the value of the signal **During Reset**, **After Reset**, and **During POS**. **During Reset** refers to when the PCIRST# signal is asserted. **After Reset** is immediately after negation of PCIRST# and the signal may change value anytime thereafter. The term **High-Z** means tri-stated. The term **Undefined** means the signal could be high, low, tri-stated, or in some in-between level. Some of the power management signals are reset with the RSMRST# input signal. The functionality of these signals during RSMRST# assertion is described in the *Suspend/Resume and Power Plane Control* section.

The I/O buffer types are shown below:

Buffer Type	Description
I	input only signal
O	totem pole output
I/O	bi-direction, tri-state input/output pin
s/t/s	sustained tri-state
OD	open drain
I/OD	input/open drain output is a standard input buffer with an open drain output
V	This is not a standard signal. It is a power supply pin.
3.3V/2.5V	Indicates the buffer is 3.3V or 2.5V only, depending on the voltage (3.3V or 2.5V) connected to $V_{CC}X$ pins.
3.3V/5V	Indicates that the output is 3.3V and input is 3.3V receiver with 5V tolerance.
5V	Indicates 3.3V receiver with 5V tolerance.

All 3V output signals can drive 5V TTL inputs. Most of the 3V input signals are 5V tolerant. The 3V input signals which are powered via the RTC or Suspend power planes should not exceed their power supply voltage (see *Power Planes* chapter for additional information). The open drain (OD) CPU interface signals should be pulled up to the CPU interface signal voltage.

2.1. PIIX4 Signals

2.1.1. PCI BUS INTERFACE

Name	Type	Description
AD[31:0]	I/O	**PCI ADDRESS/DATA.** AD[31:0] is a multiplexed address and data bus. During the first clock of a transaction, AD[31:0] contain a physical byte address (32 bits). During subsequent clocks, AD[31:0] contain data. A PIIX4 Bus transaction consists of an address phase followed by one or more data phases. Little-endian byte ordering is used. AD[7:0] define the least significant byte (LSB) and AD[31:24] the most significant byte (MSB). When PIIX4 is a Target, AD[31:0] are inputs during the address phase of a transaction. During the following data phase(s), PIIX4 may be asked to supply data on AD[31:0] for a PCI read, or accept data for a PCI write. As an Initiator, PIIX4 drives a valid address on AD[31:2] and 0 on AD[1:0] during the address phase, and drives write or latches read data on AD[31:0] during the data phase. **During Reset:** High-Z **After Reset:** High-Z **During POS:** High-Z
C/BE#[3:0]	I/O	**BUS COMMAND AND BYTE ENABLES.** The command and byte enable signals are multiplexed on the same PCI pins. During the address phase of a transaction, C/BE[3:0]# define the bus command. During the data phase C/BE[3:0]# are used as Byte Enables. The Byte Enables determine which byte lanes carry meaningful data. C/BE0# applies to byte 0, C/BE1# to byte 1, etc. PIIX4 drives C/BE[3:0]# as an Initiator and monitors C/BE[3:0]# as a Target. **During Reset:** High-Z **After Reset:** High-Z **During POS:** High-Z
CLKRUN#	I/O	**CLOCK RUN#.** This signal is used to communicate to PCI peripherals that the PCI clock will be stopped. Peripherals can assert CLKRUN# to request that the PCI clock be restarted or to keep it from stopping. This function follows the protocol described in the PCI Mobile Design Guide, Revision 1.0. **During Reset:** Low **After Reset:** Low **During POS:** High
DEVSEL#	I/O	**DEVICE SELECT.** PIIX4 asserts DEVSEL# to claim a PCI transaction through positive decoding or subtractive decoding (if enabled). As an output, PIIX4 asserts DEVSEL# when it samples IDSEL active in configuration cycles to PIIX4 configuration registers. PIIX4 also asserts DEVSEL# when an internal PIIX4 address is decoded or when PIIX4 subtractively or positively decodes a cycle for the ISA/EIO bus or IDE device. As an input, DEVSEL# indicates the response to a PIIX4 initiated transaction and is also sampled when deciding whether to subtractively decode the cycle. DEVSEL# is tri-stated from the leading edge of PCIRST#. DEVSEL# remains tri-stated until driven by PIIX4 as a target. **During Reset:** High-Z **After Reset:** High-Z **During POS:** High-Z
FRAME#	I/O	**CYCLE FRAME.** FRAME# is driven by the current Initiator to indicate the beginning and duration of an access. While FRAME# is asserted data transfers continue. When FRAME# is negated the transaction is in the final data phase. FRAME# is an input to PIIX4 when it is the Target. FRAME# is an output when PIIX4 is the initiator. FRAME# remains tri-stated until driven by PIIX4 as an Initiator. **During Reset:** High-Z **After Reset:** High-Z **During POS:** High-Z

Name	Type	Description
IDSEL	I	**INITIALIZATION DEVICE SELECT.** IDSEL is used as a chip select during PCI configuration read and write cycles. PIIX4 samples IDSEL during the address phase of a transaction. If IDSEL is sampled active, and the bus command is a configuration read or write, PIIX4 responds by asserting DEVSEL# on the next cycle.
IRDY#	I/O	**INITIATOR READY.** IRDY# indicates PIIX4's ability, as an Initiator, to complete the current data phase of the transaction. It is used in conjunction with TRDY#. A data phase is completed on any clock both IRDY# and TRDY# are sampled asserted. During a write, IRDY# indicates PIIX4 has valid data present on AD[31:0]. During a read, it indicates PIIX4 is prepared to latch data. IRDY# is an input to PIIX4 when PIIX4 is the Target and an output when PIIX4 is an Initiator. IRDY# remains tri-stated until driven by PIIX4 as a master. **During Reset:** High-Z **After Reset:** High-Z **During POS:** High-Z
PAR	O	**CALCULATED PARITY SIGNAL.** PAR is "even" parity and is calculated on 36 bits; AD[31:0] plus C/BE[3:0]#. "Even" parity means that the number of "1"s within the 36 bits plus PAR are counted and the sum is always even. PAR is always calculated on 36 bits regardless of the valid byte enables. PAR is generated for address and data phases and is only guaranteed to be valid one PCI clock after the corresponding address or data phase. PAR is driven and tri-stated identically to the AD[31:0] lines except that PAR is delayed by exactly one PCI clock. PAR is an output during the address phase (delayed one clock) for all PIIX4 initiated transactions. It is also an output during the data phase (delayed one clock) when PIIX4 is the Initiator of a PCI write transaction, and when it is the Target of a read transaction. **During Reset:** High-Z **After Reset:** High-Z **During POS:** High-Z
PCIRST#	O	**PCI RESET.** PIIX4 asserts PCIRST# to reset devices that reside on the PCI bus. PIIX4 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 for a minimum of 1 ms when initiated through the RC register. PCIRST# is driven asynchronously relative to PCICLK. **During Reset:** Low **After Reset:** High **During POS:** High
PHOLD#	O	**PCI HOLD.** An active low assertion indicates that PIIX4 desires use of the PCI Bus. Once the PCI arbiter has asserted PHLDA# to PIIX4, it may not negate it until PHOLD# is negated by PIIX4. PIIX4 implements the passive release mechanism by toggling PHOLD# inactive for one PCICLK. **During Reset:** High-Z **After Reset:** High **During POS:** High
PHLDA#	I	**PCI HOLD ACKNOWLEDGE.** An active low assertion indicates that PIIX4 has been granted use of the PCI Bus. Once PHLDA# is asserted, it cannot be negated unless PHOLD# is negated first.
SERR#	I/O	**SYSTEM ERROR.** SERR# can be pulsed active by any PCI device that detects a system error condition. Upon sampling SERR# active, PIIX4 can be programmed to generate a non-maskable interrupt (NMI) to the CPU. **During Reset:** High-Z **After Reset:** High-Z **During POS:** High-Z

Name	Type	Description
STOP#	I/O	**STOP.** STOP# indicates that PIIX4, as a Target, is requesting an initiator to stop the current transaction. As an Initiator, STOP# causes PIIX4 to stop the current transaction. STOP# is an output when PIIX4 is a Target and an input when PIIX4 is an Initiator. STOP# is tri-stated from the leading edge of PCIRST#. STOP# remains tri-stated until driven by PIIX4 as a slave. **During Reset:** High-Z **After Reset:** High-Z **During POS:** High-Z
TRDY#	I/O	**TARGET READY.** TRDY# indicates PIIX4's ability to complete the current data phase of the transaction. TRDY# is used in conjunction with IRDY#. A data phase is completed when both TRDY# and IRDY# are sampled asserted. During a read, TRDY# indicates that PIIX4, as a Target, has place valid data on AD[31:0]. During a write, it indicates PIIX4, as a Target is prepared to latch data. TRDY# is an input to PIIX4 when PIIX4 is the Initiator and an output when PIIX4 is a Target. TRDY# is tri-stated from the leading edge of PCIRST#. TRDY# remains tri-stated until driven by PIIX4 as a slave. **During Reset:** High-Z **After Reset:** High-Z **During POS:** High-Z

NOTES:
All of the signals in the host interface are described in the Pentium Processor data sheet. The preceding table highlights PIIX4 specific uses of these signals.

2.1.2. ISA BUS INTERFACE

Name	Type	Description
AEN	O	**ADDRESS ENABLE.** AEN is asserted during DMA cycles to prevent I/O slaves from misinterpreting DMA cycles as valid I/O cycles. When negated, AEN indicates that an I/O slave may respond to address and I/O commands. When asserted, AEN informs I/O resources on the ISA bus that a DMA transfer is occurring. This signal is also driven high during PIIX4 initiated refresh cycles. **During Reset:** High-Z **After Reset:** Low **During POS:** Low
BALE	O	**BUS ADDRESS LATCH ENABLE.** BALE is asserted by PIIX4 to indicate that the address (SA[19:0], LA[23:17]) and SBHE# signal lines are valid. The LA[23:17] address lines are latched on the trailing edge of BALE. BALE remains asserted throughout DMA and ISA master cycles. **During Reset:** High-Z **After Reset:** Low **During POS:** Low
IOCHK#/ GPI0	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. A NMI will be generated to the CPU if the NMI generation is enabled. If the EIO bus is used, this signal becomes a general purpose input.

Name	Type	Description
IOCHRDY	I/O	**I/O CHANNEL READY.** Resources on the ISA Bus negate IOCHRDY to indicate that wait states are required to complete the cycle. This signal is normally high. IOCHRDY is an input when PIIX4 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 PIIX4 register. As a PIIX4 output, IOCHRDY is driven inactive (low) from the falling edge of the ISA commands. After data is available for an ISA master read or PIIX4 latches the data for a write cycle, IOCHRDY is asserted for 70 ns. After 70 ns, PIIX4 floats IOCHRDY. The 70 ns includes both the drive time and the time it takes PIIX4 to float IOCHRDY. PIIX4 does not drive this signal when an ISA Bus master is accessing an ISA Bus slave. **During Reset:** High-Z **After Reset:** High-Z **During POS:** High-Z
IOCS16#	I	**16-BIT I/O CHIP SELECT.** This signal is driven by I/O devices on the ISA Bus to indicate support for 16-bit I/O bus cycles.
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]). The I/O slave device must hold the data valid until after IOR# is negated. IOR# is an output when PIIX4 owns the ISA Bus. IOR# is an input when an external ISA master owns the ISA Bus. **During Reset:** High-Z **After Reset:** High **During POS:** High
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]). IOW# is an output when PIIX4 owns the ISA Bus. IOW# is an input when an external ISA master owns the ISA Bus. **During Reset:** High-Z **After Reset:** High **During POS:** High
LA[23:17]/ GPO[7:1]	I/O	**ISA LA[23:17].** LA[23:17] address lines allow accesses to physical memory on the ISA Bus up to 16 Mbytes. LA[23:17] are outputs when PIIX4 owns the ISA Bus. The LA[23:17] lines become inputs whenever an ISA master owns the ISA Bus. If the EIO bus is used, these signals become a general purpose output. **During Reset:** High-Z **After Reset:** Undefined **During POS:** Last LA/GPO
MEMCS16#	I/O	**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. PIIX4 ignores MEMCS16# during I/O access cycles and refresh cycles. MEMCS16# is an input when PIIX4 owns the ISA Bus. PIIX4 drives this signal low during ISA master to PCI memory cycles. **During Reset:** High-Z **After Reset:** High-Z **During POS:** High-Z
MEMR#	I/O	**MEMORY READ.** MEMR# is the command to a memory slave that it may drive data onto the ISA data bus. MEMR# is an output when PIIX4 is a master on the ISA Bus. MEMR# is an input when an ISA master, other than PIIX4, owns the ISA Bus. This signal is also driven by PIIX4 during refresh cycles. For DMA cycles, PIIX4, as a master, asserts MEMR#. **During Reset:** High-Z **After Reset:** High **During POS:** High

Name	Type	Description
MEMW#	I/O	**MEMORY WRITE.** MEMW# is the command to a memory slave that it may latch data from the ISA data bus. MEMW# is an output when PIIX4 owns the ISA Bus. MEMW# is an input when an ISA master, other than PIIX4, owns the ISA Bus. For DMA cycles, PIIX4, as a master, asserts MEMW#. **During Reset:** High-Z **After Reset:** High **During POS:** High
REFRESH#	I/O	**REFRESH.** As an output, REFRESH# is used by PIIX4 to indicate 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 PIIX4 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. **During Reset:** High-Z **After Reset:** High **During POS:** High
RSTDRV	O	**RESET DRIVE.** PIIX4 asserts RSTDRV to reset devices that reside on the ISA/EIO Bus. PIIX4 asserts this signal during a hard reset and during power-up. RSTDRV is asserted during power-up and negated after PWROK is driven active. RSTDRV is also driven active for a minimum of 1 ms if a hard reset has been programmed in the RC register. **During Reset:** High **After Reset:** Low **During POS:** Low
SA[19:0]	I/O	**SYSTEM ADDRESS[19:0].** These bi-directional address lines define the selection with the granularity of 1 byte within the 1-Megabyte 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, and SA[19:16] are undefined. SA[19:0] are outputs when PIIX4 owns the ISA Bus. SA[19:0] are inputs when an external ISA Master owns the ISA Bus. **During Reset:** High-Z **After Reset:** Undefined **During POS:** Last SA
SBHE#	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. SBHE# is an output when PIIX4 owns the ISA Bus. SBHE# is an input when an external ISA master owns the ISA Bus. **During Reset:** High-Z **After Reset:** Undefined **During POS:** High
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. **During Reset:** High-Z **After Reset:** Undefined **During POS:** High-Z
SMEMR#	O	**STANDARD MEMORY READ.** PIIX4 asserts SMEMR# to request an ISA memory slave to drive data onto the data lines. If the access is below the 1-Mbyte range (00000000h–000FFFFFh) during DMA compatible, PIIX4 master, or ISA master cycles, PIIX4 asserts SMEMR#. SMEMR# is a delayed version of MEMR#. **During Reset:** High-Z **After Reset:** High **During POS:** High

Name	Type	Description
SMEMW#	O	**STANDARD MEMORY WRITE.** PIIX4 asserts SMEMW# to request an ISA memory slave to accept data from the data lines. If the access is below the 1-Mbyte range (00000000h–000FFFFFh) during DMA compatible, PIIX4 master, or ISA master cycles, PIIX4 asserts SMEMW#. SMEMW# is a delayed version of MEMW#. **During Reset:** High-Z **After Reset:** High **During POS:** High
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. If IOCHRDY is negated and ZEROWS# is asserted during the same clock, then ZEROWS# is ignored and wait states are added as a function of IOCHRDY.

2.1.3. X-BUS INTERFACE

Name	Type	Description
A20GATE	I	**ADDRESS 20 GATE.** This input from the keyboard controller is logically combined with bit 1 (FAST_A20) of the Port 92 Register, which is then output via the A20M# signal.
BIOSCS#	O	**BIOS CHIP SELECT.** This chip select is driven active during read or write accesses to enabled BIOS memory ranges. BIOSCS# is driven combinatorially from the ISA addresses SA[16:0] and LA[23:17], except during DMA cycles. During DMA cycles, BIOSCS# is not generated. **During Reset:** High **After Reset:** High **During POS:** High
KBCCS#/ GPO26	O	**KEYBOARD CONTROLLER CHIP SELECT.** KBCCS# is asserted during I/O read or write accesses to KBC locations 60h and 64h. It is driven combinatorially from the ISA addresses SA[19:0] and LA[23:17]. If the keyboard controller does not require a separate chip select, this signal can be programmed to a general purpose output. **During Reset:** High **After Reset:** High **During POS:** High/GPO
MCCS#	O	**MICROCONTROLLER CHIP SELECT.** MCCS# is asserted during I/O read or write accesses to IO locations 62h and 66h. It is driven combinatorially from the ISA addresses SA[19:0] and LA[23:17]. **During Reset:** High **After Reset:** High **During POS:** High
PCS0# **PCS1#**	O	**PROGRAMMABLE CHIP SELECTS.** These active low chip selects are asserted for ISA I/O cycles which are generated by PCI masters and which hit the programmable I/O ranges defined in the Power Management section. The X-Bus buffer signals (XOE# and XDIR#) are enabled while the chip select is active. (i.e., it is assumed that the peripheral which is selected via this pin resides on the X-Bus.) **During Reset:** High **After Reset:** High **During POS:** High
RCIN#	I	**RESET CPU.** This signal from the keyboard controller is used to generate an INIT signal to the CPU.

Name	Type	Description
RTCALE/ GPO25	O	**REAL TIME CLOCK ADDRESS LATCH ENABLE.** 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 on falling IOW# and remains asserted for two SYSCLKs. If the internal Real Time Clock is used, this signal can be programmed as a general purpose output. **During Reset:** Low **After Reset:** Low **During POS:** Low/GPO
RTCCS#/ GPO24	O	**REAL TIME CLOCK CHIP SELECT.** RTCCS# is asserted during read or write I/O 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. If the internal Real Time Clock is used, this signal can be programmed as a general purpose output. **During Reset:** High **After Reset:** High **During POS:** High/GPO
XDIR#/ GPO22	O	**X-BUS TRANSCEIVER DIRECTION.** XDIR# is tied directly to the direction control of a 74’245 that buffers the X-Bus data, XD[7:0]. XDIR# is asserted (driven low) for all I/O read cycles regardless if the accesses is to a PIIX4 supported device. XDIR# is asserted for memory cycles only if BIOS or APIC space has been decoded. For PCI master initiated read cycles, XDIR# is asserted from the falling edge of either IOR# or MEMR# (from MEMR# only if BIOS or APIC 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, PIIX4 negates XDIR#. For DMA read cycles from the X-Bus, XDIR# is driven low from DACKx# falling and negated from DACKx# rising. At all other times, XDIR# is negated high. If the X-Bus not used, then this signal can be programmed to be a general purpose output. **During Reset:** High **After Reset:** High **During POS:** High/GPO
XOE#/ GPO23	O	**X-BUS TRANSCEIVER OUTPUT ENABLE.** XOE# is tied directly to the output enable of a 74’245 that buffers the X-Bus data, XD[7:0], from the system data bus, SD[7:0]. XOE# is asserted anytime a PIIX4 supported X-Bus device is decoded, and the devices decode is enabled in the X-Bus Chip Select Enable Register (BIOSCS#, KBCCS#, RTCCS#, MCCS#) or the Device Resource B (PCCS0#) and Device Resource C (PCCS1#). 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 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. If an X-Bus not used, then this signal can be programmed to be a general purpose output. **During Reset:** High **After Reset:** High **During POS:** High/GPO

2.1.4. DMA SIGNALS

Name	Type	Description
DACK[0,1,2,3]# **DACK[5,6,7]#**	O	**DMA ACKNOWLEDGE.** The DACK# output lines indicate that a request for DMA service has been granted by PIIX4 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#. If the DREQ goes inactive prior to DACK# being asserted, the DACK# signal will not be asserted. **During Reset:** High **After Reset:** High **During POS:** High
DREQ[0,1,2,3] **DREQ[5,6,7]**	I	**DMA REQUEST.** The DREQ lines are used to request DMA service from PIIX4'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. All inactive to active edges of DREQ are assumed to be asynchronous. The request must remain active until the appropriate DACKx# signal is asserted.
REQ[A:C]#/ GPI[2:4]	I	**PC/PCI DMA REQUEST.** These signals are the DMA requests for PC/PCI protocol. They are used by a PCI agent to request DMA services and follow the PCI Expansion Channel Passing protocol as defined in the *PCI DMA* section. If the PC/PCI request is not needed, these pins can be used as general-purpose inputs.
GNT[A:C]#/ GPO[9:11]	O	**PC/PCI DMA ACKNOWLEDGE.** These signals are the DMA grants for PC/PCI protocol. They are used by a PIIX4 to acknowledge DMA services and follow the PCI Expansion Channel Passing protocol as defined in the *PCI DMA* section. If the PC/PCI request is not needed, these pins can be used as general-purpose outputs. **During Reset:** High **After Reset:** High **During POS:** High/GPO
TC	O	**TERMINAL COUNT.** PIIX4 asserts TC to DMA slaves as a terminal count indicator. PIIX4 asserts TC after a new address has been output, if the byte count expires with that transfer. TC remains asserted until AEN is negated, unless AEN is negated during an autoinitialization. TC is negated before AEN is negated during an autoinitialization. **During Reset:** Low **After Reset:** Low **During POS:** Low

2.1.5. INTERRUPT CONTROLLER/APIC SIGNALS

Name	Type	Description
APICACK#/ GPO12	O	**APIC ACKNOWLEDGE.** This active low output signal is asserted by PIIX4 after its internal buffers are flushed in response to the APICREQ# signal. When the I/O APIC samples this signal asserted it knows that PIIX4's buffers are flushed and that it can proceed to send the APIC interrupt. The APICACK# output is synchronous to PCICLK. If the external APIC is not used, then this is a general-purpose output. **During Reset:** High **After Reset:** High **During POS:** High/GPO
APICCS#/ GPO13	O	**APIC CHIP SELECT.** 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 I/O APIC address space. If the external APIC is not used, this pin is a general-purpose output. **During Reset:** High **After Reset:** High **During POS:** High/GPO
APICREQ#/ GPI5	I	**APIC REQUEST.** This active low input signal is asserted by an external APIC device prior to sending an interrupt over the APIC serial bus. When PIIX4 samples this pin active it will flush its F-type DMA buffers pointing towards PCI. Once the buffers are flushed, PIIX4 asserts APICACK# which indicates to the external APIC that it can proceed to send the APIC interrupt. The APICREQ# input must be synchronous to PCICLK. If the external APIC is not used, this pin is a general-purpose input.
INTR	OD	**INTERRUPT.** See CPU Interface Signals.
IRQ0/ GPO14	O	**INTERRUPT REQUEST 0.** This output reflects the state of the internal IRQ0 signal from the system timer. If the external APIC is not used, this pin is a general-purpose output. **During Reset:** Low **After Reset:** Low **During POS:** IRQ0/GPO
IRQ1	I	**INTERRUPT REQUEST 1.** IRQ1 is always edge triggered and can not be modified by software to level sensitive. A low to high transition on IRQ1 is latched by PIIX4. IRQ1 must remain asserted until after the interrupt is acknowledged. If the input goes inactive before this time, a default IRQ7 is reported in response to the interrupt acknowledge cycle.

Name	Type	Description
IRQ 3:7, 9:11, 14:15	I	**INTERRUPT REQUESTS 3:7, 9:11, 14:15.** The IRQ signals provide both system board components and ISA Bus I/O devices with a mechanism for asynchronously interrupting the CPU. These interrupts may be programmed for either an edge sensitive or a high level sensitive assertion mode. Edge sensitive is the default configuration. An active IRQ input must remain asserted until after the interrupt is acknowledged. If the input goes inactive before this time, a default IRQ7 is reported in response to the interrupt acknowledge cycle.
IRQ8#/ GPI6	I/O	**IRQ 8#.** IRQ8# is always an active low edge triggered interrupt and can not be modified by software. IRQ8# must remain asserted until after the interrupt is acknowledged. If the input goes inactive before this time, a default IRQ7 is reported in response to the interrupt acknowledge cycle. If using the internal RTC, then this can be programmed as a general-purpose input. If enabling an APIC, this signal becomes an output and must not be programmed as a general purpose input.
IRQ9OUT#/ GPO29	O	**IRQ9OUT#.** IRQ9OUT# is used to route the internally generated SCI and SMBus interrupts out of the PIIX4 for connection to an external IO APIC. If APIC is disabled, this signal pin is a General Purpose Output. **During Reset:** High **After Reset:** High **During POS:** IRQ9OUT#/GPO
IRQ 12/M	I	**INTERRUPT REQUEST 12.** In addition to providing the standard interrupt function as described in the pin description for IRQ[3:7,9:11,14:15], this pin can also be programmed to provide the mouse interrupt function. When the mouse interrupt function is selected, a low to high transition on this signal is latched by PIIX4 and an INTR is generated to the CPU as IRQ12. An internal IRQ12 interrupt continues to be generated until a Reset or an I/O read access to address 60h (falling edge of IOR#) is detected.
PIRQ[A:D]#	I/OD PCI	**PROGRAMMABLE INTERRUPT REQUEST.** The PIRQx# signals are active low, level sensitive, shareable interrupt inputs. They can be individually steered to ISA interrupts IRQ [3:7,9:12,14:15]. The USB controller uses PIRQD# as its output signal.
SERIRQ/ GPI7	I/O	**SERIAL INTERRUPT REQUEST.** Serial interrupt input decoder, typically used in conjunction with the Distributed DMA protocol. If not using serial interrupts, this pin can be used as a general-purpose input.

2.1.6. CPU INTERFACE SIGNALS

Name	Type	Description
A20M#	OD	**ADDRESS 20 MASK.** PIIX4 asserts A20M# to the CPU based on combination of Port 92 Register, bit 1 (FAST_A20), and A20GATE input signal. **During Reset:** High-Z **After Reset:** High-Z **During POS:** High-Z
CPURST	OD	**CPU RESET.** PIIX4 asserts CPURST to reset the CPU. PIIX4 asserts CPURST during power-up and when a hard reset sequence is initiated through the RC register. CPURST is driven inactive a minimum of 2 ms after PWROK is driven active. CPURST is driven active for a minimum of 2 ms when initiated through the RC register. The inactive edge of CPURST is driven synchronously to the rising edge of PCICLK. If a hard reset is initiated through the RC register, PIIX4 resets its internal registers (in both core and suspend wells) to their default state. This signal is active high for Pentium processor and active-low for Pentium II processor as determined by CONFIG1 signal. For values **During Reset**, **After Reset**, and **During POS**, see the *Suspend/Resume and Resume Control Signaling* section.
FERR#	I	**NUMERIC COPROCESSOR ERROR.** This pin functions as a FERR# signal supporting coprocessor errors. This signal is tied to the coprocessor error signal on the CPU. If FERR# is asserted, PIIX4 generates an internal IRQ13 to its interrupt controller unit. PIIX4 then asserts the INT 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.
IGNNE#	OD	**IGNORE NUMERIC EXCEPTION.** This signal is connected to the ignore numeric exception pin on the CPU. IGNNE# is only used if the PIIX4 coprocessor error reporting function is enabled. If FERR# is active, 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. **During Reset:** High-Z **After Reset:** High-Z **During POS:** High-Z
INIT	OD	**INITIALIZATION.** INIT is asserted in response to any one of the following conditions. When the System Reset bit in the Reset Control Register is reset to 0 and the Reset CPU bit toggles from 0 to 1, PIIX4 initiates a soft reset by asserting INIT. PIIX4 also asserts INIT if a Shut Down Special cycle is decoded on the PCI Bus, if the RCIN# signal is asserted, or if a write occurs to Port 92h, bit 0. When asserted, INIT remains asserted for approximately 64 PCI clocks before being negated. This signal is active high for Pentium processor and active-low for Pentium II processor as determined by CONFIG1 signal. **Pentium Processor:** **During Reset:** Low **After Reset:** Low **During POS:** Low **Pentium II Processor:** **During Reset:** High **After Reset:** High **During POS:** High
INTR	OD	**CPU INTERRUPT.** INTR is driven by PIIX4 to signal the CPU that an interrupt request is pending and needs to be serviced. It is asynchronous with respect to SYSCLK or PCICLK and is always an output. The interrupt controller must be programmed following PCIRST# to ensure that INTR is at a known state. **During Reset:** Low **After Reset:** Low **During POS:** Low

Name	Type	Description
NMI	OD	**NON-MASKABLE INTERRUPT.** NMI is used to force a nonmaskable interrupt to the CPU. PIIX4 generates an NMI when either SERR# or IOCHK# is asserted, depending on how the NMI Status and Control Register is programmed. The CPU detects an NMI when it detects a rising edge on NMI. After the NMI interrupt routine processes the interrupt, the NMI status bits in the NMI Status and Control Register are cleared by software. The NMI interrupt routine must read this register to determine the source of the interrupt. The NMI is reset by setting the corresponding NMI source enable/disable bit in the NMI Status and Control Register. To enable NMI interrupts, the two NMI enable/disable bits in the register must be set to 0, and the NMI mask bit in the NMI Enable/Disable and Real Time Clock Address Register must be set to 0. Upon PCIRST#, this signal is driven low. **During Reset:** Low **After Reset:** Low **During POS:** Low
SLP#	OD	**SLEEP.** This signal is output to the Pentium II processor in order to put it into Sleep state. For Pentium processor it is a No Connect. **During Reset:** High-Z **After Reset:** High-Z **During POS:** High-Z
SMI#	OD	**SYSTEM MANAGEMENT INTERRUPT.** SMI# is an active low synchronous output that is asserted by PIIX4 in response to one of many enabled hardware or software events. The CPU recognizes the falling edge of SMI# as the highest priority interrupt in the system, with the exception of INIT, CPURST, and FLUSH. **During Reset:** High-Z **After Reset:** High-Z **During POS:** High-Z
STPCLK#	OD	**STOP CLOCK.** STPCLK# is an active low synchronous output that is asserted by PIIX4 in response to one of many hardware or software events. STPCLK# connects directly to the CPU and is synchronous to PCICLK. **During Reset:** High-Z **After Reset:** High-Z **During POS:** High-Z

2.1.7. CLOCKING SIGNALS

Name	Type	Description
CLK48	I	**48-MHZ CLOCK.** 48-MHz clock used by the internal USB host controller. This signal may be stopped during suspend modes.
PCICLK	I	**FREE-RUNNING PCI CLOCK.** A clock signal running at 30 or 33 MHz, 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. Because many of the circuits in PIIX4 run off the PCI clock, this signal MUST be kept active, even if the PCI bus clock is not active.
OSC	I	**14.31818-MHZ CLOCK.** Clock signal used by the internal 8254 timer. This clock signal may be stopped during suspend modes.
RTCX1, RTCX2	I/O	**RTC CRYSTAL INPUTS:** These connected directly to a 32.768-kHz crystal. External capacitors are required. These clock inputs are required even if the internal RTC is not being used.
SUSCLK	O	**SUSPEND CLOCK.** 32.768-kHz output clock provided to the Host-to-PCI bridge used for maintenance of DRAM refresh. This signal is stopped during Suspend-to-Disk and Soft Off modes. For values During Reset, After Reset, and During POS, see the *Suspend/Resume and Resume Control Signaling* section.
SYSCLK	O	**ISA SYSTEM CLOCK.** SYSCLK is the reference clock for the ISA bus. It drives the ISA bus directly. The SYSCLK is generated by dividing PCICLK by 4. The SYSCLK frequencies supported are 7.5 MHz and 8.33 MHz. For PCI accesses to the ISA bus, SYSCLK may be stretched low to synchronize BALE falling to the rising edge of SYSCLK. **During Reset:** Running **After Reset:** Running **During POS:** Low

2.1.8. IDE SIGNALS

Name	Type	Description
PDA[2:0]	O	**PRIMARY DISK ADDRESS[2:0].** These signals indicate which byte in either the ATA command block or control block is being addressed. If the IDE signals are configured for Primary and Secondary, these signals are connected to the corresponding signals on the Primary IDE connector. If the IDE signals are configured for Primary 0 and Primary 1, these signals are used for the Primary 0 connector. **During Reset:** High-Z **After Reset:** Undefined[1] **During POS:** PDA
PDCS1#	O	**PRIMARY DISK CHIP SELECT FOR 1F0H–1F7H RANGE.** For ATA command register block. If the IDE signals are configured for Primary and Secondary, this output signal is connected to the corresponding signal on the Primary IDE connector. If the IDE signals are configured for Primary Master and Primary Slave, this signal is used for the Primary Master connector. **During Reset:** High **After Reset:** High **During POS:** High

Name	Type	Description
PDCS3#	O	**PRIMARY DISK CHIP SELECT FOR 3F0–3F7 RANGE.** For ATA control register block. If the IDE signals are configured for Primary and Secondary, this output signal is connected to the corresponding signal on the Primary IDE connector. If the IDE signals are configured for Primary Master and Primary Slave, this signal is used for the Primary Master connector. **During Reset:** High **After Reset:** High **During POS:** High
PDD[15:0]	I/O	**PRIMARY DISK DATA[15:0].** These signals are used to transfer data to or from the IDE device. If the IDE signals are configured for Primary and Secondary, these signals are connected to the corresponding signals on the Primary IDE connector. If the IDE signals are configured for Primary Master and Primary Slave, this signal is used for the Primary Master connector. **During Reset:** High-Z **After Reset:** Undefined[1] **During POS:** PDD
PDDACK#	O	**PRIMARY DMA ACKNOWLEDGE.** This signal directly drives the IDE device DMACK# signal. It is asserted by PIIX4 to indicate to IDE DMA slave devices that a given data transfer cycle (assertion of PDIOR# or PDIOW#) is a DMA data transfer cycle. This signal is used in conjunction with the PCI bus master IDE function. It is not associated with any AT compatible DMA channel. If the IDE signals are configured for Primary and Secondary, this signal is connected to the corresponding signal on the Primary IDE connector. If the IDE signals are configured for Primary Master and Primary Slave, this signal is used for the Primary Master connector. **During Reset:** High **After Reset:** High **During POS:** High
PDDREQ	I	**PRIMARY DISK DMA REQUEST.** This input signal is directly driven from the IDE device DMARQ signal. It is asserted by the IDE device to request a data transfer, and used in conjunction with the PCI bus master IDE function. It is not associated with any AT compatible DMA channel. If the IDE signals are configured for Primary and Secondary, this signal is connected to the corresponding signal on the Primary IDE connector. If the IDE signals are configured for Primary Master and Primary Slave, this signal is used for the Primary Master connector.

Name	Type	Description
PDIOR#	O	**PRIMARY DISK IO READ.** In normal IDE this is the command to the IDE device that it may drive data onto the PDD[15:0] lines. Data is latched by PIIX4 on the negation edge of PDIOR#. The IDE device is selected either by the ATA register file chip selects (PDCS1#, PDCS3#) and the PDA[2:0] lines, or the IDE DMA slave arbitration signals (PDDACK#). In an Ultra DMA/33 read cycle, this signal is used as DMARDY# which is negated by the PIIX4 to pause Ultra DMA/33 transfers. In an Ultra DMA/33 write cycle, this signal is used as the STROBE signal, with the drive latching data on rising and falling edges of STROBE. If the IDE signals are configured for Primary and Secondary, this signal is connected to the corresponding signal on the Primary IDE connector. If the IDE signals are configured for Primary Master and Primary Slave, this signal is used for the Primary Master connector. **During Reset:** High **After Reset:** High **During POS:** High
PDIOW#	O	**PRIMARY DISK IO WRITE.** In normal IDE mode, this is the command to the IDE device that it may latch data from the PDD[15:0] lines. Data is latched by the IDE device on the negation edge of PDIOW#. The IDE device is selected either by the ATA register file chip selects (PDCS1#, PDCS3#) and the PDA[2:0] lines, or the IDE DMA slave arbitration signals (PDDACK#). For Ultra DMA/33 mode, this signal is used as the STOP signal, which is used to terminate an Ultra DMA/33 transaction. If the IDE signals are configured for Primary and Secondary, this signal is connected to the corresponding signal on the Primary IDE connector. If the IDE signals are configured for Primary Master and Primary Slave, this signal is used for the Primary Master connector. **During Reset:** High **After Reset:** High **During POS:** High-Z
PIORDY	I	**PRIMARY IO CHANNEL READY.** In normal IDE mode, this input signal is directly driven by the corresponding IDE device IORDY signal. In an Ultra DMA/33 read cycle, this signal is used as STROBE, with the PIIX4 latching data on rising and falling edges of STROBE. In an Ultra DMA/33 write cycle, this signal is used as the DMARDY# signal which is negated by the drive to pause Ultra DMA/33 transfers. If the IDE signals are configured for Primary and Secondary, this signal is connected to the corresponding signal on the Primary IDE connector. If the IDE signals are configured for Primary Master and Primary Slave, this signal is used for the Primary Master connector. This is a Schmitt triggered input.

Name	Type	Description
SDA[2:0]	O	**SECONDARY DISK ADDRESS[2:0].** These signals indicate which byte in either the ATA command block or control block is being addressed. If the IDE signals are configured for Primary and Secondary, these signals are connected to the corresponding signals on the Secondary IDE connector. If the IDE signals are configured for Primary Master and Primary Slave, these signals are used for the Primary Slave connector. **During Reset:** High-Z **After Reset:** Undefined[1] **During POS:** SDA
SDCS1#	O	**SECONDARY CHIP SELECT FOR 170H–177H RANGE.** For ATA command register block. If the IDE signals are configured for Primary and Secondary, this output signal is connected to the corresponding signal on the Secondary IDE connector. If the IDE signals are configured for Primary Master and Primary Slave, these signals are used for the Primary Slave connector. **During Reset:** High **After Reset:** High **During POS:** High
SDCS3#	O	**SECONDARY CHIP SELECT FOR 370H–377H RANGE.** For ATA control register block. If the IDE signals are configured for Primary and Secondary, this output signal is connected to the corresponding signal on the Secondary IDE connector. If the IDE signals are configured for Primary Master and Primary Slave, these signals are used for the Primary Slave connector. **During Reset:** High **After Reset:** High **During POS:** High-Z
SDD[15:0]	I/O	**SECONDARY DISK DATA[15:0].** These signals are used to transfer data to or from the IDE device. If the IDE signals are configured for Primary and Secondary, these signals are connected to the corresponding signals on the Secondary IDE connector. If the IDE signals are configured for Primary Master and Primary Slave, these signals are used for the Primary Slave connector. **During Reset:** High-Z **After Reset:** Undefined[1] **During POS:** SDD
SDDACK#	O	**SECONDARY DMA ACKNOWLEDGE.** This signal directly drives the IDE device DMACK# signal. It is asserted by PIIX4 to indicate to IDE DMA slave devices that a given data transfer cycle (assertion of SDIOR# or SDIOW#) is a DMA data transfer cycle. This signal is used in conjunction with the PCI bus master IDE function. It is not associated with any AT compatible DMA channel. If the IDE signals are configured for Primary and Secondary, this signal is connected to the corresponding signal on the Secondary IDE connector. If the IDE signals are configured for Primary Master and Primary Slave, these signals are used for the Primary Slave connector. **During Reset:** High **After Reset:** High **During POS:** High

Name	Type	Description
SDDREQ	I	**SECONDARY DISK DMA REQUEST.** This input signal is directly driven from the IDE device DMARQ signal. It is asserted by the IDE device to request a data transfer, and used in conjunction with the PCI bus master IDE function. It is not associated with any AT compatible DMA channel. If the IDE signals are configured for Primary and Secondary, this signal is connected to the corresponding signal on the Secondary IDE connector. If the IDE signals are configured for Primary Master and Primary Slave, these signals are used for the Primary Slave connector.
SDIOR#	O	**SECONDARY DISK IO READ.** In normal IDE mode, this is the command to the IDE device that it may drive data onto the SDD[15:0] lines. Data is latched by the PIIX4 on the negation edge of SDIOR#. The IDE device is selected either by the ATA register file chip selects (SDCS1#, SDCS3#) and the SDA[2:0] lines, or the IDE DMA slave arbitration signals (SDDACK#). In an Ultra DMA/33 read cycle, this signal is used as DMARDY# which is negated by the PIIX4 to pause Ultra DMA/33 transfers. In an Ultra DMA/33 write cycle, this signal is used as the STROBE signal, with the drive latching data on rising and falling edges of STROBE. If the IDE signals are configured for Primary and Secondary, this signal is connected to the corresponding signal on the Secondary IDE connector. If the IDE signals are configured for Primary Master and Primary Slave, these signals are used for the Primary Slave connector. **During Reset:** High **After Reset:** High **During POS:** High
SDIOW#	O	**SECONDARY DISK IO WRITE.** In normal IDE mode, this is the command to the IDE device that it may latch data from the SDD[15:0] lines. Data is latched by the IDE device on the negation edge of SDIOW#. The IDE device is selected either by the ATA register file chip selects (SDCS1#, SDCS3#) and the SDA[2:0] lines, or the IDE DMA slave arbitration signals (SDDACK#). In read and write cycles this signal is used as the STOP signal, which is used to terminate an Ultra DMA/33 transaction. If the IDE signals are configured for Primary and Secondary, this signal is connected to the corresponding signal on the Secondary IDE connector. If the IDE signals are configured for Primary Master and Primary Slave, these signals are used for the Primary Slave connector. **During Reset:** High **After Reset:** High **During POS:** High

Name	Type	Description
SIORDY	I	**SECONDARY IO CHANNEL READY.** In normal IDE mode, this input signal is directly driven by the corresponding IDE device IORDY signal. In an Ultra DMA/33 read cycle, this signal is used as STROBE, with the PIIX4 latching data on rising and falling edges of STROBE. In an Ultra DMA write cycle, this signal is used as the DMARDY# signal which is negated by the drive to pause Ultra DMA/33 transfers. If the IDE signals are configured for Primary and Secondary, this signal is connected to the corresponding signal on the Secondary IDE connector. If the IDE signals are configured for Primary Master and Primary Slave, these signals are used for the Primary Slave connector. This is a Schmitt triggered input.

NOTES:

1. After reset, all undefined signals on the primary channel will default to the same values as the undefined signals on the secondary channel.

2.1.9. UNIVERSAL SERIAL BUS SIGNALS

Name	Type	Description
OC[1:0]#	I	**OVER CURRENT DETECT.** These signals are used to monitor the status of the USB power supply lines. The corresponding USB port is disabled when its over current signal is asserted.
USBP0+, USBP0–	I/O	**SERIAL BUS PORT 0.** This signal pair comprises the differential data signal for USB port 0. **During Reset:** High-Z **After Reset:** High-Z **During POS:** High-Z
USBP1+, USBP1–	I/O	**SERIAL BUS PORT 1.** This signal pair comprises the differential data signal for USB port 1. **During Reset:** High-Z **After Reset:** High-Z **During POS:** High-Z

2.1.10. POWER MANAGEMENT SIGNALS

Name	Type	Description
BATLOW#/ GPI9	I	**BATTERY LOW.** Indicates that battery power is low. PIIX4 can be programmed to prevent a resume operation when the BATLOW# signal is asserted. If the Battery Low function is not needed, this pin can be used as a general-purpose input.
CPU_STP#/ GPO17	O	**CPU CLOCK STOP.** Active low control signal to the clock generator used to disable the CPU clock outputs. If this function is not needed, then this signal can be used as a general-purpose output. For values **During Reset**, **After Reset**, and **During POS**, see the *Suspend/Resume and Resume Control Signaling* section.

Name	Type	Description
EXTSMI#	I/OD	**EXTERNAL SYSTEM MANAGEMENT INTERRUPT.** EXTSMI# is a falling edge triggered input to PIIX4 indicating that an external device is requesting the system to enter SMM mode. When enabled, a falling edge on EXTSMI# results in the assertion of the SMI# signal to the CPU. EXTSMI# is an asynchronous input to PIIX4. However, when the setup and hold times are met, it is only required to be asserted for one PCICLK. Once negated EXTSMI# must remain negated for at least four PCICLKs to allow the edge detect logic to reset. EXTSMI# is asserted by PIIX4 in response to SMI# being activated within the Serial IRQ function. An external pull-up should be placed on this signal.
LID/ GPI10	I	**LID INPUT.** This signal can be used to monitor the opening and closing of the display lid of a notebook computer. It can be used to detect both low to high transition or a high to low transition and these transitions will generate an SMI# if enabled. This input contains logic to perform a 16-ms debounce of the input signal. If the LID function is not needed, this pin can be used as a general-purpose input.
PCIREQ[A:D]#	I	**PCI REQUEST.** Power Management input signals used to monitor PCI Master Requests for use of the PCI bus. They are connected to the corresponding REQ[0:3]# signals on the Host Bridge.
PCI_STP#/ GPO18	O	**PCI CLOCK STOP.** Active low control signal to the clock generator used to disable the PCI clock outputs. The PIIX4 free running PCICLK input must remain on. If this function is not needed, this pin can be used as a general-purpose output. For values **During Reset**, **After Reset**, and **During POS**, see the *Suspend/Resume and Resume Control Signaling* section.
PWRBTN#	I	**POWER BUTTON.** Input used by power management logic to monitor external system events, most typically a system on/off button or switch. This input contains logic to perform a 16-ms debounce of the input signal.
RI# GPI12	I	**RING INDICATE.** Input used by power management logic to monitor external system events, most typically used for wake up from a modem. If this function is not needed, then this signal can be individually used as a general-purpose input.
RSMRST#	I	**RESUME RESET.** This signal resets the internal Suspend Well power plane logic and portions of the RTC well logic.
SMBALERT#/ GPI11	I	**SM BUS ALERT.** Input used by System Management Bus logic to generate an interrupt (IRQ or SMI) or power management resume event when enabled. If this function is not needed, this pin can be used as a general-purpose input.
SMBCLK	I/O	**SM BUS CLOCK.** System Management Bus Clock used to synchronize transfer of data on SMBus. **During Reset:** High-Z **After Reset:** High-Z **During POS:** High-Z
SMBDATA	I/O	**SM BUS DATA.** Serial data line used to transfer data on SMBus. **During Reset:** High-Z **After Reset:** High-Z **During POS:** High-Z
SUSA#	O	**SUSPEND PLANE A CONTROL.** Control signal asserted during power management suspend states. SUSA# is primarily used to control the primary power plane. This signal is asserted during POS, STR, and STD suspend states. **During Reset:** Low **After Reset:** High **During POS:** Low

Name	Type	Description
SUSB#/ GPO15	O	**SUSPEND PLANE B CONTROL.** Control signal asserted during power management suspend states. SUSB# is primarily used to control the secondary power plane. This signal is asserted during STR and STD suspend states. If the power plane control is not needed, this pin can be used as a general-purpose output. **During Reset:** Low **After Reset:** High **During POS:** High/GPO
SUSC#/ GPO16	O	**SUSPEND PLANE C CONTROL.** Control signal asserted during power management suspend states, primarily used to control the tertiary power plane. It is asserted only during STD suspend state. If the power plane control is not needed, this pin can be used as a general-purpose output. **During Reset:** Low **After Reset:** High **During POS:** High/GPO
SUS_STAT1#/ GPO20	O	**SUSPEND STATUS 1.** This signal is typically connected to the Host-to-PCI bridge and is used to provide information on host clock status. SUS_STAST1# is asserted when the system may stop the host clock, such as Stop Clock or during POS, STR, and STD suspend states. If this function is not needed, this pin can be used as a general-purpose output. **During Reset:** Low **After Reset:** High **During POS:** Low/GPO
SUS_STAT2#/ GPO21	O	**SUSPEND STATUS 2.** This signal will typically connect to other system peripherals and is used to provide information on system suspend state. It is asserted during POS, STR, and STD suspend states. If this function is not needed, this pin can be used as a general-purpose output. **During Reset:** Low **After Reset:** High **During POS:** Low/GPO
THRM#/ GPI8	I	**THERMAL DETECT.** Active low signal generated by external hardware to start the Hardware Clock Throttling mode. If enabled, the external hardware can force the system to enter into Hardware Clock Throttle mode by asserting THRM#. This causes PIIX4 to cycle STPCLK# at a preset programmable rate. If this function is not needed, this pin can be used as a general-purpose input.
ZZ/ GPO19	O	**LOW-POWER MODE FOR L2 CACHE SRAM.** This signal is used to power down a cache's data SRAMs when the clock logic places the CPU into the Stop Clock. If this function is not needed, this pin can be used as a general-purpose output. **During Reset:** Low **After Reset:** Low **During POS:** Low

2.1.11. GENERAL PURPOSE INPUT AND OUTPUT SIGNALS

Some of the General Purpose Input and Output signals are multiplexed with other PIIX4 signals. The usage is determined by the system configuration.

The default pin usage is shown in Table 1 and Table 2. The configuration can be selected via the General Configuration register and X-Bus Chip Select register.

Name	Type	Description
GPI[21:0]	I	**GENERAL PURPOSE INPUTS.** These input signals can be monitored via the GPIREG register located in Function 3 (Power Management) System IO Space at address PMBase+30h. See Table 1 for details.

Name	Type	Description
GPO[30:0]	O	**GENERAL PURPOSE OUTPUTS.** These output signals can be controlled via the GPIREG register located in Function 3 (Power Management) System IO Space at address PMBase+34h. If a GPO pin is not multiplexed with another signal or defaults to GPO, then its state after reset is the reset condition of the GPOREG register. If the GPO defaults to another signal, then it defaults to that signal's state after reset. The GPO pins that default to GPO remain stable after reset. The others may toggle due to system boot or power control sequencing after reset prior to their being programmed as GPOs. The GPO8 signal is driven low upon removal of power from the PIIX4 core power plane. All other GPO signals are invalid (buffers powered off).

Table 1. GPI Signals

Signal Name	Multiplexed With	Default	Control Register and Bit (PCI Function 1)	Notes
GPI0	IOCHK#	GPI	GENCFG Bit 0	Available as GPI only if in EIO bus mode.
GPI1#		GPI		Non-multiplexed GPI which is always available. This signal when used by power management logic is active low.
GPI[2:4]	REQ[A:C]#	GPI	GENCFG Bits 8–10	Not available as GPI if used for PC/PCI. Can be individually enabled, so for instance, GPI[4] is available if REQ[C]# is not used.
GPI5	APICREQ#	GPI	XBCS Bit 8	Not available as GPI if using an external APIC.
GPI6	IRQ8#	GPI	GENCFG Bit 14	Not available as GPI if using external RTC or external APIC.
GPI7	SERIRQ	GPI	GENCFG Bit 16	Not available as GPI if using Serial IRQ protocol.
GPI8	THRM#	THRM#	GENCFG Bit 23	Not available as GPI if using thermal monitoring.
GPI9	BATLOW#	BATLOW#	GENCFG Bit 24	Not available as GPI if using battery low feature.
GPI10	LID#	LID	GENCFG Bit 25	Not available as GPI if using LID feature.
GPI11	SMBALERT#	SMBALERT#	GENCFG Bit 15	Not available as GPI if using SMBALERT feature
GPI12	RI#	RI#	GENCFG Bit 27	Not available if using ring indicator feature
GPI[13:21]		GPI		Non-multiplexed GPIs which are always available.

Table 2. GPO Signals

Signal Name	Multiplexed With	Default	Control Register and Bit (PCI Function 1)	Notes
GPO0		GPO		Non-multiplexed GPO which is always available.
GPO[1:7]	LA[17:23]	GPO	GENCFG Bit 0	Available as GPO only if EIO mode.
GPO8		GPO		Non-multiplexed GPO which is always available. The GPO[8] signal will be driven low upon removal of power from the PIIX4 core power plane.
GPO[9:11]	GNT[A:C]#	GPO	GENCFG Bits [8:10]	Not available as GPO if using for PC/PCI. Can be individually enabled, so GPO[11] is available if REQ[C]# not used.
GPO12	APICACK#	GPO	XBCS Bit 8	Not available as GPO if using external APIC.
GPO13	APICCS#	GPO	XBCS Bit 8	Not available as GPO if using external APIC.
GPO14	IRQ0	GPO	XBCS Bit 8	Not available as GPO if using external APIC.
GPO15	SUSB#	SUSB#	GENCFG Bit 17	Not available as GPO if using for power management.
GPO16	SUSC#	SUSC#	GENCFG Bit 17	Not available as GPO if using for power management.
GPO17	CPU_STP#	CPU_STP#	GENCFG Bit 18	Not available as GPO if using for clock control.
GPO18	PCI_STP#	PCI_STP#	GENCFG Bit 19	Not available as GPO if using for clock control.
GPO19	ZZ	ZZ	GENCFG Bit 20	Not available as GPO if using for power management.
GPO20	SUS_STAT1#	SUS_STAT1#	GENCFG Bit 21	Not available as GPO if using for power management.
GPO21	SUS_STAT2#	SUS_STAT2#	GENCFG Bit 22	Not available as GPO if using for power management.
GPO22	XDIR#	XDIR#	GENCFG Bit 28	Not available as GPO if using X-bus transceiver.
GPO23	XOE#	XOE#	GENCFG Bit 28	Not available as GPO if using X-bus transceiver.
GPO24	RTCCS#	RTCCS#	GENCFG Bit 29	Not available as GPO if using external RTC that doesn’t do self decode.

Table 2. GPO Signals

Signal Name	Multiplexed With	Default	Control Register and Bit (PCI Function 1)	Notes
GPO25	RTCALE	RTCALE	GENCFG Bit 30	Not available as GPO if using external RTC that doesn't do self decode.
GPO26	KBCCS#	KBCCS#	GENCFG Bit 31	Not available as GPO if using external KBC that doesn't do self decode.
GPO[27:28]		GPO		Non-multiplexed GPOs which are always available.
GPO29	IRQ9OUT#	GPO	XBCS Bit 8	Not available as GPO if using external APIC. This signal is used for IRQ9 output in APIC mode, where it is level triggered, active low.
GPO30		GPO		Non-multiplexed GPO which is always available.

2.1.12. OTHER SYSTEM AND TEST SIGNALS

Name	Type	Description
CONFIG1	I	**CONFIGURATION SELECT 1.** This input signal is used to select the type of microprocessor being used in the system. If CONFIG1=0, the system contains a Pentium microprocessor. If CONFIG1=1, the system contains a Pentium II microprocessor. It is used to control the polarity of INIT and CPURST signals.
CONFIG2	I	**CONFIGURATION SELECT 2.** This input signal is used to select the positive or subtractive decode of FFFF0000h–FFFFFFFFh memory address range (top 64 Kbytes). If CONFIG[2]=0, the PIIX4 will positively decode this range. If CONFIG[2]=1, the PIIX4 will decode this range with subtractive decode timings only. The input value of this pin must be static and may not dynamically change during system operations.
PWROK	I	**POWER OK.** When asserted, PWROK is an indication to PIIX4 that power and PCICLK have been stable for at least 1 ms. PWROK can be driven asynchronously. When PWROK is negated, PIIX4 asserts CPURST, PCIRST# and RSTDRV. When PWROK is driven active (high), PIIX4 negates CPURST, PCIRST#, and RSTDRV.
SPKR	O	**SPEAKER.** The SPKR signal is the output of counter timer 2 and is internally “ANDed” with Port 061h bit 1 to provide the Speaker Data Enable. This signal drives an external speaker driver device, which in turn drives the ISA system speaker. **During Reset:** Low **After Reset:** Low **During POS:** Last State
TEST#	I	**TEST MODE SELECT.** The test signal is used to select various test modes of PIIX4. This signal must be pulled up to V_{CC}(SUS) for normal operation.

2.1.13. POWER AND GROUND PINS

Name	Type	Description
V_{CC}	V	**CORE VOLTAGE SUPPLY.** These pins are the primary voltage supply for the PIIX4 core and IO periphery and must be tied to 3.3V.
V_{CC} (RTC)	V	**RTC WELL VOLTAGE SUPPLY.** This pin is the supply voltage for the RTC logic and must be tied to 3.3V.
V_{CC} (SUS)	V	**SUSPEND WELL VOLTAGE SUPPLY.** These pins are the primary voltage supply for the PIIX4 suspend logic and IO signals and must be tied to 3.3V.
V_{CC} (USB)	V	**USB VOLTAGE SUPPLY.** This pin is the supply voltage for the USB input/output buffers and must be tied to 3.3V.
V_{REF}	V	**VOLTAGE REFERENCE.** This pin is used to provide a 5V reference voltage for 5V safe input buffers. V_{REF} must be tied to 5V in a system requiring 5V tolerance. In a 5V tolerant system, this signal must power up before or simultaneous to V_{CC}. It must power down after or simultaneous to V_{CC}. In a non-5V tolerant system (3.3V only), this signal can be tied directly to V_{CC}. There are then no sequencing requirements.
V_{SS}	V	**CORE GROUND.** These pins are the primary ground for PIIX4.
V_{SS} (USB)	V	**USB GROUND.** This pin is the ground for the USB input/output buffers.

2.2. Power Planes

PIIX4 has three primary internal power planes. These power planes permit parts of PIIX4 to power down to conserve battery life. Table 3 shows the internal planes and their uses.

Table 3. PIIX4 Internal Power Planes

Power Plane	Description	Signals Powered	VCC Pins	GND Pins
RTC	Contains the real-time clock and 256 bytes of battery-backed SRAM. This plane is always powered if the internal RTC is used. If the internal RTC is not used, it may be connected to the suspend plane. Typically, powered via "coin-cell" lithium battery. The input signals attached to the RTC power plane DO NOT SUPPORT 5 VOLT INPUT LEVELS. These signals must not exceed VCC (RTC). There is no reset signal for this power plane.	PWROK, RSMRST#, RTCX1, RTCX2	VCC (RTC)	VSS
SUSPEND	Contains the logic needed to resume from the Suspend-to-Disk and Suspend-to-RAM states. This plane will typically be powered by a power supply which is capable of providing a "trickle" current. The input signals attached to the SUSPEND power plane DO NOT SUPPORT 5 VOLT INPUT LEVELS. These signals must not exceed VCC (SUS). This plane is reset by assertion of the RSMRST# signal.	BATLOW#, CONFIG[1:2], EXTSMI# GPI1, GPO8, IRQ8#, LID, RI# SMBALERT#, SMBCLK SMBDATA, PWRBTN# SUS[A:C]#, SUSCLK SUS_STAT[1:2]#, TEST#	VCC (SUS)	VSS
USB	Contains the USB input/output buffers.	USBP0+, USBP0– USBP1+, USBP1–	VCC (USB)	VSS (USB)
Core	Contains all the rest of the PIIX4 logic. This plane is powered by the main system power supply. All input signals within this plane are 5V tolerant except FERR#. This plane is reset by negation of the PWROK signal.	All Other Signal Pins	VCC	VSS

2.3. Power Sequencing Requirements

There are no power sequencing requirements for the various VCC power supplies to PIIX4. The VREF signal must be tied to 5V in a system requiring 5V tolerance. In a 5V tolerant system, this signal must power up before or simultaneous to VCC. It must power down after or simultaneous to VCC. In a non-5V tolerant system (3.3V only), this signal can be tied directly to VCC. There are then no sequencing requirements. Refer to Figure 2 for an example circuit schematic that may be used to ensure the proper VREF sequencing.

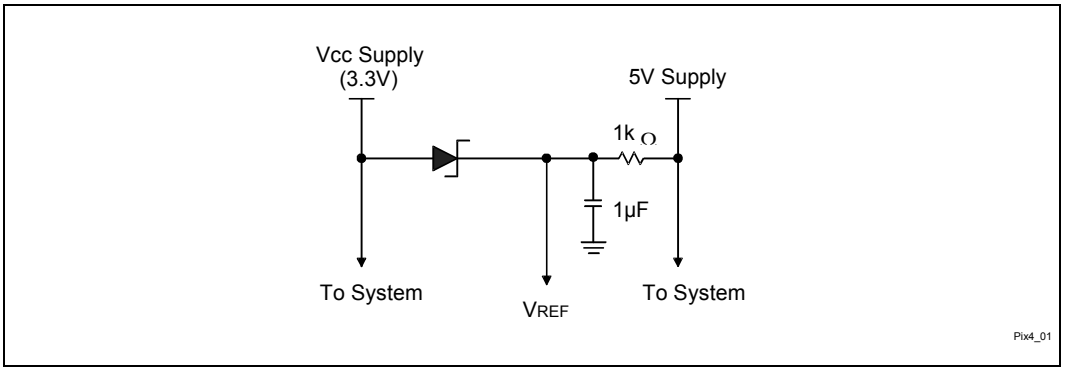


Figure 2. Example VCC5REF Sequencing Circuit

3.0. REGISTER ADDRESS SPACE

PIIX4 internal registers are organized into four functions—ISA Bridge with integrated AT compatibility logic, IDE Controller, USB Host Controller, and Enhanced Power Management. Each function has its registers divided into one set of PCI Configuration Registers and one or more register sets located in system IO space.

Some of the PIIX4 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 PIIX4 contains address locations in the PCI configuration space that are marked “Reserved.” The PIIX4 responds to accesses to these address locations by completing the Host cycle. Software should not write to reserved PIIX4 configuration locations in the device-specific region (above address offset 3Fh).

During a hard reset, the PIIX4 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.

3.1. PCI/ISA Bridge Configuration

The PIIX4 PCI function 0 contains a PCI to ISA bridge along with standard AT compatible logic including DMA controller, Interrupt controller, and counter/timers. This function also contains support for a real time clock and PCI based DMA. The register set associated with PCI to ISA Bridge and associated logic is shown below with actual register descriptions given in the “PCI to ISA/EIO Bridge Register Description” section.

3.1.1. PCI CONFIGURATION REGISTERS (FUNCTION 0)

Table 4. PCI Configuration Registers—Function 0 (PCI to ISA Bridge)

Offset Address	Mnemonic	Register Name	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/W
08h	RID	Revision Identification	RO
09–0Bh	CLASSC	Class Code	RO
0C–0Dh	—	Reserved	—
0Eh	HEDT	Header Type	RO
0F–4Bh	—	Reserved	—
4Ch	IOR	ISA I/O Recovery Timer	R/W
4Dh	—	Reserved	—
4E–4Fh	XBCS	X-Bus Chip Select	R/W
50–5Fh	—	Reserved	—
60–63h	PIRQRC[A:D]	PIRQx Route Control	R/W
64h	SERIRQC	Serial IRQ Control	R/W
65–68h	—	Reserved	—
69h	TOM	Top of Memory	R/W
6A–6Bh	MSTAT	Miscellaneous Status	R/W
6C–75h	—	Reserved	—
76–77h	MBDMA[1:0]	Motherboard Device DMA Control	R/W
78–7Fh	—	Reserved	—
80h	APICBASE	APIC Base Address Relocation	R/W
81h	—	Reserved	—
82h	DLC	Deterministic Latency Control	R/W
83–8Fh	—	Reserved	—
90–91h	PDMACFG	PCI DMA Configuration	R/W
92–95h	DDMABASE	Distributed DMA Slave Base Pointer	R/W
96–AFh	—	Reserved	—
B0–B3h	GENCFG	General Configuration	R/W
B4–CAh	—	Reserved	—
CBh	RTCCFG	Real Time Clock Configuration	R/W
CC–FFh	—	Reserved	—

3.1.2. IO SPACE REGISTERS

Table 5. ISA-Compatible Registers

Address	Aliased Addresses	Type	Register Name	Access
0000h	0010h	R/W	DMA1 CH0 Base and Current Address (CH0)	PCI
0001h	0011h	R/W	DMA1 CH0 Base and Current Count (CH0)	PCI
0002h	0012h	R/W	DMA1 CH1 Base and Current Address (CH1)	PCI
0003h	0013h	R/W	DMA1 CH1 Base and Current Count (CH1)	PCI
0004h	0014h	R/W	DMA1 CH2 Base and Current Address (CH2)	PCI
0005h	0015h	R/W	DMA1 CH2 Base and Current Count (CH2)	PCI
0006h	0016h	R/W	DMA1 CH3 Base and Current Address (CH3)	PCI
0007h	0017h	R/W	DMA1 CH3 Base and Current Count (CH3)	PCI
0008h	0018h	R/W	DMA1 Status(r) Command(w) Register	PCI
0009h	0019h	WO	DMA1 Request	PCI
000Ah	001Ah	WO	DMA1 Write Single Mask Bit	PCI
000Bh	001Bh	WO	DMA1 Channel Mode	PCI
000Ch	001Ch	WO	DMA1 Clear Byte Pointer	PCI
000Dh	001Dh	WO	DMA1 Master Clear	PCI
000Eh	001Eh	WO	DMA1 Clear Mask	PCI
000Fh	001Fh	R/W	DMA1 Read/Write All Mask Bits	PCI
0020h	0024h, 0028h, 002Ch, 0030h, 0034h, 0038h, 003Ch	R/W	Initialization Command Word 1 (INT-1) Operational Command Word 2 (INT-1) Operational Command Word 3 (INT-1)	PCI/ISA
0021h	0025h, 0029h, 002Dh, 0031h, 0035h, 0039h, 003Dh	R/W	Initialization Command Word 2 (INT-1) Initialization Command Word 3 (INT-1) Initialization Command Word 4 (INT-1) Operational Command Word 1 (INT-1)	PCI/ISA
0040h	0050h	R/W	Timer Count – Counter 0 Timer Status – Counter 0 (RO)	PCI/ISA
0041h	0051h	R/W	Timer Count – Counter 1 Timer Status – Counter 1 (RO)	PCI/ISA
0042h	0052h	R/W	Timer Count – Counter 2 Timer Status – Counter 2 (RO)	PCI/ISA
0043h	0053h	WO	Timer Control Word	PCI/ISA
0060h[1]		RO	Reset X-Bus IRQ12/M and IRQ1	PCI/ISA
0061h	0063h, 0065h, 0067h	R/W	NMI Status and Control	PCI/ISA

Table 5. ISA-Compatible Registers

Address	Aliased Addresses	Type	Register Name	Access
0070h[2]	0072h,[3] 0074h, 0076h	WO	NMI Enable	PCI/ISA
0070h[2]	0072h,[3] 0074h, 0076h	WO	RTC Index	PCI/ISA
0071h[2]	0073h,[4] 0075h, 0077h	R/W	RTC Data	PCI/ISA
0072h		R/W	RTC Extended Index	PCI/ISA
0073h		R/W	RTC Extended Data	PCI/ISA
0080h[5,6]	0090h	R/W	DMA1 Page (Reserved)	PCI/ISA
0081h[5]	0091h	R/W	DMA1 CH2 Low Page (CH2)	PCI/ISA
0082h[5]		R/W	DMA1 CH3 Low Page (CH3)	PCI/ISA
0083h[5]	0093h	R/W	DMA1 CH1 Low Page (CH1)	PCI/ISA
0084h[5,6]	0094h	R/W	DMA1 Page (Reserved)	PCI/ISA
0085h[5,6]	0095h	R/W	DMA1 Page (Reserved)	PCI/ISA
0086h[5,6]	0096h	R/W	DMA1 Page (Reserved)	PCI/ISA
0087h[5]	0097h	R/W	DMA1 CH0 Low Page (CH0)	PCI/ISA
0088h[5,6]	0098h	R/W	DMA Page (Reserved)	PCI/ISA
0089h[5]	0099h	R/W	DMA2 CH2 Low Page (CH6)	PCI/ISA
008Ah[5]	009Ah	R/W	DMA2 CH3 Low Page (CH7)	PCI/ISA
008Bh[5]	009Bh	R/W	DMA2 CH1 Low Page (CH5)	PCI/ISA
008Ch[5,6]	009Ch	R/W	DMA2 Page (Reserved)	PCI/ISA
008Dh[5,6]	009Dh	R/W	DMA2 Page (Reserved)	PCI/ISA
008Eh[5,6]	009Eh	R/W	DMA2 Page (Reserved)	PCI/ISA
008Fh[5]	009Fh	R/W	DMA2 Low Page Refresh	PCI/ISA
0092h		R/W	Port 92	PCI/ISA
00A0h	00A4h, 00A8h, 00ACh, 00B0h, 00B4h, 00B8h, 00BCh	R/W	Initialization Command Word 1 (INT-2) Operational Command Word 2 (INT-2) Operational Command Word 3 (INT-2)	PCI/ISA
00A1h	00A5h, 00A9h, 00Adh, 00B1h, 00B5h, 00B9h, 00BDh	R/W	Initialization Command Word 2 (INT-2) Initialization Command Word 3 (INT-2) Initialization Command Word 4 (INT-2) Operational Command Word 1 (INT-2)	PCI/ISA
00B2h		R/W	Advanced Power Management Control	PCI

Table 5. ISA-Compatible Registers

Address	Aliased Addresses	Type	Register Name	Access
00B3h		R/W	Advanced Power Management Status	PCI
00C0h	00C1h	R/W	DMA2 CH0 Base and Current Address (CH4)	PCI
00C2h	00C3h	R/W	DMA2 CH0 Base and Current Count (CH4)	PCI
00C4h	00C5h	R/W	DMA2 CH1 Base and Current Address (CH5)	PCI
00C6h	00C7h	R/W	DMA2 CH1 Base and Current Count (CH5)	PCI
00C8h	00C9h	R/W	DMA2 CH2 Base and Current Address (CH6)	PCI
00CAh	00CBh	R/W	DMA2 CH2 Base and Current Count (CH6)	PCI
00CCh	00CDh	R/W	DMA2 CH3 Base and Current Address (CH7)	PCI
00CEh	00CFh	R/W	DMA2 CH3 Base and Current Count (CH7)	PCI
00D0h	00D1h	R/W	DMA2 Status(r) Command(w)	PCI
00D2h	00D3h	WO	DMA2 Request	PCI
00D4h	00D5h	WO	DMA2 Write Single Mask Bit	PCI
00D6h	00D7h	WO	DMA2 Channel Mode	PCI
00D8h	00D9h	WO	DMA2 Clear Byte Pointer	PCI
00DAh	00DBh	WO	DMA2 Master Clear	PCI
00DCh	00DDh	WO	DMA2 Clear Mask	PCI
00DEh	00DFh	R/W	DMA2 Read/Write All Mask Register Bits	PCI
00F0h[1]		WO	Coprocessor Error	PCI/ISA
04D0h		R/W	INTC-1 Edge/Level Control	PCI/ISA
04D1h		R/W	INTC-2 Edge/Level Control	PCI/ISA
0CF9h		R/W	Reset Control	PCI

NOTES:

1. Read and write accesses to these locations are always broadcast to the ISA Bus.
2. Read and write accesses to these locations are broadcast to the ISA Bus, only if internal RTC is disabled in RTCCFG register.
3. Not aliased to 0072h or 0076h if extended RAM enabled.
4. Not aliased to 0073h or 0077h if extended RAM enabled.
5. The PIIX4 does not support Distributed DMA functionality for the 90h range, even if aliasing is enabled.
6. Write accesses to these locations are broadcast to the ISA Bus. Read accesses are not. If programmed in the ISA I/O Recovery Timer register, PIIX4 does not alias the entire 90h–9Fh address range. These locations are considered ISA Bus register locations and not PIIX4 registers.

3.2. IDE Configuration

The PIIX4 PCI function 1 contains an IDE Controller capable of standard Programmed IO (PIO) transfers as well as Bus Master transfer capability. This function also supports the “Ultra DMA/33” synchronous DMA mode of data transfer. The register set associated with IDE Controller is shown below with the actual register descriptions given in the “IDE Controller Register Descriptions” section.

3.2.1. PCI CONFIGURATION REGISTERS (FUNCTION 1)

Table 6. PCI Configuration Registers—Function 1 (IDE Interface)

Address Offset	Mnemonic	Register Name	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/W
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	R/W
44h	SIDETIM	Slave IDE Timing	R/W
45–47h	—	Reserved	—
48h	UDMACTL	Ultra DMA/33 Control	R/W
49h	—	Reserved	—
4A–4Bh	UDMATIM	Ultra DMA/33 Timing	R/W
4C–FFh	—	Reserved	—

3.2.2. IO SPACE REGISTERS

Table 7. PCI Bus Master IDE I/O Registers

Offset From Base Address	Mnemonic	Register Name	Access
00h	BMICP	Bus Master IDE Command (primary)	R/W
01h	—	Reserved	—
02h	BMISP	Bus Master IDE Status (primary)	R/W
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/W
0Bh	—	Reserved	—
0C–0Fh	BMIDTPS	Bus Master IDE Descriptor Table Pointer (secondary)	R/W

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

3.3. Universal Serial Bus (USB) Configuration

The PIIX4 PCI function 2 contains a Universal Serial Bus Host and Root Hub with two connected USB ports. This function supports the Universal Host Controller Interface (UHCI). The register set associated with USB Host Controller is shown below with actual register descriptions given in the “USB Host Controller Register Descriptions” section.

3.3.1. PCI CONFIGURATION REGISTERS (FUNCTION 2)

Table 8. PCI Configuration Registers—Function 2 (USB Interface)

Address Offset	Mnemonic	Register Name	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/W
08h	RID	Revision Identification	RO
09–0Bh	CLASSC	Class Code	RO
0Ch	—	Reserved	—
0Dh	MLT	Latency Timer	R/W
0Eh	HEDT	Header Type	RO
0F–1Fh	—	Reserved	—
20–23h	USBBA	USB IO Space Base Address	R/W
24–3Bh	—	Reserved	—

Table 8. PCI Configuration Registers—Function 2 (USB Interface)

Address Offset	Mnemonic	Register Name	Access
3Ch	INTLN	Interrupt Line	R/W
3Dh	INTPN	Interrupt Pin	RO
3E–5Fh	—	Reserved	—
60h	SBRNUM	Serial Bus Release Number	RO
61–BFh	—	Reserved	—
C0–C1h	LEGSUP	Legacy Support	R/W
C2–FFh	—	Reserved	RO

3.3.2. IO SPACE REGISTERS

Table 9. USB Host/Controller I/O Registers

Offset From Base Address	Mnemonic	Register Name	Access
00–01h	USBCMD	USB Command	R/W[2]
02–03h	USBSTS	USB Status	R/WC
04–05h	USBINTR	USB Interrupt Enable	R/W
06–07h	FRNUM	Frame Number	R/W[2]
08–0Bh	FLBASEADD	Frame List Base Address	R/W
0Ch	SOFMOD	Start Of Frame Modify	R/W
10–11h	PORTSC0	Port 0 Status and Control	R/WC[2]
12–13h	PORTSC1	Port 1 Status and Control	R/WC[2]

NOTES:

1. The base address is programmable via the USBBA Register (20–23h; function 2)
2. These registers are WORD writeable only. Byte writes to these registers have unpredictable effects.

3.4. Power Management Configuration

The PIIX4 PCI function 3 contains enhanced Power Management logic with support for Device Management, Suspend and Resume states, and System Clock Control. This function also supports a System Management Bus (SMBus) Host and Slave interface. The register set associated with Power Management and SMBus controller is shown below with actual register descriptions given in section 0.

Table 10. PCI CONFIGURATION REGISTERS (FUNCTION 3)

Address Offset	Mnemonic	Register Name	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
0F–3Bh	—	Reserved	—
3Ch	INTLN	Interrupt Line	R/W
3Dh	INTPN	Interrupt Pin	RO
3E–3Fh	—	Reserved	—
40–43h	PMBA	Power Management Base Address	R/W
44–47h	CNTA	Count A	R/W
48–4Bh	CNTB	Count B	R/W
4C–4Fh	GPICTL	General Purpose Input Control	R/W
50–52h	DEVRESD	Device Resource D	R/W
53h	—	Reserved	—
54–57h	DEVACTA	Device Activity A	R/W
58–5Bh	DEVACTB	Device Activity B	R/W
5C–5Fh	DEVRESA	Device Resource A	R/W
60–63h	DEVRESB	Device Resource B	R/W
64–67h	DEVRESC	Device Resource C	R/W
68–6Ah	DEVRESE	Device Resource E	R/W
6Bh	—	Reserved	—
6C–6Fh	DEVRESF	Device Resource F	R/W
70–72h	DEVRESG	Device Resource G	R/W
73h	—	Reserved	—
74–77h	DEVRESH	Device Resource H	R/W
78–7Bh	DEVRESI	Device Resource I	R/W

Table 10. PCI CONFIGURATION REGISTERS (FUNCTION 3)

Address Offset	Mnemonic	Register Name	Access
7C–7Fh	DEVRESJ	Device Resource J	R/W
80h	PMREGMISC	Miscellaneous Power Management	R/W
81–8Fh	—	Reserved	—
90–93h	SMBBA	SMBus Base Address	R/W
94–D1h	—	Reserved	—
D2h	SMBHSTCFG	SMBus Host Configuration	R/W
D3h	SMBREV	SMBus Revision ID	RO
D4h	SMBSLVC	SMBus Slave Command	R/W
D5h	SMBSHDW1	SMBus Slave Shadow Port 1	R/W
D6h	SMBSHDW2	SMBus Slave Shadow Port 2	R/W
D7–FFh	—	Reserved	—

3.4.1. IO SPACE REGISTERS

Table 11. Power Management I/O Registers

Offset From Base Address	Mnemonic	Register Name	Access
00–01h	PMSTS	Power Management Status	R/W
02–03h	PMEN	Power Management Resume Enable	R/W
04–05h	PMCNTRL	Power Management Control	R/W
06–07h	—	Reserved	—
08h	PMTMR	Power Management Timer	R/W
09–0Bh	—	Reserved	—
0C–0Dh	GPSTS	General Purpose Status	R/W
0E–0Fh	GPEN	General Purpose Enable	R/W
10–13H	PCNTRL	Processor Control	R/W
14h	PLVL2	Processor Level 2	R/W
15h	PLVL3	Processor Level 3	R/W
16–17h	—	Reserved	—
18–19h	GLBSTS	Global Status	R/W
1A–1Bh	—	Reserved	—
1Ch–1Fh	DEVSTS	Device Status	R/W
20–21h	GLBEN	Global Enable	R/W
22–27h	—	Reserved	—
28–2Bh	GLBCTL	Global Control	R/W
2C–2Fh	DEVCTL	Device Control	R/W

Table 11. Power Management I/O Registers

Offset From Base Address	Mnemonic	Register Name	Access
30–33h	GPIREG	General Purpose Input	RO
34–37h	GPOREG	General Purpose Output	R/W

NOTES:
1. The base address is programmable via the PMBA Register (40–43h; function 3)

Table 12. System Management Bus (SMBus) I/O Registers

Offset From Base Address	Mnemonic	Register Name	Access
00h	SMBHSTSTS	SMBus Host Status	R/W
01h	SMBSLVSTS	SMBus Slave Status	R/W
02h	SMBHSTCNT	SMBus Host Count	R/W
03h	SMBHSTCMD	SMBus Host Command	R/W
04h	SMBHSTADD	SMBus Host Address	R/W
05h	SMBHSTDAT0	SMBus Host Data 0	R/W
06h	SMBHSTDAT1	SMBus Host Data 1	R/W
07h	SMBBLKDAT	SMBus Block Data	R/W
08h	SMBSLVCNT	SMBus Slave Count	R/W
09h	SMBSHDWCMD	SMBus Shadow Command	R/W
0A–0Bh	SMBSLVEVT	SMBus Slave Event	R/W
0C–0Dh	SMBSLVDAT	SMBus Slave Data	R/W

NOTES:
1. The base address is programmable via the SMBBA Register (90–93h; function 3)

4.0. PCI TO ISA/EIO BRIDGE REGISTER DESCRIPTIONS

This section describes in detail the registers associated with the PIIX4 PCI-to-ISA Bridge function. This includes ISA/EIO configuration, AT compatible and PCI-based DMA control, standard AT and serial interrupt logic, counter/timers, real time clock and other miscellaneous functionality.

4.1. PCI to ISA/EIO Bridge PCI Configuration Space Registers (PCI Function 0)

4.1.1. 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.

4.1.2. DID—DEVICE IDENTIFICATION REGISTER (FUNCTION 0)

Address Offset: 02–03h
Default Value: 7110h
Attribute: Read Only

The DID Register contains the device identification number. This register, along with the VID Register, define the PIIX4. Writes to this register have no effect.

Bit	Description
15:0	**Device Identification Number.** This is a 16-bit value assigned to the PIIX4.

4.1.3. PCICMD—PCI COMMAND REGISTER (FUNCTION 0)

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

This 16-bit register provides basic control over the PIIX4'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	**SERR# Enable (SERRE).** 1=Enable. 0=Disable. When enabled (and DLC Register, bit 3=1), a delayed transaction time-out causes PIIX4 to assert the SERR# signal. The PCISTS register reports the status of the SERR# signal.
7	**Address and Data Stepping Enable (Not Implemented).** The PIIX4 does not support address and data stepping. This bit is hardwired to 0.
6	**Parity Error Detect Enable (Not Implemented).** PIIX4 does not support parity error detection. This bit is hardwired to 0.
5	**VGA Palette Snoop Enable (Not Implemented).** PIIX4 does not support VGA palette snooping. This bit is hardwired to 0.
4	**Memory Write and Invalidate Enable (Not Implemented).** PIIX4 does not generate Memory Write and Invalidate PCI transactions. This bit is hardwired to 0.
3	**Special Cycle Enable (SCE)**. 1=Enable, PIIX4 recognizes all PCI shutdown special cycles. 0=Disable, PIIX4 ignores all PCI Special Cycles.
2	**Bus Master Enable (Not Implemented).** PIIX4 does not support disabling its function 0 bus master capability. This bit is hardwired to 1.
1	**Memory Access (Not Implemented).** PIIX4 does not support disabling function 0 access to memory. This bit is hardwired to 1.
0	**I/O Space Access Enable (Not Implemented).** PIIX4 does not support disabling its function 0 response to PCI I/O cycles. This bit is hardwired to 1.

4.1.4. PCISTS—PCI DEVICE STATUS REGISTER (FUNCTION 0)

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

The PCISTS Register reports the occurrence of a PCI master-abort by PIIX4 or a PCI target-abort when PIIX4 is a master. The register also indicates PIIX4 DEVSEL# signal timing.

Bit	Description
15	**Detected Parity Error (Not Implemented).** Read as 0.
14	**Signaled SERR# Status (SERRS)—R/WC.** When PIIX4 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 (MAS)—R/WC.** When PIIX4, as a master (for function 0), generates a master-abort, MAS is set to a 1. Software sets MAS to 0 by writing a 1 to this bit location.
12	**Received Target-Abort Status (RTA)—R/WC.** When PIIX4 is a master on the PCI Bus (for function 0) 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 PIIX4 ISA bridge function is targeted with a transaction that PIIX4 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.** PIIX4 always generates DEVSEL# with medium timing for function 0 I/O cycles. Thus, DEVT=01. This DEVSEL# timing does not include Configuration cycles.
8	**PERR# Response (Not Implemented).** PIIX4 does not detect or respond to parity errors. Read as 0.
7	**Fast Back to Back—RO.** This bit indicates that PIIX4 as a target is capable of accepting fast back-to-back transactions.
6:0	**Reserved.** Read as 0s.

4.1.5. RID—REVISION IDENTIFICATION REGISTER (FUNCTION 0)

Address Offset: 08h
Default Value: Initial Stepping=00h. Refer to PIIX4 Specification Updates latest value.
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.

4.1.6. CLASSC—CLASS CODE REGISTER (FUNCTION 0)

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

This register identifies the Base Class Code, Sub-Class Code, and Device Programming interface for PIIX4 PCI function 0.

Bit	Description
23:16	**Base Class Code (BASEC).** 06h=Bridge device.
15:8	**Sub-Class Code (SCC)**. 01h=PCI-to-ISA Bridge. 80h=Other bridge Device (PCI Positive Decode Bridge). This value depends upon the programming of bit 1 of the General Configuration register (PCI function 0, address B0h). If programmed as a Subtractive Decode bridge (default), this will read 01h. If programmed as a Positive Decode bridge, this will read 80h.
7:0	**Programming Interface (PI).** 00h=No register level programming interface defined.

4.1.7. HEDT—HEADER TYPE REGISTER (FUNCTION 0)

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

The HEDT Register identifies PIIX4 as a multi-function device.

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

4.1.8. IORT—ISA I/O RECOVERY TIMER REGISTER (FUNCTION 0)

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

The I/O recovery mechanism in PIIX4 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. PIIX4 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, PIIX4 aliases PCI I/O accesses in the 90–9Fh range to the 80–8Fh range. In this case, PIIX4 only forwards PCI write accesses to 90–9Fh to the ISA Bus. ISA Master accesses to 90–9Fh range are forwarded to PCI. When DMAAC=1, PIIX4 disables aliasing for the entire 90–9Fh range (they are considered ISA bus register locations). PIIX4 forwards read and write accesses to these registers to ISA. ISA Master accesses to 90–9Fh range are ignored by PIIX4. Note, that port 92h is always a distinct register in the 90–9Fh range and is never forwarded from the PCI bus to the ISA Bus. It is also never forwarded from ISA to PCI or from ISA to the internal Port 92 register. PIIX4 does not support aliasing of the 90h range for the Distributed DMA function, even if aliasing is enabled.
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 use the recovery timing of 3.5 SYSCLKs.
5:3	**8-Bit I/O Recovery times.** When bit 6=1, this 3-bit field defines the additional number of SYSCLKs added to standard 3.5 SYSCLK 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 use the recovery timing of 3.5 SYSCLKs.
1:0	**16-Bit I/O Recovery Times.** When bit 2=1, this 2-bit defines the additional number of SYSCLKs added to standard 3.5 SYSCLK recovery time for 16-bit I/O. **Bit[1:0] SYSCLK** 00 3 01 1 10 2 11 4

4.1.9. XBCS—X-BUS CHIP SELECT REGISTER (FUNCTION 0)

Address Offset: 4E–4Fh
Default Value: 03h
Attribute: Read/Write

This register enables or disables accesses to an external RTC, keyboard controller, I/O APIC, a secondary controller, and BIOS. Disabling any of these bits prevents the device's chip select and X-Bus output enable control signal (XOE#) from being generated. This register also provides coprocessor error and mouse functions.

Bit	Description
15:11	**Reserved.**

Bit	Description
10	**Micro Controller Address Location Enable.** 1=Enable MCCS# and positive PCI decode for address locations 62h and 66h. 0=Disable MCCS# and positive PCI decode for accesses to these locations.
9	**1-Meg Extended BIOS Enable.** When bit 9=1, PCI master accesses to locations FFF00000–FFF7FFFFh are forwarded to ISA and result in the generation of BIOSCS# and XOE#. When forwarding the additional 512-Kbyte region, PIIX4 allows the PCI address A[23:20] to propagate to the ISA LA[23:20] lines as all 1's, aliasing this 512-Kbyte region to the top of the 16-Mbyte space. To avoid contention, ISA memory must not be present in this region (00F00000–00F7FFFFh). When bit 9=0, PIIX4 does not generate BIOSCS# or XOE# for accesses to this memory region.
8	**APIC Chip Select.** When enabled (bit 8=1), APICCS# is asserted for PCI memory accesses to the programmable I/O APIC region. This cycle is forwarded to the ISA bus. The default I/O APIC 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 PIIX4 and APICCS# and XOE# are not generated. Note that APICCS# is not generated for ISA-originated cycles. This bit is also used to select between GPIO functionality and APIC functionality on APICREQ#, APICACK#, APICCS#, IRQ0, IRQ8#, and IRQ9OUT# signals. When disabled, these signals become General Purpose Inputs or Outputs.
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, PIIX4 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 memory must not be present in this region (00F80000–00FDFFFFh). When bit 7=0, PIIX4 does not generate BIOSCS# or XOE# for accesses to this memory region.
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 memory must not be present in this region (00FE0000–00FEFFFFh). When bit 6=0, PIIX4 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. 0=Disable.
4	**IRQ12/M Mouse Function Enable.** 1=Mouse function; 0=Standard IRQ12 interrupt function.
3	**Port 61h Alias Enable.** 1=PIIX4 aliases accesses to 63h, 65h, and 67h to 61h. 0=PIIX4 does not alias 63h, 65h, and 67h to 61h.
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	**KBCCS# Enable.** 1=Enable KBCS# and XOE# for address locations 60h and 64h. 0=Disable KBCS#/XOE# for accesses to these locations.
0	**RTCCS#/RTCALE Enable.** 1=Enable RTCCS#/RTCALE and XOE# for accesses to address locations 70–77h. 0=Disable RTCCS#/RTCALE and XOE# for these accesses.

4.1.10. 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, PIIX4 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				

4.1.11. SERIRQC—SERIAL IRQ CONTROL REGISTER (FUNCTION 0)

Address Offset : 64h
Default Value: 10h
Attribute: R/W

This register controls the Start Frame Pulse Width generated on the Serial Interrupt signal (SERIRQ).

Bit	Description
7	**Serial IRQ Enable.** 1=Enable (bit 16 in register offset B0h–B3h must also be 1). 0=Disable.
6	**Serial IRQ Mode Select.** 1=Serial Interrupts operate in Continuous mode. 0=Serial Interrupts operate in Quiet mode.
5:2	**Serial IRQ Frame Size.** These bits select the frame size used by the Serial IRQ logic. The default is 0100b indicating a frame size of 21 (17+4). These bits are readable and writeable, however the only programmed value supported by PIIX4 is 0100b. All other frame sizes are unsupported.
1:0	**Start Frame Pulse Width.** These bits define the Start Frame pulse width generated by the Serial Interrupt control logic.

Bits[1:0]	Pulse Width (PCI Clocks)
00	4 Clocks
01	6 Clocks
10	8 Clocks
11	Reserved

4.1.12. 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).

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 PIIX4 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, PIIX4 Top Of Memory should be set at 15 Mbytes.
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	**640–768-Kbyte Memory Region (A0000–BFFFFh) Enable.** 1=Enable (ISA Master and DMA cycles forwarded to PCI); 0=Disable (contained to ISA).
1	**ISA/DMA 512–640-Kbyte Region Forwarding Enable**. 1=Enable (ISA Master and DMA cycles forwarded to PCI); 0=Disable (contained to ISA).
0	**Reserved.**

4.1.13. MSTAT—MISCELLANEOUS STATUS REGISTER (FUNCTION 0)

Address Offset: 6A–6Bh
Default Value: 0000h
Attribute: Read/Write

This register provides miscellaneous status and control functions.

Bit	Description
15	**SERR# Generation Due To Delayed Transaction Time-out—R/WC.** PIIX4 sets this bit to a 1 when it generates SERR# due to a delayed transaction time-out caused by expiration of the PCI delayed transaction discard timer. Software sets this bit to a 0 by writing a 1 to it.
14:8	**Reserved.**
7	**Host-to-PCI Bridge Retry Enable—R/W.** 1=Enable. 0=Disable. This bit, when enabled, causes PIIX4 to retry, without initiating a delayed transaction, CPU initiated, non-LOCK#, PCI cycles. No delayed transactions to PIIX4 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, PIIX4 accepts these cycles as normal, which may include retry with initiation of a delayed transaction.
6:0	**Reserved.**

4.1.14. MBDMA[1:0]—MOTHERBOARD DEVICE DMA CONTROL REGISTERS (FUNCTION 0)

Address Offset : 76h—MBDMA0#; 77h—MBDMA1#
Default Value: 04h
Attribute: R/W

These registers enable and disable a type F DMA transfer (3 SYSCLK) for a particular DMA channel.

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	**Reserved.** Read as 1.
2:0	**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

4.1.15. APICBASE—APIC BASE ADDRESS RELOCATION REGISTER (FUNCTION 0)

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 I/O APICs can be uniformly addresses in the memory space). The default value of 00h provides mapping of the I/O APIC 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 I/O APIC address ranges. When bit 6=1, address bit 12 is ignored allowing the APICCS# signal to be generated for two consecutive I/O APIC address ranges. External logic is needed to select individual I/O APICs 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 I/O APIC 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.

4.1.16. DLC—DETERMINISTIC LATENCY CONTROL REGISTER (FUNCTION 0)

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

This register enables and disables the Delayed Transaction and Passive Release functions. When enabled, these functions make PIIX4 PCI revision 2.1 compliant.

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. PIIX4 provides a programmable delayed completion mechanism described in the PCI specification to meet the required target latencies. This includes a Discard Timer which times out if a PCI Master with an outstanding delayed transaction has not retried the transaction for greater than 2^{15} PCI clocks.

ISA bridges also support Guaranteed Access Time (GAT) mode, which will now violate the spirit of the PCI specification. PIIX4 provides a programmable passive release mechanism to meet the required master latencies. When passive release is enabled in PIIX4, 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 Enable (Due To Delayed Transaction Time-out).** 1=Enable. 0=Disable.
2	**USB Passive Release Enable (USBPR).** 1=Enable. 0=Disable. When enabled, this allows PIIX4 to use Passive Release while transferring control information or data for USB transactions. When disabled, PIIX4 will perform PCI accesses for USB without using Passive Release.
1	**Passive Release Enable.** 1=Enable the Passive Release mechanism encoded on the PHOLD# signal when PIIX4 is a PCI Master. 0=Disable Passive Release.
0	**Delayed Transaction Enable.** 1=Enable the Delayed Transaction mechanism when PIIX4 is the target of a PCI transaction. 0=Disable Delayed Transaction mechanism.

4.1.17. PDMACFG—PCI DMA CONFIGURATION REGISTER (FUNCTION 0)

Address Offset: 90–91h
Default Value: 0000h
Attribute: Read/Write

This register defines the type of DMA performed by a particular DMA channel. If a channel is programmed for Distributed DMA mode, PIIX4 does not respond to either the ISA DREQ signal or to the PC/PCI encoding for that channel.

Bit	Description
15:14	**DMA CH 7 Select.** These bits define the type of DMA performed on this channel. **Bits[15:14] DMA Type** 00 Normal ISA DMA (default) 01 PC/PCI DMA 10 Distributed DMA 11 Reserved
13:12	**DMA CH 6 Select.** This field define the type of DMA performed on this channel. **Bits[13:12] DMA Type** 00 Normal ISA DMA (default) 01 PC/PCI DMA 10 Distributed DMA 11 Reserved
11:10	**DMA CH 5 Select.** These bits define the type of DMA performed on this channel. **Bits[11:10] DMA Type** 00 Normal ISA DMA (default) 01 PC/PCI DMA 10 Distributed DMA 11 Reserved
9:8	**Reserved.**

Bit	Description
7:6	**DMA CH 3 Select.** This field defines the type of DMA performed on this channel. **Bits[7:6] DMA Type** 00 Normal ISA DMA (default) 01 PC/PCI DMA 10 Distributed DMA 11 Reserved
5:4	**DMA CH 2 Select.** This field defines the type of DMA performed on this channel. **Bits[5:4] DMA Type** 00 Normal ISA DMA (default) 01 PC/PCI DMA 10 Distributed DMA 11 Reserved
3:2	**DMA CH 1 Select.** This field defines the type of DMA performed on this channel. **Bits[3:2] DMA Type** 00 Normal ISA DMA (default) 01 PC/PCI DMA 10 Distributed DMA 11 Reserved
1:0	**DMA CH 0 Select.** This field defines the type of DMA performed on this channel. **Bits[1:0] DMA Type** 00 Normal ISA DMA (default) 01 PC/PCI DMA 10 Distributed DMA 11 Reserved

4.1.18. DDMABP—DISTRIBUTED DMA SLAVE BASE POINTER REGISTERS (FUNCTION 0)

Address Offset: 92–93h (CH0-3); 94–95h (CH5-7)
Default Value: 0000h
Attribute: Read/Write

These registers provide the base address for distributed DMA slave channel registers, one for each DMA controller. Bits 5:0 are reserved to provide access to a 64-byte IO space (16 bytes per channel). The channels are accessed using offset from base address as follows (Note that Channel 4 is reserved and is not accessible).

Base Offset	Channel
00–0Fh	0,4
10–1Fh	1,5
20–2Fh	2,6
30–3Fh	3,7

Bit	Description
15:6	**Base Pointer.** IO Address pointer to DMA Slave Channel registers. Corresponds to PCI address AD[15:6].
5:0	**Reserved.** Read as 0.

4.1.19. GENCFG—GENERAL CONFIGURATION REGISTER (FUNCTION 0)

Address Offset: B0–B3h
Default Value: 0000h
Attribute: Read/Write

This register provides general system configuration for PIIX4, including signal and GPIO selects, ISA/EIO select, IDE signal configuration, and IDE signal enables.

Bit	Description
31	**KBCCS#/GPO26 Signal Pin Select.** 0=KBCCS# signal (default). 1=GPO26. This bit selects the functionality multiplexed onto the KBCCS# pin.
30	**RTCALE/GPO25 Signal Pin Select.** 0=RTCALE signal (default). 1=GPO25. This bit selects the functionality multiplexed onto the RTCALE pin.
29	**RTCCS#/GPO24 Signal Pin Select.** 0=RTCCS# signal (default). 1=GPO2. This bit selects the functionality multiplexed onto the RTCCS# pin.
28	**XOE# and XDIR#/GPO[22:23] Signal Pin Select.** 0=XOE# and XDIR# signals (default). 1=GPO[23] and GPO[22], respectively. This bit selects the functionality multiplexed onto the XOE# and XDIR# pins.
27	**Ring Indicate (RI#)/GPI12 Signal Pin Select.** 0=RI# signal (default). 1=GPI12. This bit selects the functionality multiplexed onto the RI# pin.
26	**Reserved.**
25	**LID/GPI10 Signal Pin Select.** 0=LID signal (default). 1=GPI10. This bit selects the functionality multiplexed onto the LID pin.
24	**BATLOW#/GPI9 Signal Pin Select.** 0=BATLOW# signal (default). 1=GPI9. This bit selects the functionality multiplexed onto the BATLOW# pin.
23	**THRM#/GPI8 Signal Pin Select.** 0=THRM# signal (default). 1=GPI8. This bit selects the functionality multiplexed onto the THRM# pin.
22	**SUS_STAT2#/GPO21 Signal Pin Select.** 0=SUS_STAT2# signal (default). 1=GPO21. This bit selects the functionality multiplexed onto the SUS_STAT2# pin.
21	**SUS_STAT1#/GPO20 Signal Pin Select.** 0=SUS_STAT1# signal (default). 1=GPO20. This bit selects the functionality multiplexed onto the SUS_STAT1# pin.
20	**ZZ/GPO19 Signal Pin Select.** 0=ZZ signal (default). 1=GPO19. This bit selects the functionality multiplexed onto the ZZ pin.
19	**PCI_STP#/GPO18 Signal Pin Select.** 0=PCI_STP# signal (default). 1=GPO18. This bit selects the functionality multiplexed onto the PCI_STP# pin.
18	**CPU_STP#/GPO17 Signal Pin Select.** 0=CPU_STP# signal (default). 1=GPO17. This bit selects the functionality multiplexed onto the CPU_STP# pin.
17	**SUSB# and SUSC#/GPO[15:16] Signal Pin Select.** 0=SUSB# and SUSC# signals (default). 1=GPO15 and GPO16 respectively. This bit selects the functionality multiplexed onto the SUSB# and SUSC# pins.

Bit	Description
16	**SERIRQ/GPI7 Signal Pin Select.** 0=GPI7 (default). 1=SERIRQ signal. This bit selects the functionality multiplexed onto the SERIRQ pin.
15	**SMBALERT#/GPI11 Signal Pin Select.** 0=SMBALERT# signal (default). 1=GPI11. This bit selects the functionality multiplexed onto the SMBALERT# pin.
14	**IRQ8#/GPI6 Signal Pin Select.** 0=GPI6 (default). 1=IRQ8# signal. This bit selects the functionality multiplexed onto the IRQ8# pin.
13	**Reserved.**
12	**Secondary IDE Signal Interface Tri-State.** 0=Enable Secondary IDE signal pin interface (default). 1=Tri-state (disable) Secondary IDE signal pin interface. This bit functions independently of bit 4.
11	**Primary IDE Signal Interface Tri-State.** 0=Enable Primary IDE signal pin interface (default). 1=Tri-state (disable) Primary IDE signal pin interface. This bit functions independently of bit 4.
10	**PC/PCI REQ[C] and GNT[C]/GPI4 and GPO11 Signal Pin Select.** 0=GPI4 and GPO11 (default). 1=PC/PCI REQC and GNTC respectively. This bit selects the functionality multiplexed onto the REQC and GNTC pins.
9	**PC/PCI REQ[B] and GNTB/GPI3 and GPO10 Signal Pin Select.** 0=GPI3 and GPO10 (default). 1=PC/PCI REQB and GNTB respectively. This bit selects the functionality multiplexed onto the REQB and GNTB pins.
8	**PC/PCI REQA and GNTA/GPI2 and GPO9 Signal Pin Select.** 0=GPI2 and GPO9 (default). 1=PC/PCI REQA and GNTA respectively. This bit selects the functionality multiplexed onto the REQA and GNTA pins.
7	**Reserved.**
6	**Plug and Play (PnP) Address Decode Enable.** 0=Disable PnP address positive decode (default). 1=Enable PnP address positive decode and forwarding to the ISA bus. The PnP addresses which are decoded are 279h and A79h. If bit 1 is set for positive decode, this bit must be set for these address to be forwarded to ISA.
5	**Alternate Access Mode Enable.** 0=Disable Alternate Access Mode (default). 1=Enables Alternate Access Mode to allow access to shadow registers as described in *the Power Management Functional Description* section. Enabling this bit allows special access to various internal PIIX4 registers. See special access restrictions prior to setting this bit.
4	**IDE Signal Configuration.** 0=Primary and Secondary interface enable (default). 1=Primary 0 and Primary 1 interface enable. This bit selects whether the IDE interfaces are split for Primary and Secondary channels allowing access to 4 IDE devices or are split into Primary Drive 0 and Primary Drive 1 channels allowing access to only the two Primary IDE devices.
3	**CONFIG 2 Status (RO).** This bit provides indication of signal present on CONFIG2 pin. Its meaning is currently undefined. The use of this pin is RESERVED and should be tied low through a pull down resistor.
2	**CONFIG 1 Status (RO).** 0=Pentium processor. 1=Pentium II processor. This bit provides indication of signal present on CONFIG1 pin. It is used to change the polarity of the INIT and CPURST signals to match the requirements of the microprocessors.

Bit	Description
1	**Positive or Subtractive Decode Configuration.** 0=Subtractive Decode (default). 1=Positive Decode. This bit determines how PIIX4 decodes accesses on the PCI bus for forwarding to ISA. If set for positive decode, PIIX4 positively decodes (with medium decode timing) and forwards PCI access to ISA only for address ranges which are enabled within PIIX4. If set for subtractive decode, PIIX4 positively decodes and forwards those cycles whose addresses are enabled within PIIX4. It subtractively decodes and forwards all other cycles not positively decoded by another device on the PCI bus. The setting and functionality of this bit is independent of bit 0, ISA or EIO Select.
0	**ISA or EIO Select.** 0=EIO (default). 1=ISA. This bit determines whether the expansion bus on PIIX4 supports the full Industry Standard Architecture (ISA) bus or whether it supports the Extended I/O (EIO) bus. See the *ISA/EIO Interface* section for details concerning ISA and EIO interface differences. This bit also selects the functionality multiplexed onto the IOCHK# and LA[17:23] pins. 0=GPI0 and GPO[1:7] (default). 1=IOCHK# and LA[17:23] respectively. These are the signals which are not used in EIO mode.

4.1.20. RTCCFG—REAL TIME CLOCK CONFIGURATION REGISTER (FUNCTION 0)

Address Offset: CBh
Default Value: 21h
Attribute: Read/Write

This register is used to configure the internal Real Time Clock.

Bit	Description
7:6	**Reserved.**
5	**RTC Positive Decode Enable.** 0=PIIX4 Subtractively Decodes for RTC I/O registers. 1=PIIX4 Positively Decodes for RTC I/O registers (default). The PCI cycles with addresses 70–73h are either positively or subtractively decoded based on this bit. The cycles are then routed to the internal RTC controller or forwarded to ISA based on bits 2 (extended bank) and bit 0 (standard bank) below. This bit should be set to 0 if PIIX4's internal RTC is not used and the external RTC is on the PCI bus, or if subtractive decode is desired for an external RTC on the ISA or X-Bus.
4	**Lock Upper RAM Bytes.** 0=Upper RAM data bytes 38h–3Fh in the extended bank are readable and writeable (default). 1=Upper RAM data bytes 38h–3Fh in the extended bank are neither readable nor writeable. This is used to lock bytes 38h–3Fh in the upper 128-byte bank of RAM. Write cycles will have no effect and read cycles will not return a guaranteed value. **Warning:** This is a write-once register that can only be reset by a hardware reset. No software means is possible to reset this bit.
3	**Lock Lower RAM Bytes.** 0=Lower RAM data bytes 38h–3Fh in the standard bank are readable and writeable (default). 1=Lower RAM data bytes 38h–3Fh in the standard bank are neither readable nor writeable. This is used to lock bytes 38h–3Fh in the lower 128-byte bank of RAM. Write cycles will have no effect and read cycles will not return a guaranteed value. **Warning:** This is a write-once register that can only be reset by a hardware reset. No software means is possible to reset this bit.

Bit	Description
2	**Upper RAM Enable.** 0=Accesses to RTC Upper 128-byte extended bank at I/O address 72–73h is disabled. Accesses will be forwarded to ISA bus as determined by bit 5 of this register (default). 1=Accesses to 72–73h are forwarded to RTC Upper 128-byte extended bank.
1	**Reserved.**
0	**RTC Enable.** 0=Accesses to RTC Lower 128-byte standard bank at I/O address 70–71h is disabled. Accesses will be forwarded to ISA bus as determined by bit 5 of this register. 1=Accesses to 70–71h are forwarded to RTC Lower 128-byte standard bank. When this bit is reset, the upper bank of RAM may still be accessed (enabled via bit 2 in this register).

4.2. PCI to ISA/EIO Bridge IO Space Registers (IO)

4.2.1. DMA REGISTERS

PIIX4 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.

4.2.1.1. DCOM—DMA Command Register (IO)

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.

4.2.1.2. DCM—DMA Channel Mode Register (IO)

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
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)

4.2.1.3. DR—DMA Request Register (IO)

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)

4.2.1.4. WSMB—Write Single Mask Bit (IO)

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)

4.2.1.5. RWAMB—Read/Write All Mask Bits (IO)

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)

4.2.1.6. DS—DMA Status Register (IO)

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)

4.2.1.7. DBADDR—DMA Base and Current Address Registers (IO)

I/O Address:	DMA Channel 0—000h	DMA Channel 4—0C0h
	DMA Channel 1—002h	DMA Channel 5—0C4h
	DMA Channel 2—004h	DMA Channel 6—0C8h
	DMA Channel 3—006h	DMA Channel 7—0CCh
Default Value:	Undefined (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.

4.2.1.8. DBCNT—DMA Base and Current Count Registers (IO)

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:	Undefined (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.

4.2.1.9. DLPAGE—DMA Low Page Registers (IO)

I/O Address: DMA Channel 0—087h, DMA Channel 1—083h, DMA Channel 2—081h, DMA Channel 3—082h, DMA Channel 5—08Bh, DMA Channel 6—089h, DMA Channel 7—08Ah
Default Value: Undefined (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.

4.2.1.10. DCBP—DMA Clear Byte Pointer Register (IO)

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.

4.2.1.11. DMC—DMA Master Clear Register (IO)

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.

4.2.1.12. DCLM—DMA Clear Mask Register (IO)

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.

4.2.2. INTERRUPT CONTROLLER REGISTERS

PIIX4 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.

4.2.2.1. ICW1—Initialization Command Word 1 Register (IO)

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 PIIX4-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 PIIX4 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 PIIX4.
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 PIIX4.
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.

4.2.2.2. ICW2—Initialization Command Word 2 Register (IO)

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.

4.2.2.3. ICW3—Initialization Command Word 3 Register (IO)

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.

4.2.2.4. ICW3—Initialization Command Word 3 Register (IO)

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.

4.2.2.5. ICW4—Initialization Command Word 4 Register (IO)

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

Both PIIX4 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.

4.2.2.6. OCW1—Operational Control Word 1 Register (IO)

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 as 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.

4.2.2.7. OCW2—Operational Control Word 2 Register (IO)

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-Spec 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)

4.2.2.8. OCW3—Operational Control Word 3 Register (IO)

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 does 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

4.2.2.9. ELCR1—Edge/Level Control Register (IO)

I/O Address: INT CNTRL-1—4D0h

Default Value: 00h
Attribute: Read/Write

ELCR1 register allows IRQ[3:7] to be edge or level programmable on an interrupt by interrupt basis. IRQ0, IRQ1 and IRQ2 are not programmable and are always edge sensitive. When level triggered, the interrupt is signaled active when input IRQ signal is high.

Bit	Description
7	**IRQ7 ECL.** 0=Edge Triggered mode; 1=Level Triggered mode.
6	**IRQ6 ECL.** 0=Edge Triggered mode; 1=Level Triggered mode.
5	**IRQ5 ECL.** 0=Edge Triggered mode; 1=Level Triggered mode.
4	**IRQ4 ECL.** 0=Edge Triggered mode; 1=Level Triggered mode.
3	**IRQ3 ECL.** 0=Edge Triggered mode; 1=Level Triggered mode.
2:0	**Reserved.** Must be 0.

4.2.2.10. ELCR2—Edge/Level Control Register (IO)

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. When level triggered, the interrupt is signaled active when input IRQ signal is high.

Bit	Description
7	**IRQ15 ECL.** 0=Edge Triggered mode; 1=Level Triggered mode.
6	**IRQ14 ECL.** 0=Edge Triggered mode; 1=Level Triggered mode.
5	**Reserved.** Must be 0.
4	**IRQ12 ECL.** 0=Edge Triggered mode; 1=Level Triggered mode.
3	**IRQ11 ECL.** 0=Edge Triggered mode; 1=Level Triggered mode.
2	**IRQ10 ECL.** 0=Edge Triggered mode; 1=Level Triggered mode.
1	**IRQ9 ECL.** 0=Edge Triggered mode; 1=Level Triggered mode.
0	**Reserved.** Must be 0.

4.2.3. COUNTER/TIMER REGISTERS

4.2.3.1. TCW—Timer Control Word Register (IO)

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=Binary 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.

Bit	Description
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.
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.

4.2.3.2. TMRSTS—Timer Status Registers (IO)

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.
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.

4.2.3.3. TMRCNT—Timer Count Registers (IO)

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.

4.2.4. 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.

4.2.4.1. NMISC—NMI Status and Control Register (IO)

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

This register reports the status of different system components, controls the output of the speaker counter (Counter 2), and gates 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.
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 PIIX4, the SERR# signal can be for a special protocol between the host-to-PCI bridge and PIIX4 (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.

4.2.4.2. NMIEN—NMI Enable Register (Shared with Real-Time Clock Index Register) (IO)

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. Reads to register 70h will cause X-Bus reads, but no RTCCS# or RTCALE will be generated. (The RTC has traditionally been write-only to port 70h.)

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

4.2.5. REAL TIME CLOCK REGISTERS

4.2.5.1. RTCI—Real-Time Clock Index Register (Shared with NMI Enable Register) (IO)

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

This port is shared with the NMI enable. 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. Reads to register 70h will cause X-Bus reads, but no RTCCS# or RTCALE will be generated. (The RTC has traditionally been write-only to port 70h.)

Bit	Description
7	**NMI Enable.** Used by PIIX4 NMI logic.
6:0	**Real Time Clock Address.** Latched by the Real Time Clock to address memory locations within the standard RAM bank accessed via the Real Time Clock Data Register (071h).

4.2.5.2. RTCD—Real-Time Clock Data Register (IO)

I/O Address: 071h
Default Value: Undefined
Attribute: Read/Write

The data port for accesses to the RTC standard RAM bank.

Bit	Description
7:0	**Standard RAM Data Port.** Data written to standard RAM bank address selected via RTC Index Register (070h).

4.2.5.3. RTCEI—Real-Time Clock Extended Index Register (IO)

I/O Address: 072h
Default Value: Unknown
Attribute: Write Only

The index port for accesses to the RTC extended RAM bank.

Bit	Description
7	**Reserved.**
6:0	**Real Time Clock Extended Address.** Latched by the Real Time Clock to address memory locations within the extended RAM bank accessed via the Real Time Clock Extended Data Register (073h).

4.2.5.4. RTCED—Real-Time Clock Extended Data Register (IO)

I/O Address: 073h
Default Value: Unknown
Attribute: Read/Write

The data port for accesses to the RTC extended RAM bank.

Bit	Description
7:0	**Extended RAM Data Port.** Data written to standard RAM bank address selected via RTC Extended Index Register (072h).

4.2.6. ADVANCED POWER MANAGEMENT (APM) REGISTERS

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

4.2.6.1. APMC—Advanced Power Management Control Port (IO)

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. PIIX4 operation is not affected 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 the APMC_EN bit (PCI function 3, offset 58h, bit 25) is set to 1. Reads do not generate an SMI.

4.2.6.2. APMS—Advanced Power Management Status Port (IO)

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

This register passes status information between the OS and the SMI handler. PIIX4 operation is not affected 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.

4.2.7. X-BUS, COPROCESSOR, AND RESET REGISTERS

4.2.7.1. RIRQ—Reset X-Bus IRQ12/M and IRQ1 Register (IO)

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

This register clears the mouse interrupt function (IRQ12/M) and the keyboard interrupt (IRQ1). Reads and writes to this address are accepted by PIIX4 and sent to ISA (Keyboard accesses must be enabled if in Positive decode). PIIX4 latches low to high transitions on IRQ1 and IRQ12/M (when enabled as mouse interrupt). A read of 60h clears the internally latched signals of IRQ1 and IRQ12/M.

Bit	Description
7:0	**Reset IRQ12 and IRQ1.** No specific pattern. A read of address 60h clears the internally latched IRQ1 and IRQ12/M signals.

4.2.7.2. P92—Port 92 Register (IO)

I/O Address: 92h
Default Value: 00h
Attribute: Read/Write

Bit	Description
7:2	**Reserved.** Returns 0 when read.
1	**FAST_A20**. 1=Causes A20M# signal to be asserted to 0. 0=A20M# signal determined by A20GATE signal. This signal is internally combined with the A20GATE input signal. The result is then output via the A20M# signal to the processor for support of real mode compatible software. The A20GATE signal generated by the keyboard is used in conjunction with the FAST_A20 bit in the P92 register to generate the A20M# signal that goes to the CPU. The A20M# signal is generated according to the following table: **Bit 1 — A20GATE (Input Signal) — A20M# (Output Signal)** 0 — Negated (Low) — Asserted (Low) 0 — Asserted (High) — Negated (High) 1 — Negated (Low) — Negated (High) 1 — Asserted (High) — Negated (High)
0	**FAST_INIT.** This read/write bit provides a fast software executed processor reset function. This function provides an alternate means to reset the system processor to effect a mode switch from Protected Virtual Address Mode to the Real Address Mode. This provides a faster means of reset than is provided by the Keyboard controller. Writing a 1 to this bit will cause the INIT signal to pulse active (high) for approximately 16 PCI Clocks. Before another INIT pulse can be generated via this register, this bit must be written back to a 0.

4.2.7.3. CERR—Coprocessor Error Register (IO)

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

Writing to this register causes PIIX4 to assert IGNNE#. PIIX4 also negates IRQ13 (internal to PIIX4). Note, that IGNNE# is not asserted unless FERR# is active. Reads/writes flow through to the ISA Bus.

Bit	Description
7:0	**Assert IGNNE#.** No special pattern required**.** A write to address F0h causes assertion of IGNNE# if FERR# is asserted.

4.2.7.4. RC—Reset Control Register (IO)

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

Bits 1 and 2 in this register are used by PIIX4 to generate a hard reset or a soft reset. During a hard reset, PIIX4 asserts CPURST, PCIRST#, and RSTDRV, as well as reset its core and suspend well logic. During a soft reset, PIIX4 asserts INIT.

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. PIIX4 will also initiate a hard reset when PWROK is asserted. This bit cannot be read as a 1.
1	**System Reset (SRST).** This bit is used to select the type of reset generated when bit 2 in this register is set to 1. When SRST=1, PIIX4 initiates a hard reset to the CPU when bit 2 in this register transitions from 0 to 1. When SRST=0, PIIX4 initiates a soft reset when bit 2 in this register transitions from 0 to 1.
0	**Reserved.**

5.0. IDE CONTROLLER REGISTER DESCRIPTIONS (PCI FUNCTION 1)

This section describes in detail the registers associated with PIIX4 IDE Controller function. This includes Programmed I/O (PIO), Bus Master, and “Ultra DMA/33” synchronous DMA functionality.

5.1. IDE Controller PCI Configuration Registers (PCI Function 1)

5.1.1. 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.

5.1.2. DID—DEVICE IDENTIFICATION REGISTER (FUNCTION 1)

Address Offset: 02–03h
Default Value: 7111h
Attribute: Read Only

The DID Register contains the device identification number. This register, along with the VID Register, define the PIIX4 function. Writes to this register have no effect.

Bit	Description
15:0	**Device Identification Number.** This is a 16-bit value assigned to the PIIX4 IDE Controller function.

5.1.3. PCICMD—PCI 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 (Not Implemented).** This bit is hardwired to 0.

Bit	Description
3	**Special Cycle Enable (Not Implemented).** This bit is hardwired to 0.
2	**Bus Master Function Enable (BME).** 1=Enable. 0=Disable.
1	**Memory Space Enable (Not Implemented).** This bit is hardwired to 0.
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.

5.1.4. 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 PIIX4's DEVSEL# signal timing.

Bit	Description
15	**Detected Parity Error (Not Implemented).** Read as 0.
14	**SERR# Status (Not Implemented).** Read as 0.
13	**Master-Abort Status (MAS)—R/WC.** When the Bus Master IDE interface function, as a master, generates a master abort, MAS is set to a 1. Software sets MAS to 0 by writing a 1 to this bit.
12	**Received Target-Abort Status (RTA)—R/WC.** 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/WC.** This bit is set when the PIIX4 IDE interface function is targeted with a transaction that PIIX4 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 PIIX4, DEVT=01 indicating medium timing for DEVSEL# assertion when performing a positive decode. 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 PIIX4 as a target, is capable of accepting fast back-to-back transactions.
6:0	**Reserved**. Read as 0's.

5.1.5. RID—REVISION IDENTIFICATION REGISTER (FUNCTION 1)

Address Offset: 08h
Default Value: Initial Stepping=00h. Refer to PIIX4 Specification Updates for other values programmed here.
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.

5.1.6. CLASSC—CLASS CODE REGISTER (FUNCTION 1)

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

This register identifies the Base Class Code, Sub Class Code, and Device Programming interface for PIIX4 PCI function 1.

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.

5.1.7. MLT—MASTER LATENCY TIMER REGISTER (FUNCTION 1)

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

MLT controls the amount of time PIIX4, 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. The Master Latency Timer is cleared and suspended when PIIX4 is not asserting FRAME#. When PIIX4 asserts FRAME#, the counter begins counting. If PIIX4 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), PIIX4 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 PIIX4.

Bit	Description
7:4	**Master Latency Timer Count Value (MLTC).** PIIX4-initiated PCI burst cycles can last indefinitely, as long as PHLDA# remains active. However, if PHLDA# is negated after the burst cycle is initiated, PIIX4 limits the burst cycle to the number of PCI Bus clocks specified by this field.
3:0	**Reserved.**

5.1.8. HEDT—HEADER TYPE REGISTER (FUNCTION 1)

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

This register identifies the IDE Controller module as a single function device.

Bit	Description
7:0	**Device Type (DEVICET).** 00. Multi-function device capability for PIIX4 is defined by the HEDT register in Function 0.

5.1.9. 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 (BMIBA).** 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.

5.1.10. IDETIM—IDE TIMING REGISTER (FUNCTION 1)

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

This register controls PIIX4's IDE interface and selects the timing characteristics of the PCI Local Bus IDE cycle for PIO and standard Bus Master transfers. Note that primary and secondary denotations distinguish between the cables and the 0/1 denotations distinguish between master (0) and slave (1). See Table 14 for programming values for various PIO Timing Modes.

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	**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 drive 0 through the fields in this register and these values can be programmed for each drive 1 through the SIDETIM Register. When disabled, the ISP and RTC values programmed in this register apply to both drive 0 and drive 1 on each channel.
13:12	**IORDY Sample Point (ISP).** This field selects the number of PCI 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 PCI 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=0, both DMA (bus master) and PIO data transfers for drive 1 use the fast timing mode (this is the preferred setting for optimal performance). When DTE1=1, fast timing mode is enabled for DMA data transfers for drive 1. PIO transfers 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. When PPE1=0, prefetch and posting is disabled 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 1, 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. 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 fast timings. PIO 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 always use the 8-bit compatible timings.
3	**DMA Timing Enable Only (DTE0).** When DTE1=0, both DMA and PIO data transfers for drive 0 use the fast timing mode (this is the preferred setting for optimal performance). When DTE0=1, fast timing mode is enabled for DMA data transfers for drive 0. PIO transfers run in compatible timing.
2	**Prefetch and Posting Enable (PPE0).** When PPE0=1, prefetch and posting to the IDE data port is enabled for drive 0. When PPE0=0, prefetch and posting is disabled 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. 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 fast timings. PIO 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 always use the 8-bit compatible timings.

5.1.11. SIDETIM—SLAVE IDE TIMING REGISTER (FUNCTION 1)

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

This register controls PIIX4'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. See Table 14 for programming values for various PIO Timing Modes.

Bit	Description
7:6	**Secondary Drive 1 IORDY Sample Point (SISP1).** This field selects the number of PCI clocks between SDIOx# assertion and the first SIORDY sample point for the slave drive on the secondary channel. **Bits[7:6] Number Of Clocks** 00 5 01 4 10 3 11 2
5:4	**Secondary Drive 1 Recovery Time (SRTC1).** This field selects the minimum number of PCI clocks between the last SIORDY# sample point and the SDIOx# 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 PCI clocks between PDIOx# assertion and the first PIORDY 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 PCI clocks between the last PIORDY# sample point and the PDIOx# 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

5.1.12. UDMACTL—ULTRA DMA/33 CONTROL REGISTER (FUNCTION 1)

Address Offset: 48h
Default Value: 00h
Attribute: Read/Write

This register enables each individual channel and drive for Ultra DMA/33 operation. For non-Ultra DMA/33 operation, this register should be left programmed to its default value.

Bit	Description
7:4	**Reserved.**
3	**Secondary Drive 1 UDMA Enable (SSDE1).** 1=Enable Ultra DMA/33 mode for secondary channel drive 1. 0=Disable (default).
2	**Secondary Drive 0 UDMA Enable (SSDE0).** 1=Enable Ultra DMA/33 mode for secondary channel drive 0. 0=Disable (default).
1	**Primary Drive 1 UDMA Enable (PSDE1).** 1=Enable Ultra DMA/33 mode for primary channel drive 1. 0=Disable (default).
0	**Primary Drive 0 UDMA Enable (PSDE0).** 1=Enable Ultra DMA/33 mode for primary channel drive 0. 0=Disable (default).

5.1.13. UDMATIM—ULTRA DMA/33 TIMING REGISTER (FUNCTION 1)

Address Offset: 4A–4Bh
Default Value: 00h
Attribute: Read/Write Only

This register controls the timings used by each Ultra DMA/33 enabled device. For non-Ultra DMA/33 operation, this register should be left programmed to its default value.

Bit	Description
15:14	**Reserved.**
13:12	**Secondary Drive 1 Cycle Time (SCT1).** These bit settings determine the minimum data write strobe Cycle Time (CT) and minimum Ready to Pause time (RP). **Bits[13:12] Time** 00 CT=4 PCICLK, RP=6 PCICLK 01 CT=3 PCICLK, RP=5 PCICLK 10 CT=2 PCICLK, RP=4 PCICLK 11 Reserved
11:10	**Reserved.**
9:8	**Secondary Drive 0 Cycle Time (SCT0).** These bit settings determine the minimum data write strobe Cycle Time (CT) and minimum Ready to Pause time (RP). **Bits[13:12] Time** 00 CT=4 PCICLK, RP=6 PCICLK 01 CT=3 PCICLK, RP=5 PCICLK 10 CT=2 PCICLK, RP=4 PCICLK 11 Reserved
7:6	**Reserved.**
5:4	**Primary Drive 1 Cycle Time (PCT1).** These bit settings determine the minimum data write strobe Cycle Time (CT) and minimum Ready to Pause time (RP). **Bits[13:12] Time** 00 CT=4 PCICLK, RP=6 PCICLK 01 CT=3 PCICLK, RP=5 PCICLK 10 CT=2 PCICLK, RP=4 PCICLK 11 Reserved
3:2	**Reserved.**
1:0	**Primary Drive 0 Cycle Time (PCT0).** These bit settings determine the minimum data write strobe Cycle Time (CT) and minimum Ready to Pause time (RP). **Bits[13:12] Time** 00 CT=4 PCICLK, RP=6 PCICLK 01 CT=3 PCICLK, RP=5 PCICLK 10 CT=2 PCICLK, RP=4 PCICLK 11 Reserved

Table 13. Ultra DMA/33 Timing Mode Settings

Bit Setting	Mode 0 (120 ns Strobe Period)	Mode 1 (90 ns Strobe Period)	Mode 2 (60 ns Strobe Period)
Cycle Time Bit Settings	00	01	10

Table 14. DMA/PIO Timing Values (Based on PIIX4 Cable Mode and System Speed)

PIIX4 Drive Mode	IORDY Sample Point (ISP)	Recovery Time (RCT)	IDETIM[15:8] Drive 0 (Master) If Slave Attached	IDETIM[15:8] Drive 0 (Master) If no Slave attached or Slave is Mode 0[1]	SIDETIM Pri[3:0] Sec[7:4] Drive 1 (Slave)	Resultant Cycle Time Base operating frequency and cycle time
PIO0/ Compatible	5 clocks (default)	4 clocks (default)	C0h	80h	0	30 MHz: 900 ns 33 MHz: 900 ns
PIO2/SW2	4 clocks	4 clocks	D0h	90h	4	30 MHz: 256 ns 33 MHz: 240 ns
PIO3/MW1	3 clocks	3 clocks	E1h	A1h	9	30 MHz: 198 ns 33 MHz: 180 ns
PIO4/MW2	3 clocks	1 clock	E3h	A3h	B	30 MHz: 132 ns 33 MHz: 120 ns

NOTES:

1. This table assumes that if the attached slave drive is Mode 0 or is not present, the SITRE bit is set to 0.
2. The table assumes that 25 MHz is not supported as a target PCI system speed. If the DMA Timing Enable Only (DTE) bit has been enabled for that drive, this resultant cycle time applies to data transfers performed with DMA only.

5.2. IDE Controller IO Space Registers

The PCI IDE function uses 16 bytes of I/O space, allocated via the BMIBA 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.

5.2.1. BMICX—BUS MASTER IDE COMMAND REGISTER (IO)

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. While a synchronous DMA transfer is in progress, this bit will be READ ONLY. The bit will return to read/write once the synchronous DMA transfer has been completed or halted.
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 while 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 by PIIX4 rather than 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.

5.2.2. BMISX—BUS MASTER IDE STATUS REGISTER (IO)

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 15 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 (IDEINTS)—R/WC.** This bit, when set to a 1, indicates when an IDE device has asserted its interrupt line. When bit 2=1, 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 IRQ15 is used for the secondary channel. Note that, if the interrupt status bit is set to a 0 by writing a 1 to this bit while the interrupt line 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 PIIX4 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.** PIIX4 sets this bit to 1 when bit 0 in the BMICx Register is set to 1. PIIX4 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). PIIX4 also sets this bit to 0 when bit 0 of the BMICx Register is set to 0. When this bit is read as a 0, 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 15. 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.
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 specified a smaller buffer size than the programmed IDE transfer size.

5.2.3. BMIDTPX—BUS MASTER IDE DESCRIPTOR TABLE POINTER REGISTER (IO)

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 (DTBA).** Bits [31:2] correspond to A[31:2].
1:0	**Reserved.**

6.0. USB HOST CONTROLLER REGISTER DESCRIPTIONS (PCI FUNCTION 2)

This section describes in detail the registers associated with the PIIX4 USB Host Controller function. This includes UHCI compatible registers and Legacy Keyboard registers.

6.1. USB Host Controller PCI Configuration Registers (PCI Function 2)

6.1.1. VID—VENDOR IDENTIFICATION REGISTER (FUNCTION 2)

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.

6.1.2. DID—DEVICE IDENTIFICATION REGISTER (FUNCTION 2)

Address Offset: 02–03h
Default Value: 7112h
Attribute: Read Only

The DID Register contains the device identification number. This register, along with the VID Register, define the PIIX4 USB Host Controller. Writes to this register have no effect.

Bit	Description
15:0	**Device Identification Number.** This is a 16-bit value assigned to the PIIX4 USB Host Controller.

6.1.3. PCICMD—PCI COMMAND REGISTER (FUNCTION 2)

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 (Not Implemented).** This bit is hardwired to 0.
8:5	**Reserved.** Read as 0.
4	**Memory Write and Invalidate Enable (Not Implemented).** This bit is hardwired to 0.
3	**Special Cycle Enable (Not Implemented).** This bit is hardwired to 0.
2	**Bus Master Enable (BME).** This bit controls PIIX4'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 USB transactions can start.
1	**Memory Space Enable (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.

6.1.4. PCISTS—PCI DEVICE STATUS REGISTER (FUNCTION 2)

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.

Bit	Description
15	**Detected Parity (Not Implemented).** Read as 0.
14	**SERR# Status (Not Implemented).** Read as 0.
13	**Master-Abort Status (MAS)—R/WC**. When the Serial Bus module receives a master-abort from a PCI transaction, MAS is set to a 1. Software sets MAS 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.

Bit	Description
10:9	**DEVSEL# Timing Status (DEVT)—RO.** This 2-bit field defines the timing for DEVSEL# assertion. These read only bits indicate PIIX4's DEVSEL# timing when performing a positive decode. Since PIIX4 always generate the DEVSEL# with medium timing, DEVT=01. This DEVSEL# timing does not include Configuration cycles.
8	**Data Parity Detected (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.

6.1.5. RID—REVISION IDENTIFICATION REGISTER (FUNCTION 2)

Address Offset: 08h
Default Value: Initial Stepping=00h. Refer to PIIX4 Specification Updates for other values programmed here.
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.

6.1.6. CLASSC—CLASS CODE REGISTER (FUNCTION 2)

Address Offset: 09–0Bh
Default Value: 0C0300h
Attribute: Read Only

This register identifies the Base Class Code, Sub Class Code, and Device Programming interface for PIIX4 PCI function 2.

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

6.1.7. MLT—MASTER LATENCY TIMER REGISTER (FUNCTION 2)

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 PIIX4 is not asserting FRAME#. When PIIX4 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, PIIX4 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 PIIX4, after which it must surrender the bus as soon as the current transaction is completed.

Bit	Description
7:4	**Master Latency Counter Value.** PIIX4 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, PIIX4 limits the duration of the transactions to the number of PCI bus clocks specified by this field.
3:0	**Reserved.**

6.1.8. HEDT—HEADER TYPE REGISTER (FUNCTION 2)

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 PIIX4 is defined by the HEDT register in Function 0.

6.1.9. INTLN—INTERRUPT LINE REGISTER (FUNCTION 2)

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 PIIX4 hardware operations.

6.1.10. INTPN—INTERRUPT PIN (FUNCTION 2)

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#.

6.1.11. SBRNUM—SERIAL BUS RELEASE NUMBER (FUNCTION 2)

Address Offset: 60h
Default Value: 10h
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

6.1.12. LEGSUP—LEGACY SUPPORT REGISTER (FUNCTION 2)

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 clears this bit by writing a 1 to it.
14	**Reserved.**
13	**USB PIRQ Enable (USBPIRQDEN)—R/W.** 1 (default)=USB interrupt is routed to PIRQD. 0=USB interrupt does not route to PIRQD. This bit prevents the USB controller from generating an interrupt. Note that it will probably 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.** This bit is set 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.** This bit is set 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.** This bit is set 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.** This bit is set 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.** This bit is set 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 (default)=Disable. 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 (default)=Not currently executing the A20GATE pass-through sequence. 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 (default)=Disable. 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 seen in the above sequence will cause 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# will not be generated during the sequence, even if the various enable bits are set. Note that during a Pass-through sequence, the above status bits will not be 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 (default)=Disable.
3	**Trap/SMI On 64h Write Enable (64WEN)—R/W.** 1=Enable I/O Trap and SMI# generation on port 64h write. 0 (default)=Disable.
2	**Trap/SMI On 64h Read Enable (64REN)—R/W.** 1=Enable I/O Trap and SMI# generation on port 64h read. 0 (default)=Disable.
1	**Trap/SMI On 60h Write Enable (60WEN)—R/W.** 1=Enable I/O Trap and SMI# generation on port 60h write. 0 (default)=Disable.
0	**Trap/SMI On 60h Read Enable (60REN)—R/W.** 1=Enable I/O Trap and SMI# generation on port 60h read. 0 (default)=Disable.

6.1.13. USBBA—USB I/O SPACE BASE ADDRESS REGISTER (FUNCTION 2)

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.

6.1.14. MISCSUP—MISCELLANEOUS SUPPORT REGISTER (FUNCTION 2)

Address Offset: FFh
Default Value: XXh
Attribute: Read/Write (Byte accesses only)

This register provides miscellaneous control capability for the PIIX4. The following programming model must be followed to read the RTC Index register.

1. Disable USB.
2. Read MISCSUP register.
3. Change the RTC Index Read Enable bit to 1 without changing the other register bits.
4. Write new value to MISCSUP register location.
5. Read the RTC Index register at I/O location 70h. Only bits [6:0] provide RTC Index value. Bit 7 is indeterminate.
6. Read the MISCSUP register.
7. Change the RTC Index Read Enable bit to 0 without changing the other register bits.
8. Write new value to MISCSUP register location.
9. Re-enable USB as desired.

Bit	Description
7:5	**Undefined.** Hardwired to 0s. Must be written as 0s.
4	**RTC Index Read Enable (RTCIREN).** 1=Enable reads to IO address 70h to return the value located in the RTC Index register. 0=Disable.
3:0	**Undefined.** Read as 0.

6.2. USB Host Controller IO Space Registers

6.2.1. USBCMD—USB COMMAND REGISTER (IO)

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

The Command Register indicates the command to be executed by the serial bus host controller. Writing to the register causes a command to be executed. The table following the bit description provides additional information on the operation of the Run/Stop and Debug bits.

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 which could be executed under bandwidth reclamation be within this size limit.

Bit	Description
6	**Configure Flag (CF).** HCD 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.
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 will remain a 1 until the EOP has completed.
3	**Enter Global Suspend Mode (EGSM).** 1=Host Controller enters the Global Suspend mode. No USB transactions occurs 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 USB 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, 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 effects on Hub registers are slightly different from 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 HCHalted 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 16. 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 HCHalted bit in the Status Register is set.

6.2.2. USBSTS—USB STATUS REGISTER (IO)

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

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 (debug mode or 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 TD. When this error occurs, the Host Controller clears the Run/Stop bit in the Command register to prevent further schedule execution. A hardware interrupt is generated to the system.
3	**Host System Error.** The Host Controller sets this bit to 1 when a serious error occurs during a host system access involving the Host Controller module. In a PCI system, conditions that set this bit to 1 include PCI Parity error, 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 interrupt is generated to the system.

Bit	Description
2	**Resume Detect.** 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 is in a global suspend 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 TD 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 packet detection is enabled in that TD.

6.2.3. USBINTR—USB INTERRUPT ENABLE REGISTER (IO)

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

This register enables and 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 by the host controller. 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 Interrupt Enable.** 1=Enabled. 0=Disabled.
0	**Time-out/CRC Interrupt Enable.** 1=Enabled. 0=Disabled.

6.2.4. FRNUM—FRAME NUMBER REGISTER (IO)

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

Bits [10:0] of this register contain the current frame number which 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. This register is updated at the end of each frame time.

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 is set (USBCMD register) 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].

6.2.5. FLBASEADD—FRAME LIST BASE ADDRESS REGISTER (IO)

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

This 32-bit register contains the beginning address of the Frame List in the system memory. HCD 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 0 (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 least significant bits are always 00. This requires DWord alignment for all list entries. This configuration supports 1,024 Frame List entries.

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

6.2.6. SOFMOD—START OF FRAME (SOF) MODIFY REGISTER (IO)

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

This 1-byte register is used to modify the value used in the generation of SOF timing on the USB. Only the seven least significant bits are used. When a new value is written into 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 clock source 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. Its 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 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 cycle time (number of SOF counter clock periods to generate a SOF frame length) is equal to 11,936 + value in this field. The default value is decimal 64 which gives a SOF cycle time of 12,000. 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 period.

Frame Length (# 12-MHz Clocks) (decimal)	SOF Reg. Value (decimal)
11,936	0
11,937	1
.	.
.	.
11,999	63
12,000	64
12,001	65
.	.
.	.
12,062	126
12,063	127

6.2.7. PORTSC—PORT STATUS AND CONTROL REGISTER (IO)

I/O Address: Base + (10–11h)—Port 0
Base + (12–13h)—Port 1
Default: 0080h
Access: Read/Write (WORD writeable only)

After a Power-up reset, Global reset, or Host Controller 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 make take up to 64 USB bit times for the port transition to occur. 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 global resume. Refer to Chapter 11 of the USB Specification for details on hub operation.

Bit	Description
15:13	**Reserved.** Must be written as 0s when writing this register.
12	**Suspend—R/W.** 1=Port in suspend state. 0=Port not in suspend 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 propagation 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 is sensitive to resume detection. 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	**Over-current Indicator Change—R/WC.** 1=A change from 1 to 0 has been detected on the Over-current (OC[X]#) pin for this port. 0=No change has been detected. Software sets this bit to 0 by writing a 1 to it.
10	**Over-current Indicator—RO.** 1=Overcurrent pin (OC[X]#) for this port is at logic 0 indicating over-current condition. 0=Overcurrent pin for this port is at logic 1 indicating a normal condition. If asserted, the corresponding port is disabled.
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—R/W.** 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 signaling. 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 will remain a 1 until the EOP has completed.

Bit	Description
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 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, overcurrent condition, 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 or enabling a port if there is a transaction currently in progress on the USB.
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 Connect Status Change bit (Bit 1) to be set.

7.0. POWER MANAGEMENT REGISTER DESCRIPTIONS

This section describes in detail the registers associated with the PIIX4 Power Management function. This includes device monitoring, suspend and resume functionality, clock control, and SMBus operation.

7.1. Power Management PCI Configuration Registers (PCI Function 3)

7.1.1. VID—VENDOR IDENTIFICATION REGISTER (FUNCTION 3)

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.

7.1.2. DID—DEVICE IDENTIFICATION REGISTER (FUNCTION 3)

Address Offset: 02–03h
Default Value: 7113h
Attribute: Read Only

The DID Register contains the device identification number. This register, along with the VID Register, define the PIIX4 Power Management Controller. Writes to this register have no effect.

Bit	Description
15:0	**Device Identification Number.** This is a 16-bit value assigned to the PIIX4 Power Management Controller.

7.1.3. PCICMD—PCI COMMAND REGISTER (FUNCTION 3)

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 (Not Implemented).** This bit is hardwired to 0.
8:5	**Reserved.** Read as 0.
4	**Memory Write and Invalidate Enable (Not Implemented).** This bit is hardwired to 0.

Bit	Description
3	**Special Cycle Enable (Not Implemented).** This bit is hardwired to 0.
2	**Bus Master Enable (Not Implemented).** This bit is hardwired to 0.
1	**Memory Space Enable (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 SMBus I/O space registers whose base address is described in the SMBus Base Address register. If this bit is set, access to the SMBus IO registers is enabled. The base register for the I/O registers must be programmed before this bit is set. When disabled, all IO accesses associated with SMBus Base Address are disabled. This bit functions independent of the state of Function 3 Power Management IO Space Enable (PMIOSE) bit (PMREGMISC register, bit 0).

7.1.4. PCISTS—PCI DEVICE STATUS REGISTER (FUNCTION 3)

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

DSR is a 16-bit status register that reports the occurrence of a PCI target-abort when the Power Management function is a target device. The register also indicates the Power Management DEVSEL# signal timing that is hardwired in the module.

Bit	Description
15	**Detected Parity (Not Implemented).** This bit is hardwired to 0.
14	**SERR# Status (Not Implemented).** This bit is hardwired to 0.
13	**Master-Abort Status (Not Implemented).** This bit is hardwired to 0.
12	**Received Target-Abort Status (Not Implemented).** This bit is hardwired to 0.
11	**Signaled Target-Abort Status (STA)—R/WC.** This bit is set when the Power Management function is targeted with a transaction that it 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 PIIX4's DEVSEL# timing when performing a positive decode. Since PIIX4 always generate the DEVSEL# with medium timing, DEVT=01. This DEVSEL# timing does not include Configuration cycles.
8	**Data Parity Detected (Not Implemented).** This bit is hardwired to 0.
7	**Fast Back to Back Capable (FBC)—RO.** Hardwired to 1. This bit indicates to the PCI Master that Power Management as a target is capable of accepting fast back-to-back transactions.
6:0	**Reserved**. Read as 0's.

7.1.5. RID—REVISION IDENTIFICATION REGISTER (FUNCTION 3)

Address Offset: 08h
Default Value: Initial Stepping=00h. Refer to PIIX4 Specification Updates for other values programmed here.
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.

7.1.6. CLASSC—CLASS CODE REGISTER (FUNCTION 3)

Address Offset: 09–0Bh
Default Value: 068000h
Attribute: Read Only

This register identifies the Base Class Code, Sub Class Code, and Device Programming interface for PIIX4 PCI function 3.

Bit	Description
23:16	**Base Class Code (BASEC).** 06h=Bridge device.
15:8	**Sub Class Code (SCC)**. 80h=Other Bridge device.
7:0	**Programming Interface (PI).** 00h=No specific register level programming defined.

7.1.7. HEDT—HEADER TYPE REGISTER (FUNCTION 3)

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

This register identifies the Power Management module as a single function device.

Bit	Description
7:0	**Device Type (DEVICET).** Multi-function device capability for PIIX4 is defined by the HEDT register in Function 0.

7.1.8. INTLN—INTERRUPT LINE REGISTER (FUNCTION 3)

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

Software programs this register with interrupt information concerning the Power Management module.

Bit	Description
7:0	**Interrupt Line.** The value in this register has no affect on PIIX4 hardware operations.

7.1.9. INTPN—INTERRUPT PIN (FUNCTION 3)

Address Offset: 3Dh
Default Value: 01h
Attribute: Read only

This register indicates that PCI interrupt pin PIRQA# is used for the Power Management module.

Bit	Description
7:3	**Reserved.**
2:0	**Serial Bus Module Interrupt Routing.** This field is hardwired to 01h to indicate that PCI interrupt pin PIRQA# is used.

7.1.10. PMBA—POWER MANAGEMENT BASE ADDRESS (FUNCTION 3)

Address Offset: 40–43h
Default Value: 00000001h
Attribute: Read/Write

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

Bit	Description
31:16	**Reserved.** Hardwired to 0s. Must be written as 0s.
15:6	**Index Register Base Address.** Bits [15:6] correspond to I/O address signals AD [15:6], respectively.
5:1	**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.

7.1.11. CNTA—COUNT A (FUNCTION 3)

Address Offset: 44–47h
Default Value: 00h
Attribute: Read/Write

This register contains the initial counts of the idle timers for devices 0–11, the selection bits for the timer granularity of the timers for devices 0, 1, 2 and 3. In addition, it contains the count for the slow burst timer.

Bit	Description
31:28	**Slow Burst Count (SB_CNT)—R/W.** Specifies the initial and reload value of the slow burst timer.
27:23	**Idle Timer Count D (IDL_CNTD)—R/W.** Specifies the initial and reload count of the device 11 (user interface) idle timer.
22	**Device 11 Idle Timer Resolution (IDL_SEL_DEV11)—R/W.** Selects the clock resolution of the device 11 (user interface) idle timer. 0=1 second granular. 1=1 minute granular.
21:17	**Idle Timer Count C (IDL_CNTC)—R/W.** Specifies the initial and reload count of the device 9–10 (generic range) idle timers.
16:12	**Idle Timer Count B (IDL_CNTB)—R/W.** Specifies the initial and reload count of the device 4–7 (audio, floppy, serial ports, parallel port) idle timers.
11:8	**SW Idle Timer Count (SW_CNT)—R/W.** Specifies the initial and reload count of the device 3 (secondary IDE drive 1, software SMI) idle timer.
7	**Device 3 Idle Timer Resolution (IDL_SEL_DEV3)—R/W.** Selects the clock source for the device 3 (secondary IDE drive 1, software SMI) idle timer. 0=8 second granular. 1=1 ms granular.
6	**Device 2 Idle Timer Resolution (IDL_SEL_DEV2)—R/W.** Selects the clock source for the device 2 (secondary IDE drive 0) idle timer. 0=8 second granular. 1=1 second granular.
5	**Device 1 Idle Timer Resolution (IDL_SEL_DEV1)—R/W.** Selects the clock source for the device 1 (primary IDE drive 1) idle timer. 0=8 second granular. 1=1 second granular.
4	**Device 0 Idle Timer Resolution (IDL_SEL_DEV0)—R/W.** Selects the clock source for the device 0 (primary IDE drive 0) idle timer. 0=8 second granular. 1=1 second granular.
3:0	**Idle Timer Count A (IDL_CNTA)—R/W.** Specifies the initial and reload count of the device 0–2 (primary IDE drives 0 and 1, secondary IDE drive 0) idle timers.

7.1.12. CNTB—COUNT B (FUNCTION 3)

Address Offset: 48–4Bh
Default Value: 00h
Attribute: Read/Write

This register contains the counts for Fast Burst Timer, the CPU select and lock bits, the thermal duty cycle programming bits, the ZZ enable bits, the clock granularity selection for device 8, and the Video Status bit.

Bit	Description
31:25	**Reserved.** Read as 0.
24	**Video Status (VID_STS)—R/WC.** 1=The PCI bus utilization monitor has detected PCI activity which exceeds its defined threshold (see description for Device Monitor 11). This bit is set by hardware and reset by writing a 1 to this bit position.
23	**Reserved.** Read as 0.

Bit	Description
22:18	**Bus Master Timer Count (BM_CNT)—R/W.** Specifies the initial and reload count of the device 8 (parallel port and PCI bus master) idle timer.
17:16	**Reserved.** Read as 0.
15	**Device 8 Idle Timer Resolution (IDL_SEL_DEV8)—R/W.** Selects the clock source for the device 8 (parallel port) idle timer. 0=1 second granular. 1=1 ms granular.
14	**ZZ Enable (ZZ_EN)—R/W.** 1=Enable PIIX4 assertion of the ZZ signal. 0=Disable. When enabled, PIIX4 will assert ZZ signal under certain conditions when entering clock control mode. Whether or not ZZ is asserted depends on: 1. Time from STPCLK# assertion to Stop Grant cycle. 2. Frequency of any enabled Stop Break or Burst events. 3. Programmed throttle duty cycle if throttling enabled. **NOTE** The L2 cache can not be snooped with ZZ signal asserted; therefore, must be disabled in Level 2 power state such as Stop Grant.
13:11	**Thermal Duty Cycle (THRM_DTY)—R/W.** This 3-bit field determines the duty cycle for the clock control thermal throttling mode (THRM# is asserted). The duty cycle indicates the percentage of time the STPCLK# signal is asserted while in the thermal throttle mode. The field is decoded as follows: **Bits[13:11] Mode — Bits[13:11] Mode** 000 Reserved — 100 50% 001 12.5% — 101 62.5% 010 25% — 110 75% 011 37.5% — 111 87.5% **NOTE** Software must set this 3-bit field to a non-zero value for proper operation.
10:6	**Processor PLL Lock Count (CPU_LCK)—R/W.** Specifies the initial count of fast burst timer when used to measure the processor PLL lock time. The fast burst timer is loaded with the CPU_LCK value and the appropriate clock source selected when the processor transitions from the stop clock or deep sleep state.
5	**Processor PLL Lock Resolution (CPU_SEL)—R/W.** Selects the clock resolution used for the fast burst timer when it is used to count the processor's PLL lock time. 0=1 ms granular. 1=1 μs granular.
4:0	**Fast Burst Timer Count (FB_CNT)—R/W.** Specifies the initial and reload count of fast burst timer.

7.1.13. GPICTL—GENERAL PURPOSE INPUT CONTROL (FUNCTION 3)

Address Offset: 4C–4Fh
Default Value: 00h
Attribute: Read/Write

This register contains the enable bits, the polarity bits and edge selection bits for the General Purpose IO in device monitors 1–13.

Bit	Description
31:28	**Reserved.**
27	**GPI Edge Select (GPI_EDG_DEV13)—R/W.** Selects edge or level sensitivity of device monitor 13 GPI signal. 0=level. 1=edge.
26	**GPI Edge Select (GPI_EDG_DEV12)—R/W.** Selects edge or level sensitivity of device monitor 12 GP signal. 0=level. 1=edge.
25:13	**GPI Polarity Select (GPI_POL_DEV[1:13])—R/W.** Selects the assertion polarity for an enabled GPI signal for device monitors 1–13. 0=Asserted high. 1=Asserted low. Bit 25 corresponds to device monitor 13 and bit 13 corresponds to device monitor 1.
12:0	**GPI Enable (GPI_EN_DEV[1:13])—R/W.** 1=Enable the device monitor's GPI signal into the trap and idle decode logic for devices [13:1]. 0=Disable. Bit 12 corresponds to device monitor 13 and bit 0 corresponds to device monitor 1.

Table 17 illustrates which GPI signals correspond with which device.

Table 17. GPI to Device Monitor Translation

Device Monitoring	Optional GPI Signal	Device Monitoring	Optional GPI Signal
DEV0	None	DEV7	GPI16
DEV1	GPI5	DEV8	GPI17
DEV2	GPI6	DEV9	GPI4
DEV3	GPI0	DEV10	GPI18
DEV4	GPI13	DEV11	GPI19
DEV5	GPI14	DEV12	GPI20
DEV6	GPI15	DEV13	GPI21

7.1.14. DEVRESD—DEVICE RESOURCE D (FUNCTION 3)

Address Offset: 50–52h
Default Value: 00h
Attribute: Read/Write

This register contains the event enable bits for DMA channels 0,1,3,5,6,7. It also contains the floppy disk controller monitor enable bit, serial port monitor enable bits, Device 11 IRQ1 monitor enable bit, Device 11 IRQ12 monitor enable bit and LPT DMA select bits.

Bit	Description
23	**Reserved.**
22:21	**LPT DMA select (LPT_DMA_SEL)—R/W.** Selects the active DACK signal used to reload the idle timer for device 8 (parallel port). Enabled by RES_EN_DEV8 bit. **Bits[22:21] DACK Signal** 00 DACK0 01 DACK1 10 DACK3 11 Reserved
20	**Device 11 IRQ 12 Enable (IRQ12_EN_DEV11)—R/W.** 1=Enable an asserted IRQ12/M signal (mouse activity) to generate a device 11 (user interface) decode event. 0=Disable.
19	**Device 11 IRQ1 Enable (IRQ1_EN_DEV11)—R/W.** 1=Enable an asserted IRQ1 signal (keyboard activity) to generate a device 11 (user interface) decode event. 0=Disable.
18	**LPT Port Enable (LPT_MON_EN)—R/W.** 1=Enable accesses to parallel port address range (LPT_DEC_SEL) to generate a device 8 (parallel port) decode event. 0=Disable.
17	**LPT DMA Monitor Enable (RES_EN_DEV8)—R/W.** 1=Enable the selected DACKs (LPT_DMA_SEL) to generate a device 8 (parallel port) decode event. 0=Disable.
16	**Serial Port B Monitor Enable (SB_MON_EN)—R/W.** 1=Enable accesses to serial port address range (COMB_DEC_SEL) to generate a device 7 (serial port B) decode event. 0=Disable.
15	**Reserved.**
14	**Serial Port A Monitor Enable (SA_MON_EN)—R/W.** 1=Enable accesses to serial port address range (COMA_DEC_SEL) to generate a device 6 (serial port A) decode event. 0=Disable.
13	**Reserved.**
12	**Floppy Disk Controller Monitor Enable (FDC_MON_EN)—R/W.** 1=Enable accesses to floppy disk controller address range (FDC_DEC_SEL) to generate a device 5 (floppy controller) decode event. 0=Disable.
11	**FDC DMA Monitor Enable (RES_EN_DEV5)—R/W.** 1=Enable DACK2# to generate a device 5 (floppy controller) reload event. 0=Disable.
10:6	**Reserved**.
5	**DACK7 Enable (DACK7_EN_DEV4)—R/W.** 1=Enable DACK7# to generate a device 4 (audio controller) reload event. 0=Disable.
4	**DACK6 Enable (DACK6_EN_DEV4)—R/W.** 1=Enable DACK6# to generate a device 4 (audio controller) reload event. 0=Disable.
3	**DACK5 Enable (DACK5_EN_DEV4)—R/W.** 1=Enable DACK5# to generate a device 4 (audio controller) reload event. 0=Disable.
2	**DACK3 Enable (DACK3_EN_DEV4)—R/W.** 1=Enable DACK4# to generate a device 4 (audio controller) reload event. 0=Disable.
1	**DACK1 Enable (DACK1_EN_DEV4)—R/W.** 1=Enable DACK1# to generate a device 4 (audio controller) reload event. 0=Disable.
0	**DACK0 Enable (DACK0_EN_DEV4)—R/W.** 1=Enable DACK0# to generate a device 4 (audio controller) reload event. 0=Disable.

7.1.15. DEVACTA—DEVICE ACTIVITY A (FUNCTION 3)

Address Offset: 54–57h
Default Value: 00h
Attribute: Read/Write

This register contains bits that enable Device Activity as Global Timer Reload events or Clock Events (Burst or Break).

Bit	Description
31	**Device 5 Reload Select (BRLD_SEL_DEV5)—R/W.** Selects which burst timer is reloaded upon an enabled device monitor 5 idle event. 0=reload the slow burst timer. 1=reload the fast burst timer.
30	**Device 3 Reload Select (BRLD_SEL_DEV3)—R/W.** Selects which burst timer is reloaded upon an enabled device monitor 3 idle event. 0=reload the slow burst timer. 1=reload the fast burst timer.
29	**Device 2 Reload Select (BRLD_SEL_DEV2)—R/W.** Selects which burst timer is reloaded upon an enabled device monitor 2 idle event. 0=reload the slow burst timer. 1=reload the fast burst timer.
28	**Device 1 Reload Select (BRLD_SEL_DEV1)—R/W.** Selects which burst timer is reloaded upon an enabled device monitor 1 idle event. 0=reload the slow burst timer. 1=reload the fast burst timer.
27:14	**Burst Timer Reload Enable (BRLD_EN_DEV[0–13])—R/W.** 1=Enable reload events from the respective device monitor to reload the enabled burst timer or generate a Stop Break event. 0=Disable. Bit 27 corresponds to device monitor 13 and bit 14 corresponds to device monitor 0.
13:0	**Global Timer Reload Enable Bits (GRLD_EN_DEV[0–13])—R/W.** 1=Enable reload events from the respective device monitor to reload the global standby timer. 0=Disable. Bit 13 corresponds to device monitor 13 and bit 0 corresponds to device monitor 0.

7.1.16. DEVACTB—DEVICE ACTIVITY B (FUNCTION 3)

Address Offset: 58–5Bh
Default Value: 00h
Attribute: Read/Write

This register contains the Clock Event and Global Timer Reload enables for IRQs, PCI access, PME events, Video.

Bit	Description
31:25	**Reserved.**
25	**APMC Enable (APMC_EN)—R/W.** 1=Enable generation of SMI# when APMC register is read and SMI# is enabled. 0=Disable.
24	**Video Enable (VIDEO_EN)—R/W.** 1=Enable the video detect (PCI Bus utilization) logic to generate a timer reload event for device monitor 11. 0=Disable. This logic detects PCI bus utilization as set by the two fields: BUS_UTIL, and %BUS_UTIL.
23:16	**Percentage Bus Utilization Threshold (%BUS_UTIL)—R/W.** This field controls the percentage of time that the minimum bus utilization threshold (represented by the BUS_UTIL field) must be maintained in order to generate a video event. The actual count is measured by the number of time slices that exceed BUS_UTIL within a 256 time slice window.
15:8	**Bus Utilization Threshold (BUS_UTIL)—R/W.** This field controls the threshold for bus utilization detection. If the video detect logic finds more PCI data phases than specified by BUS_UTIL within a 256 clock period (time slice), then that time slice is counted.
7	**Reserved.**
6	**IRQ Global Reload Enable (GRLD_EN_IRQ)—R/W.** 1=Enable an unmasked IRQ[1,3:7,9:15], NMI, or INIT to, when asserted, reload the Global Standby Timer. 0=Disable.
5	**IRQ8# Clock Event Enable (BRLD_EN_IRQ8)—R/W.** 1=Enable an unmasked IRQ8# to, when asserted, generate a Fast Burst Timer reload or Stop Break event. 0=Disable.
4	**PME Clock Event Enable (BRLD_EN_PME)—R/W.** 1=Enable an asserted SMI#, GPI1#, PWRBTN#, or LID signal to generate a Fast Burst Timer reload or Stop Break event. 0=Disable.
3	**Undefined.** Must be written as a 0.
2	**Keyboard/Mouse Global Reload Enable (GRLD_EN_KBC_MS)—R/W.** 1=Enable an assertion of IRQ1 or IRQ12/M to reload the Global Standby Timer. 0=Disable.
1	**IRQ Clock Event Enable (BRLD_EN_IRQ)—R/W.** 1=Enable an unmasked IRQ[1,3:7,9:15], NMI, or INIT to generate a Burst event or Stop Break event. 0=Disable.
0	**IRQ0 Clock Event Enable (BRLD_EN_IRQ0)—R/W.** 1=Enable an unmasked IRQ0 to generate a Burst event or Stop Break event. 0=Disable.

7.1.17. DEVRESA—DEVICE RESOURCE A (FUNCTION 3)

Address Offset: 5C–5Fh
Default Value: 00h
Attribute: Read/Write

Bit	Description
31	**Device 8 EIO Enable (EIO_EN_DEV8)—R/W.** 1=Enable PCI access to the device 8 enabled I/O ranges to be claimed by PIIX4 and forwarded to the ISA/EIO bus. 0=Disable. The LPT_MON_EN must be set to enable the decode.
30	**Device 13 EIO Enable (EIO_EN_DEV13)—R/W.** 1=Enable PCI accesses to the device 13 enabled memory and I/O ranges to be claimed by PIIX4 and forwarded to the ISA/EIO bus. 0=Disable. The MEM_EN_DEV13 or IO_EN_DEV13 must be set to enable the memory or IO decodes respectively.
29	**Device 12 EIO Enable (EIO_EN_DEV12)—R/W.** 1=Enable PCI accesses to the device 12 enabled memory and I/O ranges to be claimed by PIIX4 and forwarded to the ISA/EIO bus. 0=Disable. The MEM_EN_DEV12 or IO_EN_DEV12 must be set to enable the memory or IO decodes respectively.
28	**Device 11 Keyboard Enable (KBC_EN_DEV11)—R/W.** 1=Enable PCI bus decode for accesses to keyboard controller I/O ports (60h and 64h). 0=Disable. The EIO enable bit, idle enable bit, or trap enable bit for this device must also be set in order to enable these respective functions.
27	**Graphics A/B Segment Memory Enable (GRAPH_AB _EN)—R/W.** 1=Enable PCI bus decode for accesses to the PC compatible frame buffer ranges (A and B segments). 0=Disable. PIIX4 does not positive decode these accesses for forwarding to the ISA bus.
26	**Graphics I/O Enable (GRAPH_IO_EN)—R/W.** 1=Enable PCI bus decode for accesses to the VGA I/O addresses (3B0h–3DFh). 0=Disable. PIIX4 does not positive decode these accesses for forwarding to the ISA bus.
25	**SoundBlaster EIO Enable (SB_EIO_EN)—R/W.** 1=Enable PCI bus decode for accesses to the SoundBlaster device enabled decode ranges (bits[3,5:6]) to be claimed by PIIX4 and forwarded to the ISA/EIO bus. The SB_EN bit must be set to enable their respective ranges. 0=Disable.
24	**Linear Frame Buffer Decode Enable (LFB_DEC_EN)—R/W.** 1=Enable PCI bus decode for accesses to the generic memory range for linear frame buffer. 0=Disable. The linear frame buffer address range is defined by the linear frame buffer base address and mask bits (bits[23:10]). PIIX4 does not positive decode these accesses for forwarding to the ISA bus.
23:22	**Linear Frame Buffer Mask (LFB_MASK_DEV11)—R/W.** This field defines a 2-bit mask for the linear frame buffer address, corresponding to AD[21:20]. A ‘1’ in a bit position indicates that the corresponding address bit is masked (i.e. ignored) when performing the decode. This field defines the size of the linear frame buffer window. Note that programming these bits to ‘10’ results in a split address range.
21:10	**Linear Frame Buffer Base Address (LFB_BASE_DEV11)—R/W.** This field defines the 12-bit memory base address range, corresponding to AD[31:20] for the linear frame buffer address. This field in conjunction with the LFB_MASK_DEV11 field defines a 1–8-Mbyte linear frame buffer that can be enabled for monitoring using the device monitoring system 11.

Bit	Description
9:8	**Microsoft* Sound System Decode Select (MSS_SEL)—R/W.** Selects the Microsoft Sound System decode range enabled with bit 7. This field is decoded as follows: **Bits[9:8] MSS Decode** 00 530h–537h 01 604h–60Bh 10 E80h–E87h 11 F40h–F47h
7	**Microsoft Sound System Decode Enable (MSS_EN)—R/W.** 1=Enable PCI bus decode for accesses to the I/O address range selected by the MSS_SEL field. 0=Disable. The EIO enable bit, idle enable bit, or trap enable bit for device 4 must also be set in order to enable those respective functions.
6:5	**Sound Blaster Decode Select (SB_SEL)—R/W.** Selects the Sound Blaster decode range enabled with bit 3. This field is decoded as follows: **Bits[6:5] Sound Blaster Decode** 00 220–22Fh, 230–233h 01 240–24Fh, 250–253h 10 260–26Fh, 270–273h 11 280–28Fh, 290–293h
4	**Game Port Enable (GAME_EN).** 1=Enable PCI bus decode for accesses to the Game port I/O address range (200–207h). 0=Disable. The Game Port EIO enable bit, or Device 4 idle enable bit or trap enable must also be set to enable these respective functions.
3	**Sound Blaster 8/16-bit Decode Enable (SB_EN)—R/W.** 1=Enable PCI bus decode for accesses to the I/O address range selected by the SB_SEL field and to the ADLIB (388–38Bh) address ranges. 0=Disable. The SoundBlaster EIO enable bit, or Device 4 idle enable bit, or trap enable bit must also be set to enable these respective functions.
2:1	**MIDI Decode Select (MIDI_SEL)—R/W.** Selects the MIDI decode range enabled with bit 1. This field is decoded as follows: **Bits[2:1] MIDI Decode** 00 300–303h 01 310–313h 10 320–323h 11 330–333h
0	**MIDI Enable (MIDI_EN)—R/W.** 1=Enable PCI bus decode for accesses to the I/O address range selected by the MIDI_SEL field. 0=Disable. The EIO enable bit, idle enable bit, or trap enable bit for device 4 must also be set in order to enable these respective functions.

7.1.18. DEVRESB—DEVICE RESOURCE B (FUNCTION 3)

Address Offset: 60–63h
Default Value: 00h
Attribute: Read/Write

Bit	Description
31	**Game Port EIO Enable (GAME_EIO_EN)—R/W.** 1=Enable PCI bus decode for accesses to the Game Port enabled decode ranges to be claimed by PIIX4 and forwarded to the ISA/EIO bus. 0=Disable. The GAME_EN bit must be set to enable this range.
30	**Keyboard EIO Enable (KBC_EIO_EN)—R/W.** 1=Enable PCI access to the keyboard controller enabled I/O ranges (60h and 64h) to be claimed by PIIX4 and forwarded to the ISA/EIO bus. 0=Disable. The KBC_EN_DEV11 bit must be set to enable the decode.
29	**Device 5 EIO Enable (EIO_EN_DEV5)—R/W.** 1=Enable PCI access to the floppy disk controller enabled I/O ranges selected by FDC_DEC_SEL field to be claimed by PIIX4 and forwarded to the ISA/EIO bus. 0=Disable. The FDC_MON_EN bit must be set to enable the decode.
28	**Floppy Disk Controller Decode Select (FDC_DEC_SEL)—R/W.** 1=Secondary FDC Address (370h–375h, 377h). 0=Primary FDC Address (3F0h–3F5h, 3F7h). This field selects the floppy disk controller I/O range enabled with bit 3.
27	**Reserved.**
26:25	**LPT Controller Decode Select (LPT_DEC_SEL)—R/W.** Selects the parallel port (device 8) I/O range enabled with the LPT_MON_EN bit. This field is decoded as follows: **Bits[26:25] LPT Decode** 00 3BCh–3BFh, 7BCh–7BEh 01 378h–37Fh, 778h–77Ah 10 278h–27Fh, 678h–67Ah 11 Reserved
24	**Microsoft Sound System EIO Enable (MSS_EIO_EN)—R/W.** 1=Enable PCI bus decode for accesses to the Microsoft Sound System enabled decode ranges (DEVRESA: Bits[7:9]) to be claimed by PIIX4 and forwarded to the ISA/EIO bus. 0=Disable. The MSS_EN bit must be set to enable this range.
23	**Device 9 Generic Decode Chip-select (CS_EN_DEV9)—R/W.** 1=Enable assertion of the chip-select signal PCS0# for all accesses within the device 9 I/O decode range. 0=Disable. The EIO_EN_DEV9 bit must also be set to enable this function.
22	**Device 9 EIO Enable (EIO_EN_DEV9)—R/W.** 1=Enable PCI access to the device 9 enabled I/O range or embedded controller IO range to be claimed by PIIX4 and forwarded to the ISA/EIO bus. 0=Disable. The GDEC_MON_DEV9 bit or EC_EN_DEV9 bit must be set to enable the decode.
21	**Device 9 Generic Decode Monitor Enable (GDEC_MON_DEV9)—R/W.** 1=Enable PCI bus decode for accesses to the I/O address range selected by the BASE_DEV9 and MASK_DEV9 fields. 0=Disable. The EIO enable bit, idle enable bit, or trap enable bit for device 4 must also be set in order to enable these respective functions.
20	**Midi EIO Enable (MIDI_EIO_EN)—R/W.** 1=Enable PCI bus decode for accesses to the Midi enabled decode ranges (DEVRESA: Bits[0:2]) to be claimed by PIIX4 and forwarded to the ISA/EIO bus. 0=Disable. The MIDI_EN bit must be set to enable this range.
19:16	**Device 9 Generic Decode Mask (MASK_DEV9)—R/W.** Specifies the 4-bit I/O base address mask used to determine the IO address range size for device 9 accesses. MASK_DEV9 (bits[19:16]) correspond to AD[3:0]. A ‘1’ in a bit position indicates that the corresponding address bit is masked (i.e. ignored) when performing the decode. Note that programming these bits to certain patterns (such as ‘1001’) results in a split address range.

Bit	Description
15:0	**Device 9 Generic Decode Base Address (BASE_DEV9)—R/W.** Specifies the 16-bit I/O base address range (AD[15:0]) for the device 9 I/O range. When this field is combined with MASK_DEV9 field, an I/O range is defined starting from the base address register value to the size defined by the mask register.

7.1.19. DEVRESC—DEVICE RESOURCE C (FUNCTION 3)

Address Offset: 64–67h
Default Value: 00h
Attribute: Read/Write

Bit	Description
31	**Device 7 EIO Enable (EIO_EN_DEV7)—R/W.** 1=Enable PCI access to the device 7 (serial port B) enabled I/O ranges selected by COMB_DEC_SEL field to be claimed by PIIX4 and forwarded to the ISA/EIO bus. 0=Disable. The SB_MON_EN bit must be set to enable the decode.
30:28	**Serial Port B Decode Select (COMB_DEC_SEL)—R/W.** Selects the I/O range that the Serial Port B (Device 7) decode responds to. This field is decoded as follows: **Bits[30:28] Serial B Decode** — **Bits[30:28] Serial B Decode** 000 3F8h–3FFh (COM1) — 100 238h–23Fh 001 2F8h–2FFh (COM2) — 101 2E8h–2EFh (COM4) 010 220h–227h — 110 338h–33Fh 011 228h–22Fh — 111 3E8h–3EFh (COM3)
27	**Device 6 EIO Enable (EIO_EN_DEV6)—R/W.** 1=Enable PCI access to the device 6 (serial port A) enabled I/O ranges selected by COMA_DEC_SEL field to be claimed by PIIX4 and forwarded to the ISA/EIO bus. 0=Disable. The SA_MON_EN bit must be set to enable the decode.
26:24	**Serial Port A Decode Select (COMA_DEC_SEL)—R/W.** Selects the I/O range that the Serial Port A (Device 6) decode responds to. This field is decoded as follows: **Bits[26:24] Serial A Decode** — **Bits[26:24] Serial A Decode** 000 3F8h–3FFh (COM1) — 100 238h–23Fh 001 2F8h–2FFh (COM2) — 101 2E8h–2EFh (COM4) 010 220h–227h — 110 338h–33Fh 011 228h–22Fh — 111 3E8h–3EFh (COM3)
23	**Device 10 Generic Decode Chip-select (CS_EN_DEV10)—R/W.** 1=Enable assertion of the chip-select signal PCS1# for all accesses within the device 10 I/O decode range. 0=Disable. The EIO_EN_DEV10 bit must also be set to enable this function.
22	**Device 10 EIO Enable (EIO_EN_DEV10)—R/W.** 1=Enable PCI access to the device 10 enabled I/O range to be claimed by PIIX4 and forwarded to the ISA/EIO bus. 0=Disable. The GDEC_MON_DEV10 bit must be set to enable the decode.
21	**Device 10 Generic Decode Monitor Enable (GDEC_MON_DEV10)—R/W.** 1=Enable PCI bus decode for accesses to the I/O address range selected by the BASE_DEV10 and MASK_DEV10 fields. 0=Disable. The EIO enable bit, idle enable bit, or trap enable bit for device 4 must also be set in order to enable these respective functions.
20	**Reserved.**

Bit	Description
19:16	**Device 10 Generic Decode Mask (MASK_DEV10)—R/W.** Specifies the 4-bit I/O base address mask used to determine the IO address range size for device 10 accesses. MASK_DEV10 (bits[19:16]) correspond to AD[3:0]. A '1' in a bit position indicates that the corresponding address bit is masked (i.e. ignored) when performing the decode. Note that programming these bits to certain patterns (such as '1001') results in a split address range.
15:0	**Device 10 Generic Decode Base Address (BASE_DEV10)—R/W.** Specifies the 16-bit I/O base address range (AD[15:0]) for the device 10 I/O range. When this field is combined with MASK_DEV10 field, an I/O range is defined starting from the base address register value to the size defined by the mask register.

7.1.20. DEVRESE—DEVICE RESOURCE E (FUNCTION 3)

Address Offset: 68–6Ah
Default Value: 00h
Attribute: Read/Write

Bit	Description
23:21	**Reserved.**
20	**Device 12 I/O Monitor Enable (IO_EN_DEV12)—R/W.** 1=Enable PCI bus decode for accesses to the I/O address range selected by the IBASE_DEV12 and IMASK_DEV12 fields. 0=Disable. The EIO enable bit, or trap enable bit for device 12 must also be set in order to enable these respective functions.
19:16	**Device 12 I/O Decode Mask (IMASK_DEV12)—R/W.** Specifies the 4-bit I/O base address mask used to determine the IO address range size for device 12 accesses. IMASK_DEV12 (bits[19:16]) correspond to AD[3:0]. A '1' in a bit position indicates that the corresponding address bit is masked (i.e. ignored) when performing the decode. Note that programming these bits to certain patterns (such as '1001') results in a split address range.
15:0	**Device 12 I/O Decode Base Address (IBASE_DEV12)—R/W.** Specifies the 16-bit I/O base address range (AD[15:0]) for the device 12 I/O range. When this field is combined with IMASK_DEV12 field, an I/O range is defined starting from the base address register value to the size defined by the mask register.

7.1.21. DEVRESF—DEVICE RESOURCE F (FUNCTION 3)

Address Offset: 6C–6Fh
Default Value: 00h
Attribute: Read/Write

Bit	Description
31:15	**Device 12 Memory Decode Base Address (MBASE_DEV12)—R/W.** Specifies the 17-bit memory base address range (AD[31:15]) for the device 12 memory range. When this field is combined with the MMASK_DEV12 field, a memory range is defined from the base address value to the size defined by the mask register.
14:8	**Reserved.**

Bit	Description
7	**Device 12 Memory Monitor Enable (MEM_EN_DEV12)—R/W.** 1=Enable PCI bus decode for accesses to the memory address range selected by the MBASE_DEV12 and MMASK_DEV12 fields. 0=Disable. The EIO enable bit, or trap enable bit for device 12 must also be set in order to enable these respective functions.
6:0	**Device 12 Memory Decode Mask (MMASK_DEV12)—R/W.** Specifies the 7-bit memory base address mask used to determine the memory address range size for device 12 accesses. MMASK_DEV12 (bits[6:0]) correspond to AD[21:15]. A '1' in a bit position indicates that the corresponding address bit is masked (i.e. ignored) when performing the decode. Note that programming these bits to certain patterns (such as '1110011') results in split address ranges.

7.1.22. DEVRESG—DEVICE RESOURCE G (FUNCTION 3)

Address Offset: 70–72h
Default Value: 00h
Attribute: Read/Write

Bit	Description
23:21	**Reserved.**
20	**Device 13 I/O Monitor Enable (IO_EN_DEV13)—R/W.** 1=Enable PCI bus decode for accesses to the I/O address range selected by the IBASE_DEV13 and IMASK_DEV13 fields. 0=Disable. The EIO enable bit or trap enable bit for device 13 must also be set in order to enable these respective functions.
19:16	**I/O Decode Mask (IMASK_DEV13)—R/W.** Specifies the 4-bit I/O base address mask used to determine the IO address range size for device 13 accesses. IMASK_DEV13 (bits[19:16]) correspond to AD[3:0]. A '1' in a bit position indicates that the corresponding address bit is masked (i.e. ignored) when performing the decode. Note that programming these bits to certain patterns (such as '1001') results in a split address range.
15:0	**I/O Decode Base Address (IBASE_DEV13)—R/W.** Specifies the 16-bit I/O base address range (AD[15:0]) for the device 13 I/O range. When this field is combined with IMASK_DEV13 field, an I/O range is defined starting from the base address register value to the size defined by the mask register.

7.1.23. DEVRESH—DEVICE RESOURCE H (FUNCTION 3)

Address Offset: 74–77h
Default Value: 00h
Attribute: Read/Write

Bit	Description
31:15	**Memory Decode Base Address (MBASE_DEV13)—R/W.** Specifies the 17-bit memory base address range (AD[31:15]) for the device 13 memory range. When this field is combined with the MMASK_DEV13 field, a memory range is defined from the base address value to the size defined by the mask register.
14:8	**Reserved.**
7	**Device 13 Memory Monitor Enable (MEM_EN_DEV13)—R/W.** 1=Enable PCI bus decode for accesses to the memory address range selected by the MBASE_DEV13 and MMASK_DEV13 fields. 0=Disable. The EIO enable bit or trap enable bit for device 13 must also be set in order to enable these respective functions.
6:0	**Memory Decode Mask (MMASK_DEV13)—R/W.** Specifies the 7-bit memory base address mask used to determine the memory address range size for device 13 accesses. MMASK_DEV13 (bits[6:0]) correspond to AD[21:15]. A ‘1’ in a bit position indicates that the corresponding address bit is masked (i.e. ignored) when performing the decode. Note that programming these bits to certain patterns (such as ‘1110011’) results in split address ranges.

7.1.24. DEVRESI—DEVICE RESOURCE I (FUNCTION 3)

Address Offset: 78–7Bh
Default Value: 00h
Attribute: Read/Write

Bit	Description
31:21	**Reserved.**
20	**Generic I/O Decode 0 Enable (IO_EN_GDEC0)—R/W.** 1=Enable accesses to the I/O address range selected by the IO_MASK_GDEC0 and IO_BASE_GDEC0 fields to be claimed by PIIX4 and forwarded to the ISA/EIO bus. 0=Disable.
19:16	**Generic Decode 0 I/O Mask (IO_MASK_GDEC0)—R/W.** This field specifies the 4-bit I/O base address mask used to determine the IO address range size. IO_MASK_GDEC0(bits[19:16]) correspond to AD[3:0]. A ‘1’ in a bit position indicates that the corresponding address bit is masked (i.e., ignored) when performing the decode. Note that programming these bits to certain patterns (such as 1001) results in a split address range.
15:0	**Generic Decode 0 I/O Base Address (IO_BASE_GDEC0)—R/W.** Specifies the 16-bit I/O base address range (AD[15:0]) for the generic decode range 0. When this field is combined with IO_MASK_GDEC0 field, an I/O range is defined starting from the base address register value to the size defined by the mask register.

7.1.25. DEVRESJ—DEVICE RESOURCE J (FUNCTION 3)

Address Offset: 7C–7Fh

Default Value: 00h
Attribute: Read/Write

Bit	Description
31:21	**Reserved.**
20	**Generic I/O Decode 1 Enable (IO_EN_GDEC1)—R/W.** 1=Enable accesses to the I/O address range selected by the IO_MASK_GDEC1 and IO_BASE_GDEC1 fields to be claimed by PIIX4 and forwarded to the ISA/EIO bus. 0=Disable.
19:16	**Generic Decode 1 I/O Mask (IO_MASK_GDEC1)—R/W.** This field specifies the 4-bit I/O base address mask used to determine the IO address range size. IO_MASK_GDEC1(bits[19:16]) correspond to AD[3:0]. A '1' in a bit position indicates that the corresponding address bit is masked (i.e., ignored) when performing the decode. Note that programming these bits to certain patterns (such as 1001) results in a split address range.
15:0	**Generic Decode 1 I/O Base Address (IO_BASE_GDEC1)—R/W.** This field specifies the 16-bit I/O base address range (AD[15:0]) for the generic decode range 1. When this field is combined with IO_MASK_GDEC1 field, an I/O range is defined starting from the base address register value to the size defined by the mask register.

7.1.26. PMREGMISC—MISCELLANEOUS POWER MANAGEMENT (FUNCTION 3)

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

This register contains miscellaneous functionality associated with the PIIX4 Power Management capabilities.

Bit	Description
7:1	**Reserved.**
0	**Power Management IO Space Enable (PMIOSE)—R/W.** 1=Enable. 0=Disable. This bit controls the access to the Power Management I/O space registers whose base address is described in the Power Management Base Address register. If this bit is set, access to the power management IO registers are enabled. The base address register for the I/O registers must be programmed before this bit is set. When disabled, all IO accesses associated with Power Management Base Address are disabled. This bit functions independent of the state of Function 3 IO Space Enable (IOSE) bit (PCICMD register, bit 0).

7.1.27. SMBBA—SMBUS BASE ADDRESS (FUNCTION 3)

Address Offset: 90–93h
Default Value: 00000001h
Attribute: Read/Write

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

Bit	Description
31:16	**Reserved.** Hardwired to 0s. Must be written as 0s.
15:4	**Index Register Base Address.** Bits [15:4] correspond to I/O address signals AD [15:4], respectively.
3:1	**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.

7.1.28. SMBHSTCFG—SMBUS HOST CONFIGURATION (FUNCTION 3)

Address Offset: D2h
Default Value: 00h
Attribute: Read/Write

Bit	Description
7:4	**Reserved.**
3:1	**SMBus Interrupt Select (SMB_INTRSEL)—R/W.** Selects the type of interrupt generated by the SMBus controller. This field is decoded as follows: **Bits[3:1] — SMBus Interrupt:** 000 SMI#; 001 Reserved; 010 Reserved; 011 Reserved; 100 IRQ9; 101 Reserved; 110 Reserved; 111 Reserved
0	**SMBus Controller Host Interface Enable (SMB_HST_EN)—R/W.** 1=Enable the SMBus Controller Host Interface. 0=Disable.

7.1.29. SMBSLVC—SMBUS SLAVE COMMAND (FUNCTION 3)

Address Offset: D3h
Default Value: 00h
Attribute: Read/Write

Bit	Description
7:0	**SMBus Host Slave Command (SMBCMD)—R/W.** Specifies the command values to be matched for SMBus master accesses to the SMBus controller host slave interface (SMBus port 10h).

7.1.30. SMBSHDW1—SMBUS SLAVE SHADOW PORT 1 (FUNCTION 3)

Address Offset: D4h
Default Value: 00h
Attribute: Read/Write

Bit	Description
7:1	**SMBus Slave Address for shadow port 1 (SLVPORT1)—R/W.** Specifies the address used to match against incoming SMBus addresses for shadow port 1.
0	**Read/Write for shadow port 1 (SLVPORT1RW)—R/W.** This bit must be programmed to 0 since PIIX4 SMBus slave controller only responds to Word Write transactions.

7.1.31. SMBSHDW2—SMBUS SLAVE SHADOW PORT 2 (FUNCTION 3)

Address Offset: D5h
Default Value: 00h
Attribute: Read/Write

Bit	Description
7:1	**SMBus Slave Address for shadow port 2 (SLVPORT2)—R/W.** Specifies the address used to match against incoming SMBus addresses for shadow port 2.
0	**Read/Write for shadow port 2 (SLVPORT2RW)—R/W.** This bit must be programmed to 0 since PIIX4 SMBus slave controller only responds to Word Write transactions.

7.1.32. SMBREV—SMBUS REVISION IDENTIFICATION (FUNCTION 3)

Address Offset: D6h
Default Value: 00h
Attribute: Read Only

Bit	Description
7:0	**Revision ID (REVID)—RO.** This register returns the current revision ID for the SMBus Host/Slave controller.

7.2. Power Management IO Space Registers

The “Base” address is programmed in the PIIX4 PCI Configuration Space for Function 3, Offset 40h–43h.

7.2.1. PMSTS—POWER MANAGEMENT STATUS REGISTER (IO)

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

Bit	Description
15	**Resume Status (RSM_STS)—R/WC.** 1=An enabled resume event has occurred. 0=An enabled resume event has not occurred. PIIX4 sets this bit to 1 upon detection of the resume event and then transitions the system to the on state. This bit can only be set by hardware and can only be cleared by writing a 1 to this bit position.
14:12	**Reserved.**
11	**Power Button Override Status (PWRBTNOR_STS)—R/WC.** 1=Power Button Override has been signaled. 0=Power Button Override has not been signaled. This bit is set when Power Button Override has been enabled and the PWRBTN# signal has been continuously asserted for greater than 4 seconds. PIIX4 automatically transitions the system into the soft off state and clears the PWRBTN_STS bit. This bit is only set by hardware and can only be reset by writing a 1 to this bit position.
10	**RTC Status (RTC_STS)—R/WC.** 1=RTC alarm has been signaled. 0=RTC alarm has not been signaled. This bit is set when the internal RTC asserts its IRQ8 signal. This bit is only set by hardware and can only be reset by writing a 1 to this bit position.
9	**Reserved.**
8	**Power Button Status (PWRBTN_STS)—R/WC.** 1=PWRBTN# signal has been asserted. 0=PWRBTN# signal has not been asserted. There is a 16 ms delay from external signal assertion to the setting of this bit due to internal switch debounce circuitry. This bit is only set by hardware and can only be reset by writing a 1 to this bit position. If the PWRBTN# signal is held LOW for more than 4 seconds, then this bit is cleared and the PWRBTNOR_STS bit is set.
7:6	**Reserved.**
5	**Global Status (GBL_STS)—R/WC.** 1=SCI has been generated due a write of 1 to the BIOS_RLS bit. 0=No SCI has been generated due to write to BIOS_RLS bit. The GLB_EN bit must be set to enable the SCI generation. This bit is only set by hardware and can only be reset by writing a 1 to this bit position.
4	**Bus Master Status (BM_STS)—R/WC.** 1=PCIREQ[0:3] or PHOLD# has been asserted (PCI Bus Master request). 0=No Bus Master request. This bit is set when PCIREQ[0:3] or PHOLD# is asserted and can only be cleared by writing a 1 to this bit position.
3:1	**Reserved.**
0	**Timer Overflow Status (TMROF_STS)—R/WC.** 1=Bit 23 of the 24-bit Power Management timer has toggled. 0=Bit has not toggled. When the TMROF_EN is set then the setting of the TMROF_STS bit will additionally generate an SCI. This bit is only set by hardware and can only be reset by writing a 1 to this bit position.

7.2.2. PMEN—POWER MANAGEMENT RESUME ENABLE REGISTER (IO)

I/O Address: Base + (02h)
Default Value: 00h
Attribute: Read/Write

Bit	Description
15:11	**Reserved.**
10	**RTC Enable (RTC_EN)—R/W.** 1=Enable the generation of a resume event upon setting of the RTC_STS bit. 0=Disable.
9	**Reserved.**
8	**Power Button Enable (PWRBTN_EN)—R/W.** 1=Enable the generation of an SMI# or SCI on setting the STS bit. 0=Disable. The PWRBTN# signal is always enabled to generate resume events.
7:6	**Reserved.**
5	**Global Enable (GBL_EN)—R/W.** 1=Enable SCI generation upon setting of the GBL_STS bit. 0=Disable.
4:1	**Reserved.**
0	**Power Management Timer (TMROF_EN)—R/W.** 1=Enable SCI generation upon setting of the TMROF_STS bit. 0=Disable.

7.2.3. PMCNTRL—POWER MANAGEMENT CONTROL REGISTER (IO)

I/O Address: Base + (04h)
Default Value: 0000h
Attribute: Read/Write

Bit	Description
15:14	**Reserved.**
13	**Suspend Enable (SUS_EN)—R/W.** This is a write-only bit and reads to it always return a 0. Writing this bit to a 1 causes the system to automatically sequence into the suspend state defined by the SUS_TYP field. This bit corresponds to the SLP_EN bit in ACPI specification.
12:10	**Suspend Type (SUS_TYP)—R/W.** Specifies the type of hardware suspend mode the system should enter when the SUS_EN bit is set. This field is decoded as follows: **Bits[12:10] Suspend Type** 000 Soff/STD (Soft OFF or Suspend to Disk) 001 STR (Suspend To RAM) 010 POSCL (Powered On Suspend, Context Lost) 011 POSCCL (Powered On Suspend, CPU Context Lost) 100 POS (Powered On Suspend, Context Maintained) 101 Working (Clock Control) 110 Reserved 111 Reserved The SUS_TYP field may also be used by the BIOS and OS code to determine the type of suspend state the system is resuming from. Before entering any low power clock control state (LVL2 or LVL3), this field should be programmed to the Working state (101). This does not cause any action by PIIX4, but is for information storage only.

Bit	Description
9:3	**Reserved**.
2	**Global Release (GBL_RLS)—R/W.** 1=A 1 written to this bit position will cause an SMI# to be generated and BIOS_STS bit set if enabled by the BIOS_EN bit. 0=No SMI# generated. This bit is used by the ACPI software to raise an event to the BIOS software.
1	**Bus Master Reload Enable (BRLD_EN_BM)—R/W.** 1=Enable the generation of a Burst or Stop Break event upon setting of the BM_STS bit. 0=Disable.
0	**SCI Enable (SCI_EN)—R/W.** 1=Enable generation of SCI upon assertion of PWRBTN_STS, LID_STS, THRM_STS, or GPI_STS bits. 0=Disable.

7.2.4. PMTMR—POWER MANAGEMENT TIMER REGISTER (IO)

I/O Address: Base + (08h)
Default Value: 00h
Attribute: Read Only

Bit	Description
23:0	**Timer Value (TMR_VAL)—RO.** This field returns the running count of the power management timer. This is a 24-bit counter that runs off a 3.579545-MHz clock. The timer is reset to an initial value of 0 during a PCI reset, and then continues counting unless the 14.31818-MHz OSC input to the chip is stopped. If the clock is restarted without a PCI reset, then the counter will continue counting from where it stopped. When bit 23 of the timer transitions from high-to-low or low-to-high, the TMROF_STS bit is set. If the TMROF_EN bit is set an SCI interrupt is also generated.

7.2.5. GPSTS—GENERAL PURPOSE STATUS REGISTER (IO)

I/O Address: Base + (0Ch)
Default Value: 00h
Attribute: Read/Write

Bit	Description
15:12	**Reserved.**
11	**LID Status (LID_STS)—R/WC.** 1=LID signal has been asserted. 0=LID signal has not been asserted. Assertion level is dependent upon polarity enable bit LID_POL value. If the LID_EN bit is set then the setting of the LID_STS bit will generate an SCI, SMI# or resume event. This bit is only set by hardware and can only be reset by writing a 1 to this bit position.
10	**Ring Status (RI_STS)—R/WC.** 1=Ring Indicate RI# signal has been asserted. 0=RI# has not been asserted. If the RI_EN bit is set, the setting of the RI_STS bit generates a resume event. This bit is only set by hardware and can only be reset by writing a 1 to this bit position.
9	**GPI Status (GPI_STS)—R/WC.** 1=GPI1# signal has been asserted. 0=GPI1# has not been asserted. If the GPI_EN bit is set then the setting of the GPI_STS bit will generate an SCI, SMI# or resume event. This bit is only set by hardware and can only be reset by writing a 1 to this bit position.

Bit	Description
8	**USB Status (USB_STS)—R/WC.** 1=USB interface has indicated that a USB resume has been driven onto one of the two USB ports while in Power On Suspend. 0=No USB resume has been detected. If the USB_EN bit is set the setting of the USB_STS bit will generate a resume event. This bit is only set by hardware and can only be reset by writing a 1 to this bit position.
7	**Thermal Override Status (THRMOR_STS)—R/WC.** 1=THRM# signal has been asserted LOW and thermal clock throttling has been initiated. 0=Thermal clock throttling has not been initiated. This bit is set anytime the thermal state machine generates a thermal override condition and starts throttling the CPU's clock at the THRM_DTY ratio. This bit is set by hardware and can only be cleared by writing a 1 to this bit position.
6:1	**Reserved.**
0	**Thermal Status (THRM_STS)—R/WC.** 1=THRM# signal has been asserted. 0=THRM# signal has not been asserted. Assertion level is dependent upon polarity enable bit THRM_POL. If the THRM_EN bit is set then the setting of the THRM_STS bit will generate an SCI or SMI. This bit is only set through hardware and is cleared by writing a 1 to this bit position.

7.2.6. GPEN—GENERAL PURPOSE ENABLE REGISTER (IO)

I/O Address: Base + (0Eh)
Default Value: 00h
Attribute: Read/Write

Bit	Description
15:12	**Reserved.**
11	**Lid Enable (LID_EN)—R/W.** 1=Enable the generation of an SMI#, SCI, or resume event upon the setting of the LID_STS bit. 0=Disable.
10	**Ring Enable (RI_EN)—R/W.** 1=Enable the generation of a resume event upon the setting of the RI_STS bit. 0=Disable.
9	**GPI Enable (GPI_EN)—R/W.** 1=Enable the generation of an SMI#, SCI, or resume event upon the setting of the GPI_STS bit. 0=Disable.
8	**USB Enable (USB_EN)—R/W.** 1=Enable the generation of a resume event upon the setting of the USB_STS bit. 0=Disable.
7:1	**Reserved.**
0	**Thermal Enable (THRM_EN)—R/W.** 1=Enable the generation of an SMI# or SCI upon the setting of the THRM_STS bit. 0=Disable.

7.2.7. PCNTRL—PROCESSOR CONTROL REGISTER (IO)

I/O Address: Base + (10h)
Default Value: 00h
Attribute: Read/Write

Bit	Description
31:18	**Reserved.**
17	**Clock Control Status (CC_STS)—RO.** 1=PIIX4 clock control active. 0=PIIX4 clock control inactive.
16:14	**Reserved.**
13	**Clock Run Enable (CLKRUN_EN)—R/W.** 1=Enable PCI Clock Run (CLKRUN#) protocol. 0=Disable. When enabled, PIIX4 requests to stop the PCI clock when the PCI bus has been idle for 26 PCI clocks.
12	**Stop Clock Enable (STPCLK_EN)—R/W.** 1=Enable stopping of Host clock when placed into a LVL3 clock control condition. 0=Disable.
11	**Sleep Enable (SLEEP_EN)—R/W.** 1=Enable assertion of SLP# signal when placed into LVL3 clock control condition. 0=Disable. This enables Sleep or Deep Sleep clock control for Pentium II processor.
10	**Burst Enable (BST_EN)—R/W.** 1=Enable clock control bursting which causes enabled system events to become Burst events and reload the burst timers. 0=Disable clock control bursting which causes enabled system events to become Stop Break events and restore the system to normal full-speed clocked operation.
9	**Clock Control Enable (CC_EN)—R/W.** 1=Enable clock control. 0=Disable. This enables reads to the LVL2 and LVL3 registers to cause PIIX4 to enter the enabled clock mode.
8	**Reserved.**
7:5	**Reserved.**
4	**Throttle Enable (THT_EN)—R/W.** 1=Enable system throttle clock control. 0=Disable.
3:1	**Throttle Duty Programming Bits (THTL_DTY)—R/W.** Selects the duty cycle of the STPCLK# signal when the system is in the system throttling mode. The duty cycle indicates the percentage of time the STPCLK# signal is asserted while in the throttle mode. The field is decoded as follows: **Bits[2:0] Mode Bits[2:0] Mode** 000 Reserved 100 50% 001 12.5% 101 62.5% 010 25% 110 75% 011 37.5% 111 87.5%
0	**Reserved.**

7.2.8. PLVL2—PROCESSOR LEVEL 2 REGISTER (IO)

I/O Address: Base + (14h)
Default Value: 00h
Attribute: Read/Write (Byte Readable Only)

Bit	Description
7:0	**Level 2 Power State Entry (LVL2)—R/W.** Reads to this register cause PIIX4 to transition into a Stop Grant or Quick Start power state (LVL2) and return a value of 00h. Writes to this register have no effect.

7.2.9. PLVL3—PROCESSOR LEVEL 3 REGISTER (IO)

I/O Address: Base + (15h)
Default Value: 00h
Attribute: Read/Write (Byte Readable Only)

Bit	Description
7:0	**Level 3 Power State Entry (LVL3)—R/W.** Reads to this register cause PIIX4 to transition into a Stop Clock, Sleep, or Deep Sleep power state (LVL3) and return a value of 00h. Writes to this register have no effect.

7.2.10. GLBSTS—GLOBAL STATUS REGISTER (IO)

I/O Address: Base + (18h)
Default Value: 00h
Attribute: Read/Write

Bit	Description
15:12	**Reserved.**
11	**IRQ Resume Status (IRQ_RSM_STS)—R/W.** 1=System was resumed from a Powered On Suspend (POS) state due to an interrupt assertion (IRQ[1,3:15]). 0=System was not resumed due to IRQ. This bit is only set by hardware and can only be reset by writing a 1 to this bit position.
10	**External SMI Status (EXTSMI_STS)—R/WC.** 1=EXTSMI# signal was asserted. 0=EXTSMI# was not asserted. This bit is only set by hardware and can only be reset by writing a 1 to this bit position.
9	**Reserved.**
8	**Global Standby Status (GSTBY_STS)—R/WC.** 1=Global Standby timer expired (counted down to 0). 0=Global Standby timer did not expire. This bit is only set by hardware and can only be reset by writing a 1 to this bit position.
7	**GP Status (GP_STS)—RO.** 1=Indicates that one of the status bits in the GPSTS register is set. 0=All bits in GPSTS register are reset. This bit can only be reset by resetting all bits in the GPSTS register.
6	**PM1 Status (PM1_STS)—RO.** 1=Indicates that one of the status bits in the PMSTS register is set. 0=All bits in PMSTS register are reset. This bit can only be reset by resetting all bits in the PMSTS register.
5	**APM Status (APM_STS)—R/WC.** 1=A write occurred to the APMC register causing generation of an SMI#. 0=A write has not occurred to the APMC register causing generation of an SMI#. This bit is cleared by writing a 1 to this bit position.
4	**All Devices Status (DEV_STS)—RO.** 1=Indicates that one of the status bits in the DEV_STS register is set. 0=All bits in DEV_STS register are reset. This bit can only be reset by resetting all bits in the DEV_STS register.
3	**Reserved.**
2	**P4MA Status (P4MA_STS)—R/WC.** 1=An SMI# was generated due to a PIIX4 PCI cycle being Master Aborted. 0=No SMI# was generated due to PIIX4 PCI cycles having been Master Aborted. This bit is only set by hardware and can only be reset by writing a 1 to this bit position.
1	**Legacy USB Status (LEGACY_USB_STS)—R/WC.** 1=USB legacy keyboard logic generated an SMI#. 0=USB legacy keyboard logic did not generate an SMI#. This bit is only set by hardware and can only be reset clearing the children status bits in the USB status register.
0	**BIOS Status (BIOS_STS)—R/WC.** 1=A write of 1 occurred to the GBL_RLS bit. 0=A write of 1 did not occur to the GBL_RLS bit. This bit is set by hardware and is cleared by writing a 1 to it.

7.2.11. DEVSTS—DEVICE STATUS REGISTER (IO)

I/O Address: Base + (1Ch)
Default Value: 00h
Attribute: Read/Write

Bit	Description
31:30	**Reserved.**
29:16	**Device [0–13] Trap Status Bits (TRP_STS_DEV[0–13])—R/WC.** 1=An SMI# was generated by an I/O trap to the associated device monitor's enabled address range. 0=No SMI# was generated. Bit 29 corresponds to device monitor 13 and bit 16 corresponds to device monitor 0. This bit is cleared by writing a 1 to its bit position.
15:12	**Reserved.**
11:0	**Device [0–11] Idle Status Bits (IDL_STS_DEV[0–11])—R/WC.** 1=An SMI# was generated by the expiration of the associated device monitor's idle timer. 0=No SMI# was generated. Bit 11 corresponds to device monitor 11 and bit 0 corresponds to device monitor 0. This bit is cleared by writing a 1 to its bit position.

7.2.12. GLBEN—GLOBAL ENABLE REGISTER (IO)

I/O Address: Base + (20h)
Default Value: 00h
Attribute: Read/Write

Bit	Description
15	**Battery Low Enable (BATLOW_EN)—R/W.** 1=Enable BATLOW# assertion to prevent a system resume from any suspend state. 0=Disable.
14:12	**Reserved.**
11	**IRQ Resume Enable (IRQ_RSM_EN)—R/W.** 1=Enable an unmasked interrupt (IRQ[1, 3:15]) assertion to generate a resume from the Power On Suspend (POS) state. 0=Disable.
10	**External SMI Enable (EXTSMI_EN)—R/W.** 1=Enable the setting of the EXTSMI_STS bit to generate an SMI# or resume event. 0=Disable.
9	**Reserved.**
8	**Global Stand By Enable (GSTBY_EN)—R/W.** 1=Enable the setting of the GSTBY_STS bit to generate an SMI# or resume event. 0=Disable.
7:5	**Reserved.**
4	**PIIX4 Master Abort Enable (P4MA_EN)—R/W.** 1=Enable the setting of the P4MA_STS bit to generate an SMI#. 0=Disable.
3	**Bus Master Trap Enable (BM_TRP_EN)—R/W.** 1=Enable the setting of the BM_STS bit to generate an SMI#.
2	**Reserved.**
1	**BIOS Enable (BIOS_EN)—R/W.** 1=Enable the generation of an SMI# by writing a 1 to the GBL_RLS bit. 0=Disable.
0	**Legacy USB Enable (LEGACY_USB_EN)—R/W.** 1=Enable the USB legacy function to generate an SMI#. 0=Disable.

7.2.13. GLBCTL—GLOBAL CONTROL REGISTER (IO)

I/O Address: Base + (28h)
Default Value: 00h
Attribute: Read/Write

Bit	Description
31:27	**Reserved.**
26	**Global Standby Timer Clocking Select B (GSTBY_SELB)—R/W.** This bit in conjunction with bit 8 selects the clock source for the Global Standby Timer. See bit 8 description for timing programming combinations.
25	**Lid Polarity (LID_POL)—R/W.** 1=Active low LID assertion sets the LID_STS bit. 0=Active high LID assertion sets the LID_STS bit.
24	**System Management Freeze (SM_FREEZE)—R/W.** 1=Disable all Device Monitor idle timers and the Global Standby Timer from counting. 0=Enable timers to count.
23:17	**Reserved.**
16	**End of SMI (EOS)—R/W.** 1=Enable PIIX4 to assert an SMI#. 0=Disable. This bit is cleared automatically upon generation of an SMI#.
15:9	**Global Standby Timer Initial Count (GSTBY_CNT)—R/W.** Specifies the initial and reload count of the Global Standby Timer.
8	**Global Standby Timer Clocking Select A (GSTBY_SELA)—R/W.** This bit in conjunction with bit 26 selects the clock source for the Global Standby Timer. **Bits[26,8] Clock Rate** 0,0 32 seconds (default) 0,1 4 minutes 1,0 4 milliseconds 1,1 4 seconds
7:3	**Reserved.**
2	**Thermal Polarity (THRM_POL)—R/W.** 1=Active low THRM# assertion sets the THRM_STS bit. 0=Active high THRM# assertion sets the THRM_STS bit.
1	**BIOS Release (BIOS_RLS)—R/W.** 1=A 1 written to this bit position causes an SCI to be generated and GBL_STS bit set, if enabled by the GBL_EN bit. 0=No SCI generated. This bit is used by the BIOS software to raise an event to the ACPI software. This bit always reads a 0.
0	**SMI ENABLE (SMI_EN)—R/W.** 1=Enable the generation of SMI# upon any enabled SMI# event. 0=Disable. This bit is reset by a PCI reset event.

7.2.14. DEVCTL—DEVICE CONTROL REGISTER (IO)

I/O Address: Base + (2Ch)
Default Value: 00h
Attribute: Read/Write

Bit	Description
31:28	**Reserved.**
27	**Device 8 Bus Master Reload Enable (BM_RLD_DEV8)—R/W.** 1=Enable any PCI Bus Master request (PHOLD#, PCIREQA[0:3]) to reload the device monitor idle timer. 0=Disable.
26	**Device 3 Idle Reload Enable (IDL_RLD_EN_DEV3)—R/W.** 1=Enable the device monitor 3 idle reload events to reload the device monitor 3 idle timer. 0=Disable. When device 3 is being used as a software SMI timer, this bit should be cleared to prevent any events from reloading the timer.
25	**Device 13 Trap Enable (TRP_EN_DEV13)—R/W.** 1=Enable generation of a trap SMI for accesses to the device monitor 13 enabled trap decode ranges. 0=Disable.
24	**Device 12 Trap Enable (TRP_EN_DEV12)—R/W.** 1=Enable generation of a trap SMI for accesses to the device monitor 12 enabled trap decode ranges. 0=Disable.
23	**Device 11 Trap Enable (TRP_EN_DEV11)—R/W.** 1=Enable generation of a trap SMI for accesses to the device monitor 11 enabled trap decode ranges. 0=Disable.
22	**Device 11 Idle Enable (IDL_EN_DEV11)—R/W.** 1=Enable the device monitor 11 idle reload events to reload the device monitor 11 idle timer. 0=Disable.
21	**Device 10 Trap Enable (TRP_EN_DEV10)—R/W.** 1=Enable generation of a trap SMI for accesses to the device monitor 10 enabled trap decode ranges. 0=Disable.
20	**Device 10 Idle Enable (IDL_EN_DEV10)—R/W.** 1=Enable the device monitor 10 idle reload events to reload the device monitor 10 idle timer. 0=Disable.
19	**Device 9 Trap Enable (TRP_EN_DEV9)—R/W.** 1=Enable generation of a trap SMI for accesses to the device monitor 9 enabled trap decode ranges. 0=Disable.
18	**Device 9 Idle Enable (IDL_EN_DEV9)—R/W.** 1=Enable the device monitor 9 idle reload events to reload the device monitor 9 idle timer. 0=Disable.
17	**Device 8 Trap Enable (TRP_EN_DEV8)—R/W.** 1=Enable generation of a trap SMI for accesses to the device monitor 8 enabled trap decode ranges. 0=Disable.
16	**Device 8 Idle Enable (IDL_EN_DEV8)—R/W.** 1=Enable the device monitor 8 idle reload events to reload the device monitor 8 idle timer. 0=Disable.
15	**Device 7 Trap Enable (TRP_EN_DEV7)—R/W.** 1=Enable generation of a trap SMI for accesses to the device monitor 7 enabled trap decode ranges. 0=Disable.
14	**Device 7 Idle Enable (IDL_EN_DEV7)—R/W.** 1=Enable the device monitor 7 idle reload events to reload the device monitor 7 idle timer. 0=Disable.
13	**Device 6 Trap Enable (TRP_EN_DEV6)—R/W.** 1=Enable generation of a trap SMI for accesses to the device monitor 6 enabled trap decode ranges. 0=Disable.
12	**Device 6 Idle Enable (IDL_EN_DEV6)—R/W.** 1=Enable the device monitor 6 idle reload events to reload the device monitor 6 idle timer. 0=Disable.
11	**Device 5 Trap Enable (TRP_EN_DEV5)—R/W.** 1=Enable generation of a trap SMI for accesses to the device monitor 5 enabled trap decode ranges. 0=Disable.

Bit	Description
10	**Device 5 Idle Enable (IDL_EN_DEV5)—R/W.** 1=Enable the device monitor 5 idle reload events to reload the device monitor 5 idle timer. 0=Disable.
9	**Device 4 Trap Enable (TRP_EN_DEV4)—R/W.** 1=Enable generation of a trap SMI for accesses to the device monitor 4 enabled trap decode ranges. 0=Disable.
8	**Device 4 Idle Enable (IDL_EN DEV4)—R/W.** 1=Enable the device monitor 4 idle reload events to reload the device monitor 4 idle timer. 0=Disable.
7	**Device 3 Trap Enable (TRP_EN_DEV3)—R/W.** 1=Enable generation of a trap SMI for accesses to the device monitor 3 enabled trap decode ranges. 0=Disable.
6	**Device 3 Idle Enable (IDL_EN_DEV3)—R/W.** 1=Enable the device monitor 3 idle reload events to reload the device monitor 3 idle timer. 0=Disable.
5	**Device 2 Trap Enable (TRP_EN_DEV2)—R/W.** 1=Enable generation of a trap SMI for accesses to the device monitor 2 enabled trap decode ranges. 0=Disable.
4	**Device 2 Idle Enable (IDL_EN_DEV2)—R/W.** 1=Enable the device monitor 2 idle reload events to reload the device monitor 2 idle timer. 0=Disable.
3	**Device 1 Trap Enable (TRP_EN_DEV1)—R/W.** 1=Enable generation of a trap SMI for accesses to the device monitor 1 enabled trap decode ranges. 0=Disable.
2	**Device 1 Idle Enable (IDL_EN_DEV1)—R/W.** 1=Enable the device monitor 1 idle reload events to reload the device monitor 1 idle timer. 0=Disable.
1	**Device 0 Trap Enable (TRP_EN_DEV0)—R/W.** 1=Enable generation of a trap SMI for accesses to the device monitor 0 enabled trap decode ranges. 0=Disable.
0	**Device 0 Idle Enable (IDL_EN_DEV0)—R/W.** 1=Enable the device monitor 0 idle reload events to reload the device monitor 0 idle timer. 0=Disable.

7.2.15. GPIREG—GENERAL PURPOSE INPUT REGISTER (IO)

I/O Address: Base + (30h, 31h, 32h)
Default Value: XXh
Attribute: Read Only (Byte reads only)

Bit	Description
23:22	**Reserved.**
21:0	**General Purpose Input (GPI)—RO.** Each bit directly represents the logical value on the pin. Some of the GPI signals can be configured as another input signal. The value in this register of a bit which is not configured as a GPI is indeterminate and may change randomly.

7.2.16. GPOREG—GENERAL PURPOSE OUTPUT REGISTER (IO)

I/O Address: Base + (34h, 35h, 36h, 37h)
Default Value: 7FFFBFFFh
Attribute: Read/Write (Byte accesses only)

Bit	Description
31	**Reserved.**
30:0	**General Purpose Output (GPO)—R/W.** Each bit directly represents the logical value output onto the pin. Reads to this register return the last value written. Some GPO signals can be configured as another output signal. In that case, the output pin will not reflect the state of the corresponding GPO bit in this register. Some of the output signals default to another signal.

7.3. SMBus IO Space Registers

The “Base” address is programmed in the PIIX4 PCI Configuration Space for Function 3 (Offset 90h–93h).

7.3.1. SMBHSTSTS—SMBUS HOST STATUS REGISTER (IO)

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

This register provides status information concerning the SMBus controller host interface.

Bit	Description
7:5	**Reserved.**
4	**Failed (FAILED)—R/WC.** 1=Indicates that the source of SMBus interrupt was a failed bus transaction, set when KILL bit is set (SMBHSTCNT register). 0=SMBus interrupt not caused by KILL bit. This bit is only set by hardware and can only be reset by writing a 1 to this bit position.
3	**BUS COLLISION (BUS_ERR)—R/WC.** 1=Indicates that the source of SMBus interrupt was a transaction collision. 0=SMBus interrupt not caused by transaction collision. This bit is only set by hardware and can only be reset by writing a 1 to this bit position.
2	**Device Error (DEV_ERR)—R/WC.** 1=Indicates that the source of SMBus interrupt was the generation of an SMBus transaction error. 0=SMBus interrupt not caused by transaction error. This bit is only set by hardware and can only be reset by writing a 1 to this bit position. Transaction errors are caused by: • Illegal Command Field • Unclaimed Cycle (host initiated) • Host Device Time-out
1	**SMBus Interrupt (INTER)—R/WC.** 1=Indicates that the source of SMBus interrupt was the completion of the last host command. 0=SMBus interrupt not caused by host command completion. This bit is only set by hardware and can only be reset by writing a 1 to this bit position.
0	**Host Busy (HOST_BUSY)—RO.** 1=Indicates that the SMBus controller host interface is in the process of completing a command. 0=SMBus controller host interface is not processing a command. None of the other registers should be accessed if this bit is set.

7.3.2. SMBSLVSTS—SMBUS SLAVE STATUS REGISTER (IO)

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

This register provides status information concerning the SMBus controller slave interface.

Bit	Description
7:6	**Reserved.**
5	**Alert Status (ALERT_STS)—R/WC.** 1=Indicates that the source of SMBus interrupt or resume event was the assertion of the SMBALERT# signal. 0=SMBus interrupt not caused by SMBALERT# signal. Setting of this bit requires that the ALERT_EN bit be set. This bit is only set by hardware and can only be reset by writing a 1 to this bit position.
4	**Shadow2 Status (SHDW2_STS)—R/WC.** 1=Indicates that the source of SMBus interrupt or resume event was a slave cycle address match of the SMBSHDW2 port. 0=SMBus interrupt not caused by address match to SMBSHDW2 port. This bit is only set by hardware and can only be reset by writing a 1 to this bit position.
3	**Shadow1 Status (SHDW1_STS)—R/WC.** 1=Indicates that the source of SMBus interrupt or resume event was a slave cycle address match of the SMBSHDW1 port. 0=SMBus interrupt not caused by address match to SMBSHDW1 port. This bit is only set by hardware and can only be reset by writing a 1 to this bit position.
2	**Slave Status (SLV_STS)—R/WC.** 1=Indicates that the source of SMBus interrupt or resume event was a slave cycle event match of the SMBSLVC (command match) and SMBSLVEVT (data event match). 0=SMBus interrupt not caused by slave event match. This bit is only set by hardware and can only be reset by writing a 1 to this bit position.
1	**Reserved.**
0	**Slave Busy (SLV_BSY)—RO.** 1=Indicates that the SMBus controller slave interface is in the process of receiving data. 0=SMBus controller slave interface is not processing data. None of the other registers should be accessed if this bit is set.

7.3.3. SMBHSTCNT—SMBUS HOST CONTROL REGISTER (IO)

I/O Address: Base + (02h)
Default Value: 00h
Attribute: Read/Write

The control register is used to enable SMBus controller host interface functions. Reads to this register clears the host interface's index pointer to the block data storage array.

Bit	Description
7	**Reserved.**
6	**Start (START)—R/W.** 1=Start execution. Writing a 1 to this bit initiates the SMBus controller host interface to execute the command programmed in the SMB_CMD_PORT field. All necessary registers should be setup prior to writing a 1 to this bit position. 0=Writing a 0 has no effect. This bit always reads 0. The HOST_BUSY bit can be used to identify when the SMBus host controller has finished executing the command.
5	**Reserved.**
4:2	**SMBus Command Protocol (SMB_CMD_PROT)—R/W.** Selects the type of command the SMBus controller host interface will execute. Reads or writes are determined by bit 0 of SMBHSTADD register. This field is decoded as follows: **Bits[4:2] Protocol — Bits[4:2] Protocol** 000 Quick Read or Write — 100 Reserved 001 Byte Read or Write — 101 Block Read or Write 010 Byte Data Read or Write — 110 Reserved 011 Word Data Read or Write — 111 Reserved
1	**Kill (KILL)—R/W.** 1=Stop the current in process SMBus controller host transaction. This sets the FAILED status bit and asserts the interrupt selected by the SMB_INTRSEL field. 0=Allows the SMBus controller host interface to function normally.
0	**Interrupt Enable (INTEREN)—R/W.** 1=Enable the generation of interrupts (IRQ9OUT) on the completion of the current host transaction. 0=Disable.

7.3.4. SMBHSTCMD—SMBUS HOST COMMAND REGISTER (IO)

I/O Address: Base + (03h)
Default Value: 00h
Attribute: Read/Write

This register is transmitted by the SMBus controller host interface in the command field of the SMBus protocol.

Bit	Description
7:0	**SMBus Host Command (HST_CMD)—R/W**. This field contains the data transmitted in the command field of SMBus host transaction.

7.3.5. SMBHSTADD—SMBUS HOST ADDRESS REGISTER (IO)

I/O Address: Base + (04h)
Default Value: 00h
Attribute: Read/Write

This register is transmitted by the SMBus controller host interface in the slave address field of the SMBus protocol.

Bit	Description
7:1	**SMBus Address (SMB_ADDRESS)—R/W.** This field contains the 7-bit address of the targeted slave device.
0	**SMBus Read or Write (SMB_RW)—R/W.** 1=Execute a READ command. 0=Execute a WRITE command.

7.3.6. SMBHSTDAT0—SMBUS HOST DATA 0 REGISTER (IO)

I/O Address: Base + (05h)
Default Value: 00h
Attribute: Read/Write

This register is transmitted by the SMBus controller host interface in the Data 0 field of the SMBus protocol. On reads, Data 0 bytes are stored here.

Bit	Description
7:0	**SMBus Data 0 (SMBD0)—R/W.** This register should be programmed with the value to be transmitted in the Data 0 field of an SMBus host interface transaction. For a block write command, the count of the memory block should be stored in this field. The value of this register is loaded into the block transfer count field. This register must be programmed to a value between 1 and 32 for block command counts. A count of 0 or a count above 32 will result in unpredictable behavior. For block reads, the count received from the SMBus device is stored here.

7.3.7. SMBHSTDAT1—SMBUS HOST DATA 1 REGISTER (IO)

I/O Address: Base + (06h)
Default Value: 00h
Attribute: Read/Write

This register is transmitted by the SMBus controller host interface in the Data 1 field of the SMBus protocol. On reads, Data 1 bytes are stored here.

Bit	Description
7:0	**SMBus Data 1 (SMBD1)—R/W.** This register should be programmed with the value to be transmitted in the Data 1 field of an SMBus host interface transaction.

7.3.8. SMBBLKDAT—SMBUS BLOCK DATA REGISTER (IO)

I/O Address: Base + (07h)
Default Value: 00h
Attribute: Read/Write

Reads and writes to this register are used to access the 32-byte block data storage array. An internal index pointer is used to address the array. It is reset to 0 by reading the SMBHSTCNT register. The index pointer then increments automatically upon each access to this register. The transfer of block data into (read) or out of (write) this storage array during an SMBus transaction always starts at index address 0.

Bit	Description
7:0	**SMBus Block Data (BLK_DAT)—R/W.** This register is used to transfer data into or out of the block data storage array.

7.3.9. SMBSLVCNT—SMBUS SLAVE CONTROL REGISTER (IO)

I/O Address: Base + (08h)
Default Value: 00h
Attribute: Read/Write

The control register is used to enable SMBus controller slave interface functions.

Bit	Description
7:4	**Reserved.**
3	**SMBus Alert Enable (ALERT_EN)—R/W.** 1=Enable the generation of an interrupt or resume event on the assertion of SMBALERT# signal. 0=Disable.
2	**SMBus Shadow Port 2 Enable (SHDW2_EN)—R/W.** 1=Enable the generation of an interrupt or resume event upon an external SMBus master generating a transaction with an address that matches the SMBSHDW2 register. 0=Disable.
1	**SMBus Shadow Port 1 Enable (SHDW1_EN)—R/W.** 1=Enable the generation of an interrupt or resume event upon an external SMBus master generating a transaction with an address that matches the SMBSHDW1 register. 0=Disable.
0	**Slave Enable (SLV_EN)—R/W.** 1=Enable the generation of an interrupt or resume event upon an external SMBus master generating a transaction with an address that matches the host controller slave port of 10h, a command field which matches the SMBSLVC register, and a match of one of the corresponding enabled events in the SMBSLVEVT register. 0=Disable.

7.3.10. SMBSHDWCMD—SMBUS SHADOW COMMAND REGISTER (IO)

I/O Address: Base + (09h)
Default Value: 00h
Attribute: Read Only

This register is used to store command values for external SMBus master accesses to the host slave and slave shadow ports.

Bit	Description
7:0	**Shadow Command (SHDW_CMD)—RO.** This field contains the command value which was received during an external SMBus master access whose address field matched the host slave address (10h) or one of the slave shadow port addresses.

7.3.11. SMBSLVEVT—SMBUS SLAVE EVENT REGISTER (IO)

I/O Address: Base + (0Ah)
Default Value: 0000h
Attribute: Read/Write

This register is used to enable generation of interrupt or resume events for accesses to the host controller's slave port.

Bit	Description
15:0	**SMBus Slave Event (SMB_SLV_EVT)—R/W.** This field contains data bits used to compare against incoming data to the SMBSLVDAT register. When a bit in this register is a 1 and the corresponding bit in the SMBSLVDAT register is set, then an interrupt or resume event will be generated if the command value matches the value in the SMBSLVC register and the access was to SMBus host address 10h.

7.3.12. SMBSLVDAT—SMBUS SLAVE DATA REGISTER (IO)

I/O Address: Base + (0Ch)
Default Value: 0000h
Attribute: Read Only

This register is used to store data values for external SMBus master accesses to the shadow ports or the SMBus host controller's slave port.

Bit	Description
15:0	**Slave Data (SMB_SLV_DATA)—RO.** This field contains the data value which was transmitted during an external SMBus master access whose address field matched one of the slave shadow port addresses or the SMBus host controller slave port address of 10h.

8.0. PCI/ISA BRIDGE FUNCTIONAL DESCRIPTION

This section describes each of the major functions on the PIIX4 PCI-to-ISA Bridge including the memory and I/O address map, DMA controller, interrupt controller, timer/counter and X-Bus interfaces.

8.1. Memory and IO Address Map

PIIX4 interfaces to two system buses—PCI and ISA Buses. PIIX4 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 PIIX4 registers as described in the Register Description section. ISA masters and DMA devices do not have access to host or PCI I/O space.

8.1.1. I/O ACCESSES

PIIX4 positively decodes accesses to all internal registers, including PCI configuration registers (PCI only), ISA-Compatible IO registers (PCI and ISA), and all relocatable IO space registers (IDE, USB, Power Management). Accesses to the ISA/EIO bus can be configured to be either subtractive decode or positive decode. PIIX4 provides a wide variety of positive decode ranges for standard devices as well as a number of programmable ranges for additional devices. PIIX4 also provides positive decode for BIOS, X-Bus, and system event decode for power management support. In addition, PIIX4 positively decodes PCI Bus accesses to registers located on the IDE device, when enabled.

8.1.2. MEMORY ADDRESS MAP

For PCI accesses to ISA memory, accesses below 16 Mbytes (including BIOS space) that are not claimed by a PCI device are forwarded to ISA when subtractive decode is enabled. If subtractive decode is disabled, PIIX4 forwards cycles for programmable ranges (32 KB–4 MB) associated with power management devices 12 and 13 and for BIOS ranges described below. 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, PIIX4 returns data corresponding to the state of the ISA Bus and completes the cycle normally (i.e., 8-bit, 6 SYSCLK cycle).

For ISA or 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 18). The Top of Memory is equal to the value programmed in the Top of Memory Register (bits [7:3]). All remaining ISA originated memory accesses are confined to the ISA Bus.

PIIX4 also forwards any accesses to an enabled I/O APIC address range. See descriptions for the APIC Base Address Relocation Register and X-Bus Chip Select Register for additional information.

Table 18. 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 PCI[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.

8.1.3. BIOS MEMORY

PIIX4 supports 1 Mbyte of BIOS memory space. This includes the normal 128-Kbyte space plus an additional 384 Kbyte (extended BIOS space) and 512 Kbyte of BIOS space (1M extended BIOS area). The XBCS Register provides BIOS space access control. Access to the lower 64-Kbyte block of the 128-Kbyte space and both extended BIOS spaces can be individually enabled or 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 block is split into two 64-Kbyte blocks. Accesses to the top 64 Kbytes (000F0000–000FFFFFh) and its aliased region (FFFF0000–FFFFFFFFh) are always 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 (bit 6=1 in the XBCS Register). If this BIOS region is enabled, accesses to the aliased region at the top of 4 Gbytes (FFFE0000h–FFFEFFFFh) are also forwarded to ISA and BIOSCS# generated. If disabled, these accesses are not forwarded to ISA and BIOSCS# is not generated.

The extended BIOS space resides at FFF80000–FFFDFFFFh. If this BIOS region is enabled (bit 7=1 in the XBCS Register), these accesses are forwarded to ISA and BIOSCS# generated. The 1M extended BIOS space resides at FFF00000–FFF7FFFFh. If this BIOS region is enabled (bit 9=1 in the XBCS Register), these accesses are forwarded to ISA and BIOSCS# generated. If disabled, these accesses are not forwarded to ISA and BIOSCS# not generated. Table 19 shows the BIOS Memory Map.

PIIX4 provides a bit in the XBCS register (bit 2) that when set to 0, prevents BIOSCS# from being asserted during BIOS memory write accesses to the decoded BIOS region. When set to 1, BIOSCS# is asserted for memory read and write accesses to the decoded BIOS region. This bit defaults to 0 (BIOS write protected) at reset.

PCI accesses to enabled BIOS memory are always positively decoded, regardless of the status of the Positive/Subtractive Decode Configuration bit (bit 1, Function 0 PCI address B0h).

ISA Access to BIOS Memory

PIIX4 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.

Table 19. BIOS Memory Map

4 GB 4 GB–64 KB	Top 64 KB	FFFFFFFFh FFFF0000h	BIOSCS# always generated.
4 GB–64 KB 4 GB–128 KB	Lower 64 KB	FFFEFFFFh FFFE0000h	
4 GB–128 KB 4 GB–512 KB	Extended 384 KB	FFFDFFFFh FFF80000h	
4 GB–512 KB 4 GB–1 MB	1 MB Extended 512 KB	FFF7FFFFh FFF00000h	
4 GB–128 KB	1 MB Extended 512 KB		
16 MB 16 MB–64 KB	Top 64 KB	00FFFFFFh 00FF0000h	Reserved Memory locations on ISA bus.
16 MB–64 KB 16 MB–128 KB	Lower 64 KB	00FEFFFFh 00FE0000h	Reserved Memory locations on ISA bus if Lower BIOS enabled.
16 MB–128 KB 16 MB–512 KB	Extended 384 KB	00FDFFFFh 00F80000h	Reserved Memory locations on ISA bus if Lower BIOS enabled.
16 MB–512 KB 15 MB	1 MB Extended 512 KB	00F7FFFFh 00F00000h	Reserved Memory locations on ISA bus if Lower BIOS enabled.
1 MB 1 MB–64 KB	Top 64 KB	000FFFFFh 000F0000h	BIOSCS# always generated.
1 MB–64 KB 1 MB–128 KB	Lower 64 KB	000EFFFFh 000E0000h	
0		00000000h	

8.2. PCI Interface

PIIX4 incorporates a fully PCI Bus compatible master and slave interface. As a PCI master, PIIX4 runs cycles on behalf of DMA, ISA masters, bus master IDE, or USB. As a PCI slave, PIIX4 accepts cycles initiated by PCI masters targeted for PIIX4's internal register set or the ISA bus. PIIX4 directly supports the PCI interface running at either 30 or 33 MHz.

8.2.1. TRANSACTION TERMINATION

PIIX4 supports the standard PCI cycle terminations as described in the PCI Local Bus specification.

PIIX4 as Master—Master-Initiated Termination: PIIX4 supports three forms of master-initiated termination: 1.) Normal termination of a completed transaction, 2.) Normal termination of an incomplete transaction due to time-out (applies to line buffer operations-IDE Bus Master, 3.) Abnormal termination due to the slave not responding to the transaction (Abort).

PIIX4 as a Master—Response to Target-Initiated Termination: As a master, PIIX4 responds correctly to the standard target-terminations—Target-Abort, Retry, or Disconnect.

PIIX4 as a Target—Target-Initiated Termination: PIIX4 supports three forms of Target-initiated Termination—Disconnect, Retry, Target Abort.

8.2.2. PARITY SUPPORT

As a master, PIIX4 generates address parity for read/write cycles and data parity when PIIX4 is providing the data. As a slave, PIIX4 generates data parity for read cycles. PIIX4 does not check parity and does not generate SERR# due to an address parity error. However, PIIX4 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.

8.2.3. PCI ARBITRATION

PIIX4 requests the use of the PCI Bus on behalf of ISA devices (bus masters and DMA), IDE DMA slave devices, and the USB Host Controller 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, PIIX4 asserts PHOLD#. PIIX4 keeps DACK negated until PIIX4 has ownership of the PCI Bus and Memory. The PCI arbiter asserts PHLDA# to PIIX4 when the above conditions are met. PIIX4 gives ownership of the ISA Bus (PCI and Memory) to the ISA device after sampling PHLDA# asserted.

The USB Host Controller utilizes the arbitration advantage available through the PHOLD#/PHLDA# protocol 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 as the transactions are completed or the MLT counter has expired, whichever happens first.

PIIX4 uses the delay transaction (or delay completion) and passive release features to help raise the available bandwidth of the PCI bus.

8.3. ISA/EIO Interface

PIIX4 can incorporate a fully ISA Bus compatible master and slave interface or a subset interface called the Extended IO (EIO) Bus. PIIX4 can directly drive the equivalent of five ISA slots without external data buffers. The ISA or EIO signals are independent of other functions and no external transceivers are required. The ISA or EIO 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 PIIX4 or an external ISA master.
- ISA master-initiated memory cycles to PCI and ISA master-initiated I/O cycles to the internal PIIX4 registers, as shown in ISA-Compatible Register table in the Register Mapping section.

The EIO interface differs from ISA interface in following ways:

- ISA Master cycles are not supported.
- Only 20 bits of addressing allowed (No LA signals).
- ISA Refresh is not supported.

8.4. DMA Controller

The DMA circuitry incorporates the functionality of two 82C37 DMA controllers with seven independently programmable channels (Figure 3). DMA Controller 1 (DMA-1) corresponds to DMA Channels 0–3 and DMA Controller 2 (DMA-2) corresponds to Channels 5–7. DMA Channel 4 is used to cascade the two controllers and will default to cascade mode in the DMA Channel Mode (DCM) Register. This channel is not available for any other purpose. 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.

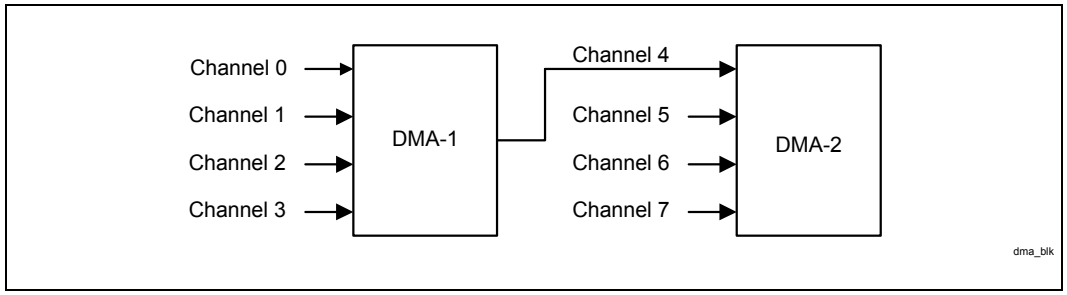


Figure 3. Internal DMA Controller

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. PIIX4 provides the timing control and data size translation necessary for the DMA transfer between the memory (ISA or DRAM) and the ISA Bus IO. ISA-Compatible and Type F-DMA timing is supported.

PIIX4 provides 24-bit addressing in compliance with the ISA-Compatible specification. Each channel includes a 16-bit ISA-Compatible Current Register which holds the 16 least-significant bits of the 24-bit address, an ISA-Compatible Page Register which contains the eight next most significant bits of address.

The DMA controller also features refresh address generation, and autoinitialization following a DMA termination.

The DMA controller is at any time either in master mode 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, PIIX4 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 PIIX4 is running a compatible DMA cycle, it drives the MEMR# or MEMW# strobes if the address is less than 16 Mbytes (000000h–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 (0000000h–00FFFFFh). If the address is greater than 16 Mbytes (1000000h–7FFFFFFh), the MEMR# or MEMW# strobe are not generated to avoid aliasing issues.

NOTE

BIOS Programming: 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 OS is higher than the top of memory programmed in PIIX4 Top of Memory register), the BIOS should not enable type F for any channel.

PIIX4 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.

8.4.1. DMA TRANSFER MODES

The channels can be programmed for any of four transfer modes. The transfer modes include single, block, demand, or cascade. Each of the three active transfer modes (single, block, and demand), can perform three different types of transfers (read, write, or verify). Note that memory-to-memory transfers are not supported by PIIX4.

Single Transfer Mode

In single transfer mode, the DMA is programmed to make one transfer only. The byte/word count is decremented and the address decremented or incremented following each transfer. When the byte/word count "rolls over" from zero to FFFFh, a Terminal Count (TC) causes an autoinitialize if the channel has been programmed to do so.

To be recognized DREQ must be held active until DACK# becomes active. If DREQ is held active throughout the single transfer, the bus is released after a single transfer. With DREQ asserted high, the DMA I/O device re-arbitrate for the bus. Upon winning the bus, another single transfer is performed. This allows other ISA bus masters a chance to acquire the bus.

Block Transfer Mode

In Block Transfer mode, the DMA is activated by DREQ to continue making transfers during the service until a TC, caused by either a byte/word count going to FFFFh, is encountered. DREQ need only be held active until DACK# becomes active. If the channel has been programmed for it, an autoinitialization occurs at the end of the service. In this mode, it is possible to lock out other devices for a period of time (including refresh) if the transfer count is programmed to a large number.

NOTE

Block mode transfers are not supported with type F DMA.

Demand Transfer Mode

In Demand Transfer mode, the DMA channel is programmed to continue making transfers until a TC (Terminal Count) is encountered, or until the DMA I/O device releases DREQ. Thus, transfers may continue until the I/O device has exhausted its data capacity. After the I/O device catches up, the DMA service is re-established when the DMA I/O device reasserts the channel's DREQ. During the time between services when the system is allowed to operate, the intermediate values of address and byte/word count are stored in the DMA controller Current Address and Current Byte/Word Count Registers. A TC can cause an autoinitialize at the end of the service, if the channel has been programmed for it.

Cascade Mode

In Cascade Mode, the DMA controller will respond to DREQ with DACK, but PIIX4 will not drive IOR#, IOW#, MEMR#, MEMW#, LA[23:17], SA[19:0], and SBHE#.

Cascade mode is also used to allow direct access of the system by 16-bit bus masters. These devices use the DREQ and DACK signals to arbitrate for the ISA Bus. The ISA master asserts its ISA master request line (DREQ[x]) to the DMA internal arbiter. If the ISA master wins the arbitration, PIIX4 responds with an ISA master acknowledge (DACK[x]) signal active. Upon sampling the DACK[x] line active, the ISA master takes control of the ISA Bus. While an ISA Master owns the bus, BALE is always driven high while AEN is always driven low. The ISA master has control of the ISA Bus and may run cycles until it negates the DREQ[x] line.

8.4.2. DMA TRANSFER TYPES

Each of the three active transfer modes (Single, Block, or Demand) can perform three different types of transfers. These are Read, Write and Verify.

Write Transfers

Write transfers move data from an ISA I/O device to memory located on the ISA Bus or in system DRAM. For transfers using compatible timing, PIIX4 will activate ISA Memory control signals to indicate a memory write as soon as the DMA provides the address. The PCI transfer is initiated after the data is valid on the ISA Bus. Data steering is used to steer the data to the correct byte lane during these DMA transfers. When the memory is located on the ISA Bus, a PCI cycle is not initiated.

The DMA device (I/O device) is either an 8- or 16-bit device and is located on the ISA Bus. The DMA device size is fixed for each channel.

Read Transfers

Read transfers move data from ISA memory or the system DRAM, to an ISA I/O device. PIIX4 activates the IOW# command and the appropriate DRAM and ISA Memory control signals to indicate a memory read. Data steering is used to steer the data to the correct byte lane during these DMA transfers. When the cycle involves DRAM, the PCI read transaction is initiated as soon as the DMA address is valid. When the memory is located on the ISA Bus, a PCI cycle is not initiated.

Verify Transfer

Verify transfers are pseudo transfers. The DMA controller generates addresses as in normal read or write transfers. However, PIIX4 does not activate the ISA memory and I/O control lines. Only the DACK lines will go active. PIIX4 asserts the appropriate DACK signal for nine SYSCLKs. If Verify transfers are repeated during Block or Demand DMA requests, each additional pseudo transfer will add eight SYSCLKs. The DACK lines will not be toggled for repeated transfers.

NOTE

Verify transfers are not supported with type F DMA.

8.4.3. DMA TIMINGS

ISA-Compatible timing is provided for ISA DMA slave devices that reside on add in cards. In addition, Type F timing (three SYSCLK period) is provided for motherboard DMA slave and ISA DMA slaves. The Type F timing (along with the 4-byte DMA buffer) is enabled by setting the MBDMAx[FAST] bit.

The repetition rate for ISA-Compatible DMA cycles is eight SYSCLK periods.

The type F cycles occur back to back at a minimum repetition rate of three SYSCLKs (360 ns min). The type F cycles are always performed using the 4-byte DMA buffer.

When PIIX4 negates PHOLD# one clock after driving FRAME# asserted for a bus master IDE transaction or a type F DMA transaction, and another transaction is pending which will cause PIIX4 to acquire the PCI bus, it will drive PHOLD# asserted for the next transaction three clocks after TRDY# is driven negated for the current transaction.

8.4.4. DMA BUFFER FOR TYPE F TRANSFERS

The DMA buffer referred to above is a 4-byte buffer that is used to reduce the PCI utilization resulting from DMA transfers by motherboard devices. The DMA buffer is always used in conjunction with the type F transfers. The type F transfers and the use of the DMA buffer are invoked by setting the MBDMAx[FAST] register bit for the appropriate channel. 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).

8.4.5. DREQ AND DACK# LATENCY CONTROL

The PIIX4 DMA arbiter maintains a minimum DREQ to DACK# latency on all DMA channels when programmed in compatible mode. This is to support older devices such as the 8272A. The DREQs are delayed by eight SYSCLKs prior to being seen by the arbiter logic. This delay guarantees a minimum 1 μsec DREQ to DACK# latency. Software requests will not have this minimum request to DACK# latency. When programmed to operate in type F timing mode (by setting MBDMA[FAST]), the eight SYSCLK latency is not in effect.

8.4.6. CHANNEL PRIORITY

For priority resolution, the DMA consists of two logical channel groups: channels 0–3 and channels 4–7. Each group may be in either fixed or rotate mode, as determined by the DMA Command Register.

DMA I/O slaves normally assert their DREQ line to arbitrate for DMA service. However, a software request for DMA service can be presented through each channel's DMA Request Register. A software request is subject to the same prioritization as any hardware request. Please see the detailed register description for Request Register programming information in the DMA Register description section.

Fixed Priority

The initial fixed priority structure is as follows:

High priority...Low priority
(0, 1, 2, 3) (5, 6, 7)

The fixed priority ordering is 0, 1, 2, 3, 5, 6, and 7. In this scheme, Channel 0 has the highest priority, and channel 7 has the lowest priority. Channels [3:0] of DMA-1 assume the priority position of Channel 4 in DMA-2, thus taking priority over channels 5, 6, and 7.

Rotating Priority

Rotation allows for "fairness" in priority resolution. The priority chain rotates so that the last channel serviced is assigned the lowest priority in the channel group (0–3, 5–7).

Channels 0–3 rotate as a group of four. They are always placed between Channel 5 and Channel 7 in the priority list.

Channel 5–7 rotate as part of a group of four. That is, channels (5–7) form the first three positions in the rotation, while channel group (0–3) comprises the fourth position in the arbitration.

8.4.7. REGISTER FUNCTIONALITY

Please see the "DMA Register description" section, for detailed information on register programming, bit definitions, and default values/functions of the DMA registers after CPURST is valid.

DMA Channel 4 is used to cascade the two DMA controllers together and should not be programmed for any mode other than cascade. The DMA Channel Mode Register for channel 4 will default to cascade mode. Special attention should also be taken when programming the Command and Mask Registers as related to channel 4.

8.4.8. ADDRESS COMPATIBILITY MODE

Whenever the DMA is operating, the addresses do not increment or decrement through the High and Low Page Registers. This is compatible with the 82C37 and Page Register implementation used in the PC-AT. This mode is set after CPURST is valid.

8.4.9. SUMMARY OF DMA TRANSFER SIZES

Table 20 lists each of the DMA device transfer sizes. The column labeled “Current Byte/Word Count Register” indicates that the register contents represents either the number of bytes to transfer or the number of 16-bit words to transfer. The column labeled “Current Address Increment/Decrement” indicates the number added to or taken from the Current Address register after each DMA transfer cycle. The DMA Channel Mode Register determines if the Current Address Register will be incremented or decremented.

Table 20. DMA Transfer Size

DMA Device Date Size and Word Count	Current Byte/Word Count Register	Current Address Increment/Decrement
8-Bit I/O, Count by Bytes	Bytes	1
16-Bit I/O, Count by Words (Address Shifted)	Words	1

8.4.9.1. Address Shifting When Programmed for 16-Bit I/O Count by Words

PIIX4 maintains compatibility with the implementation of the DMA in the PC AT which used the 82C37. The DMA shifts the addresses for transfers to/from a 16-bit device count-by-words. Note that the least significant bit of the Low Page Register is dropped in 16-bit shifted mode. When programming the Current Address Register (when the DMA channel is in this mode), the Current Address must be programmed to an even address with the address value shifted right by 1 bit. The address shifting is as follows:

Table 21. Address Shifting in 16-bit I/O DMA Transfers

Output Address	8-Bit I/O Programmed Address (Ch 0–3)	16-Bit I/O Programmed Address (Ch 5–7) (Shifted)
A0	A0	0
A[16:1]	A[16:1]	A[15:0]
A[23:17]	A[23:17]	A[23:17]

NOTES:
The least significant bit of the Page Register is dropped in 16-bit shifted mode.

8.4.10. AUTOINITIALIZE

By programming a bit in the DMA Channel Mode Register, a channel may be set up as an autoinitialize channel. When a channel undergoes autoinitialization, the original values of the Current Page, Current Address and Current Byte/Word Count Registers are automatically restored from the Base Page, Address, and Byte/Word Count Registers of that channel following TC. The Base Registers are loaded simultaneously with the Current Registers by the microprocessor when the DMA channel is programmed and remain unchanged throughout the DMA service. The mask bit is not set when the channel is in autoinitialize. Following autoinitialize, the channel is ready to perform another DMA service, without CPU intervention, as soon as a valid DREQ is detected.

8.4.11. SOFTWARE COMMANDS

There are three additional special software commands which can be executed by the DMA controller. The three software commands are:

1. Clear Byte Pointer Flip-Flop
2. Master Clear
3. Clear Mask Register

They do not depend on any specific bit pattern on the data bus.

Clear Byte Pointer Flip-Flop

This command is executed prior to writing or reading new address or word count information to/from the DMA controller. This initializes the flip-flop to a known state so that subsequent accesses to register contents by the microprocessor will address upper and lower bytes in the correct sequence.

When the Host CPU is reading or writing DMA registers, two Byte Pointer flip-flops are used; one for channels 0–3 and one for channels 4–7. Both of these act independently. There are separate software commands for clearing each of them (0Ch for channels 0–3, 0D8h for channels 4–7).

DMA Master Clear

This software instruction has the same effect as the hardware reset. The Command, Status, Request, and Internal First/Last Flip-Flop Registers are cleared and the Mask Register is set. The DMA controller will enter the idle cycle.

There are two independent master clear commands; 0Dh which acts on channels 0–3, and 0DAh which acts on channels 4–7.

Clear Mask Register

This command clears the mask bits of all four channels, enabling them to accept DMA requests. I/O port 00Eh is used for channels 0–3 and I/O port 0DCh is used for channels 4–7.

8.4.12. ISA REFRESH CYCLES

Refresh cycle requests are generated by two sources: the refresh controller inside PIIX4 or by ISA bus masters other than the PIIX4. In both cases, the PIIX4 will generate the ISA Memory refresh. The PIIX4 ISA bus controller will enable the address lines SA[7:0] so that 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. PIIX4 maintains a four deep buffer to record internally generated Refresh requests which have not been serviced.

Counter 1 in the timer register set should be programmed to provide a request for refresh about every 15 µs.

PIIX4 Initiated Refresh Cycle

This refresh cycle is initiated by the refresh logic inside PIIX4. PIIX4 asserts REFRESH# to indicate a refresh cycle. PIIX4 then drives the address lines SA[7:0] onto the ISA Bus and generates MEMR# and SMEMR#. PIIX4 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 µs, 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 PIIX4. PIIX4 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 generate a Refresh request and PIIX4's internal refresh request is not serviced within the normal 15 microseconds, a refresh queue counter is incremented. The counter records up to four incomplete refresh cycles, which are all executed as soon as PIIX4 gets the ISA bus.

8.5. PCI DMA

PIIX4 supports two types of PCI DMA protocols: PC/PCI and distributed DMA. They are completely different protocols that are used for different types of peripherals.

PC/PCI DMA uses dedicated REQUEST and GRANT signals to permit PCI devices to request transfers associated with specific DMA channels. Upon receiving a request and getting control of the PCI bus, PIIX4 performs a two-cycle transfer. For example, if data is to be moved from the peripheral to main memory, PIIX4 will first read data from the peripheral and then write it to main memory. The location in main memory is the Current Address Registers in the 8237. PIIX4 supports up to three PC/PCI REQ/GNT pairs.

Distributed DMA is based on monitoring CPU accesses to the 8237. If the accesses are associated with DMA channels that are "distributed" (in some PCI peripheral), then PIIX4 collects or distributes the data before letting the CPU complete its accesses. This way the CPU thinks that it is accessing a standard 8237-based design, even though the registers are not located in PIIX4.

A 16-bit register is included in the PIIX4 Function 0 configuration space at offset 90h. It is divided into seven 2-bit fields that are used to configure the 7 DMA channels.

Each DMA channel can be configured to one of three options:

- Standard ISA (or EIO) DMA using the standard ISA DREQ/DACK signals.
- PC/PCI style DMA using the REQ/GNT signals.
- Distributed DMA.

It is not possible for a particular DMA channel to be configured for more than one style of DMA; however, the seven channels can be programmed independently. For example, channel 3 could be set up for PC/PCI and channel 5 set up for Distributed DMA.

Additional configuration is required separately for the PC/PCI and Distributed DMA functions and is described below.

8.5.1. PC/PCI DMA

PIIX4 provides support for DMA across PCI using the PC/PCI DMA Protocol. The PCI DMA request/grant pairs, REQ[A:C]# and GNT[A:C]#, can be configured for support of a PC/PCI DMA Expansion agent. The PCI DMA Expansion agent can then provide DMA service or ISA Bus Master service using the PIIX4 DMA controller. The REQ#/GNT# pair must follow the PC/PCI serial protocol described below.

PCI DMA Expansion Protocol

The PCI expansion agent must support the PCI expansion Channel Passing Protocol defined in Figure 4 for both the REQ# and GNT# pins.

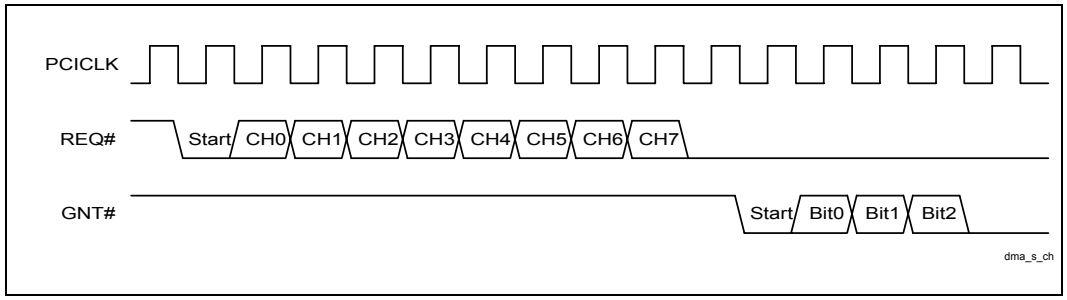


Figure 4. DMA Serial Channel Passing Protocol

The requesting device must encode the channel request information as shown above, where CH0–CH7 are one clock active high states representing DMA channel requests 0–7.

PIIX4 encodes the granted channel on the GNT# line as shown above, where the bits have the same meaning as shown in the Figure 4. For example, the sequence [start, bit 0, bit 1, bit 2]=[0,1,0,0] grants DMA channel 1 to the requesting device, and the sequence [start, bit 0, bit 1, bit 2]=[0,0,1,1] grants DMA channel 6 to the requesting device.

All PCI DMA expansion agents must use the channel passing protocol described above. They must also work as follows:

1. If a PCI DMA expansion agent has more than one request active, it must resend the request serial protocol after one of the requests has been granted the bus and it has completed its transfer. The expansion device should drive its REQ# inactive for two clocks and then transmit the serial channel passing protocol again, even if there are no new requests from the PCI expansion agent to PIIX4. For example: If a PCI expansion agent had active requests for DMA channel 1 and channel 5, it would pass this information to PIIX4 through the expansion channel passing protocol. If after receiving GNT# (assume for CH5) and having the device finish its transfer (device stops driving request to PCI expansion agent) it would then need to re-transmit the expansion channel passing protocol to inform PIIX4 that DMA channel 1 was still requesting the bus, even if that was the only request the expansion device had pending.
2. If a PCI DMA expansion agent has a request go inactive before PIIX4 asserts GNT#, it must resend the expansion channel passing protocol to update PIIX4 with this new request information. For example: If a PCI expansion agent has DMA channel 1 and 2 requests pending it will send them serially to PIIX4 using the expansion channel passing protocol. If, however, DMA channel 1 goes inactive into the expansion agent before the expansion agent receives a GNT# from PIIX4, the expansion agent MUST pull its REQ# line high for one clock and resend the expansion channel passing information with only DMA channel 2 active. Note that PIIX4 does not do anything special to catch this case because a DREQ going inactive before a DACK# is received is not allowed in the ISA DMA protocol and, therefore, does not need to work properly in this protocol either. This requirement is needed to be able to support Plug-n-Play ISA devices that toggle DREQ# lines to determine if those lines are free in the system.
3. If a PCI expansion agent has sent its serial request information and receives a new DMA request before receiving GNT# the agent must resend the serial request with the new request active. For example: If a PCI expansion agent has already passed requests for DMA channel 1 and 2 and sees DREQ 3 active before a GNT is received, the device must pull its REQ# line high for one clock and resend the expansion channel passing information with all three channels active.

The three cases above require the following functionality in the PCI DMA expansion device:

1. Drive REQ# inactive for one clock to signal new request information.
2. Drive REQ# inactive for two clocks to signal that a request that had been granted the bus has gone inactive.
3. The REQ# and GNT# state machines must run independently and concurrently (i.e., a GNT# could be received while in the middle of sending a serial REQ# or a GNT# could be active while REQ# is inactive).

PCI DMA Expansion Cycles

PIIX4's support of the Mobile PC/PCI DMA Protocol currently consists of four types of cycles: Memory to I/O, I/O to Memory, Verify, and ISA Master cycles. ISA Masters are supported through the use of a DMA channel that has been programmed for cascade mode. Single Transfer Mode is implicitly supported as the case where the DMA controller negates the DACK#/GNT# signal after one transfer has been completed or the DMA controller toggles DACK# after every transfer. Single transfer mode does not require the requesting device to negate DREQ# after a cycle has completed. Therefore, a PCI DMA agent that uses this mode must also sample the GNT# signal and remove DACK# to the I/O DMA device when GNT# goes inactive.

The DMA controller does a two cycle transfer (a load followed by a store) as opposed to the ISA "fly-by" cycle for PC/PCI DMA agents. The memory portion of the cycle generates a PCI memory read or memory write bus cycle, its address representing the selected memory.

The I/O portion of the DMA cycle generates a PCI I/O cycle to one of four I/O addresses (Table 22). Note that these cycles must be qualified by an active GNT# signal to the requesting device.

Table 22. DMA Cycle vs. I/O Address

DMA Cycle Type	DMA I/O Address	TC (A2)	PCI Cycle Type
Normal	00h	0	I/O Read/Write
Normal TC	04h	1	I/O Read/Write
Verify	0C0h	0	I/O Read
Verify TC	0C4h	1	I/O Read

For PCI DMA cycles, the I/O address indicates the type of DMA cycle taking place (whether its a normal or a verify cycle, and if this is the last transfer of the buffer). Note that the A2 address line is encoded as the terminal count signal for PCI cycles; A2 asserted during a PCI I/O cycle indicates the last transfer in the current DMA buffer. To ensure that non Mobile PC/PCI compliant PCI I/O devices do not confuse Mobile PC/PCI DMA cycles for normal I/O cycles, the addresses used by the PCI DMA cycles correspond to the slave addresses of the Mobile PC/PCI DMA controller.

All PCI DMA I/O ports must be DWord aligned and can be either byte or word in size. This means that any PCI DMA I/O port must always be connected to the lower data lines of the PCI data bus (Table 23).

The byte enables also reflect this during the I/O portion of a PCI DMA cycle. Table 24 illustrates the byte enable state for any given PCI DMA cycle:

Table 23. PCI Data Bus vs. DMA I/O port size

PCI DMA I/O Port Size	PCI Data Bus Connection
Byte	AD[7:0]

Table 23. PCI Data Bus vs. DMA I/O port size

PCI DMA I/O Port Size	PCI Data Bus Connection
Word	AD[15:0]

Table 24. DMA I/O Cycle Width vs. BE[3:0]#

BE[3:0]#	Description
1110b	8-bit DMA I/O Cycle
1100b	16-bit DMA I/O Cycle

NOTES:
For verify cycles the value of the Byte Enables (BEs) is a “don’t care.”

Every DMA device (including Secondary Bus Arbiters) must recognize a valid signal on its GNT# combined with the DMA I/O address as its command authorization to initiate a DMA access cycle. PIIX4 is required to assert the DMA I/O device’s GNT# signal until the data phase of the I/O portion of the DMA transfer.

8.5.2. DISTRIBUTED DMA

The Distributed DMA (DDMA) scheme is based on the concept that the registers associated with individual DMA can physically reside outside of PIIX4, specifically on other PCI devices. The Distributed DMA logic in PIIX4 is only used when the CPU does accesses to the 8237 registers. Data movement is the responsibility of the peripheral.

Separate algorithms are followed depending whether the CPU attempts a read cycle or write cycle. Each is covered separately. PIIX4 is able to determine if a particular DMA channel is “distributed” based on the PCI configuration space.

Additional Configuration

PIIX4 contains two registers to indicate the I/O locations for the relocated DMA registers for the DDMA peripherals. The first register indicates the offset of the registers associated with DMA channels 0–3. The second indicates the offset of the registers associated with DMA channels 5–7. Channel 4 is assumed to be unavailable. It is up to the BIOS or other configuration software to program the DDMA peripherals to the corresponding locations.

Read/Write Cycles Protocol

For read cycles on the PCI bus that correspond to distributed DMA channels, PIIX4 performs the following:

- PIIX4 issues a PCI retry to terminate this cycle.
- PIIX4 requests the PCI bus. Upon being granted access to the bus, PIIX4 performs one or more read cycles to the 8237 and/or the peripherals. The I/O location of the read cycle is calculated based on several parameters: the DDMA Base Pointer registers in the PCI Configuration space, the DMA channel number (0–3, 5–7), and the register location (0h–Fh).
- PIIX4 uses the data obtained via the read cycles (along with the value in the 8237) to construct the proper data value.
- When the CPU retries the cycle, PIIX4 responds with the proper data value.

The specific number of read cycles and merging format for each of the 8237 registers is covered in Table 25.

For write cycles on the PCI bus that correspond to distributed DMA channels, PIIX4 performs the following:

- PIIX4 issues a PCI retry to terminate this cycle.
- PIIX4 requests the PCI bus. Upon being granted access to the bus, PIIX4 performs one or more write cycles to the 8237 and/or the peripherals. The I/O location of the write cycle is calculated based on several parameters: the DDMA Base Pointer registers in the PCI Configuration space, the DMA channel number (0–3, 5–7), and the register location (0h–Fh).
- PIIX4 uses the data obtained via the CPU's original write cycles to determine the proper values to write to the peripherals and to the 8237.
- When the CPU retries the cycle, PIIX4 lets it complete normally.

Calculating the I/O Address

When PIIX4 attempts to access the PCI peripheral, it performs I/O read or write cycles. The exact address to use is calculated as follows:

Bits [31:16] are 0.

Bits [15:6] are indicated by the Base Pointer in the PCI Configuration Space for Function 0. The base pointer at offset 92h is used for DMA channels 0–3. The base pointer at offset 94h is used for DMA channels 5–7.

Bits [5:4] are determined by the DMA channel number being accessed.

DMA Channel Number	Bits [5:4]
0	00
1 or 5	01
2 or 6	10
3 or 7	11

Bits [3:0] are determined by the register being accessed.

NOTE

The mapping in the peripheral is Not the same as in the 8237. Table 25 shows the mapping of the 8237 register to the Distributed DMA peripheral.

Table 25. 8237 Register Map to Distributed DMA Peripheral

I/O Address	8237 F/F Status	R/W	8237 Register Name	"Distributed" Cycle I/O Address
0, 2, 4, or 6h, C4, C8, or CCh	0	W	Base Address Register A[0:7]	Base Pointer + channel # + 0h
0, 2, 4, or 6h, C4, C8, or CCh	0	R	Current Address Register A[0:7]	Base Pointer + channel # + 0h
0, 2, 4, or 6h, C4, C8, or CCh	1	W	Base Address Register A[8:15]	Base Pointer + channel # + 1h
0, 2, 4, or 6h, C4, C8, or CCh	1	R	Current Address Register A[8:15]	Base Pointer + channel # + 1h
87h, 83, 81, 82h 8B, 89, 8Ah	X	R/W	Page Register	Base Pointer + channel # + 2h
1, 3, 5, or 7h, C6, CA, CEh	0	W	Base Word Count Register D[0:7]	Base Pointer + channel # + 4h
1, 3, 5, or 7h, C6, CA, CEh	0	R	Current Word Count Register D[0:7]	Base Pointer + channel # + 4h
1, 3, 5, or 7h, C6, CA, CEh	1	W	Base Word Count Register D[8:15]	Base Pointer + channel # + 5h
1, 3, 5, or 7h, C6, CA, CEh	1	R	Current Word Count Register D[8:15]	Base Pointer + channel # + 5h
08h D0h	X	W	Command Register	Base Pointer + channel # + 8h
08h D0h	X	R	Status Register	Base Pointer + channel # + 8h
09h D2h	X	W	Request Register	Base Pointer + channel # + 9h
0Bh D6h	X	W	Mode Register	Base Pointer + channel # + Bh
0Dh DAh	X	W	Master Clear	Base Pointer + channel # + Dh
0Fh DEh	X	W	Write All Masks Register	Base Pointer + channel # + Fh
Ah D4h	X	W	Single Channel Mask	See comments below.
0Eh DCh	X	W	Clear Mask Register	See comments below.

Single Channel Mask Register

To make the peripherals easier to implement, the Distributed DMA specification does not have the peripherals implement the Single-Channel Mask Registers. Instead, a write to the Single Channel Mask register (which encodes the channel number in the low two bits) causes write the Write All Masks Register (which has a separate mask bit for each channel). The Distributed DMA peripheral uses bit 0 in the Write All Masks Register for that particular channel.

Thus, when a write occurs to the Single Channel Mask register, PIIX4 examines the low two data bits to determine the DMA channel number. This causes a write the peripheral at Base Pointer + channel # + Fh. The data value (bit 0) for that write cycle is determined by data bit 2 of the original CPU write.

Clear Mask Register

To make the peripherals easier to implement, the Distributed DMA specification does not have the peripherals implement the Clear Mask Command. Instead, a write to the Clear Mask Command register (which has a don't care data value causes writes to all the distributed channels associated with that 8237).

Thus, when a write occurs to the Clear Mask Command register, PIIX4 performs up to 4 writes to the Write All Masks register (Base Pointer + channel # + Fh) with a data value of 0h.

If another PCI master attempts to read or write to one of the DMA controller's registers while a Distributed DMA cycle is in progress, that cycle will be retried until the PC DMA protocol completes. This prevents two outstanding PC/PCI requests.

8.6. Interrupt Controller

PIIX4 provides an ISA-Compatible interrupt controller that incorporates the functionality of two 82C59 interrupt controllers (Figure 5). The two controllers are cascaded, providing 13 external and three internal interrupts. The master interrupt controller provides IRQ [7:0] and the slave interrupt controller provides IRQ [15:8]. The three internal interrupts are used for internal functions only. IRQ0 is available to the user only when an external IO APIC is enabled. IRQ2 is used to cascade the two controllers and is not available to the user. IRQ0 is used as a system timer interrupt and is tied to Interval Timer 1, Counter 0. 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. PIIX4 can be programmed to allow the four PCI active low interrupts (PIRQ[A:D]#) to be internally routed to one of 11 interrupts (IRQ[15:14,12:9,7:3]).

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[0:15]) 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 PIIX4. IRQ12/M is generated internally (as part of the mouse support) when bit-4 in the XBCS is set to a 1. When set to a 0, the standard IRQ12 function is provided and IRQ12 appears externally.

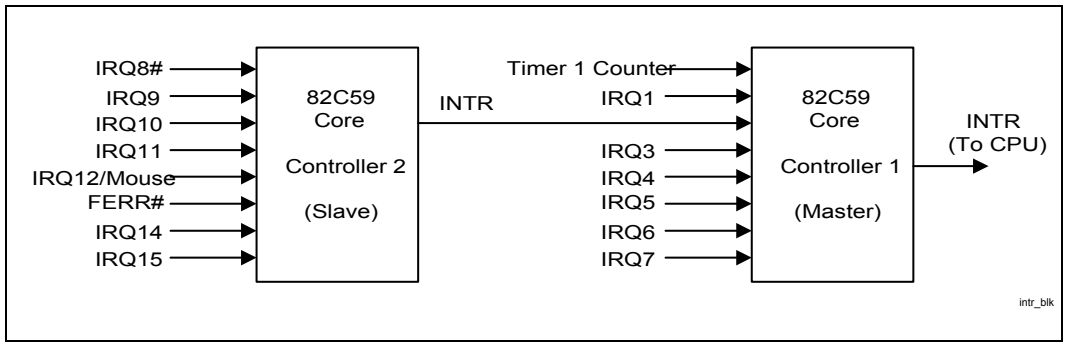


Figure 5. Interrupt Controller Block Diagram

8.6.1. PROGRAMMING THE INTERRUPT CONTROLLER

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

Initialization Command Words (ICWs)

Before normal operation can begin, each Interrupt Controller in the system must be initialized. In the 82C59, this is a 2- to 4-byte sequence. However, for PIIX4, each controller must be initialized with a 4-byte sequence. This 4-byte sequence is required to configure the interrupt controller correctly for PIIX4 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 PIIX4-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.

ICW1 starts the initialization sequence. ICW2 is programmed to provide bits [7:3] of the interrupt vector that will be released onto the data bus by the interrupt controller during an interrupt acknowledge. A different base [7:3] is selected for each interrupt controller. ICW3 is programmed differently for CNTRL-1 and CNTRL-2, and has a different meaning for each controller.

For CNTRL-1, the master controller, ICW3 is used to indicate which IRQx input line is used to cascade CNTRL-2, the slave controller. Within the PIIX4 interrupt unit, IRQ2 on CNTRL-1 is used to cascade the INTR output of CNTRL-2. Consequently, bit 2 of ICW3 on CNTRL-1 is set to a 1, and the other bits are set to 0’s.

For CNTRL-2, ICW3 is the slave identification code used during an interrupt acknowledge cycle. CNTRL-1 broadcasts a code to CNTRL-2 over three internal cascade lines if an IRQ[x] line from CNTRL-2 won the priority arbitration on the master controller and was granted an interrupt acknowledge by the CPU. CNTRL-2 compares this identification code to the value stored in ICW3, and if the code is equal to bits [2:0] of ICW3, CNTRL-2 assumes responsibility for broadcasting the interrupt vector during the second interrupt acknowledge cycle pulse.

ICW4 must be programmed on both controllers. At the very least, bit 0 must be set to a 1 to indicate that the controllers are operating in an Intel Architecture-based system.

Operation Command Words (OCWs)

These are the command words which 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 will mask 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 is used to set up reads of the ISR and IRR, to enable or disable the Special Mask Mode (SMM), and to set up the interrupt controller in polled interrupt mode. The OCWs can be written into the Interrupt Controller any time after initialization.

8.6.2. END-OF-INTERRUPT OPERATION

End of Interrupt (EOI)

The In Service (IS) bit can be set to 0 automatically following the trailing edge of the second INTA# pulse (when AEOI bit in ICW1 is set to 1) or by a command word that must be issued to the Interrupt Controller before returning from a service routine (EOI command). An EOI command must be issued twice with this cascaded interrupt controller configuration, once for the master and once for the slave.

There are two forms of EOI commands: Specific and Non-Specific. When the Interrupt Controller is operated in modes that preserve the fully nested structure, it can determine which IS bit to set to 0 on EOI. When a Non-Specific EOI command is issued, the Interrupt Controller automatically sets to 0 the highest IS bit of those that are set to 1, since in the fully nested mode the highest IS level was necessarily the last level acknowledged and serviced. A Non-Specific EOI can be issued with OCW2 (EOI=1, SL=0, R=0).

When a mode is used that may disturb the fully nested structure, the Interrupt Controller may no longer be able to determine the last level acknowledged. In this case a Specific End of Interrupt must be issued that includes as part of the command the IS level to be reset. A specific EOI can be issued with OCW2 (EOI=1, SL=1, R=0, and LO–L2 is the binary level of the IS bit to be set to 0).

Note that an IS bit that is masked by an IMR bit will not be cleared by a non-specific EOI if the Interrupt Controller is in the Special Mask Mode.

Automatic End of Interrupt (AEOI) Mode

If AEOI=1 in ICW4, then the Interrupt Controller operates in AEOI mode continuously until reprogrammed by ICW4. Note that reprogramming ICW4 implies that ICW1, ICW2, and ICW3 must be reprogrammed first, in sequence. In this mode, the Interrupt Controller automatically performs a Non-Specific EOI operation at the trailing edge of the last interrupt acknowledge pulse. Note that from a system standpoint, this mode should be used only when a nested multi-level interrupt structure is not required within a single Interrupt Controller. The AEOI mode can only be used in a master Interrupt Controller and not a slave (on CNTRL-1 but not CNTRL-2).

8.6.3. MODES OF OPERATION

Fully Nested Mode

This mode is entered after initialization unless another mode is programmed. The interrupt requests are ordered in priority from 0 through 7 (0 being the highest). When an interrupt is acknowledged, the highest priority request is determined and its vector placed on the bus. Additionally, a bit of the Interrupt Service Register (IS[0:7]) is set. This IS bit remains set until the microprocessor issues an End of Interrupt (EOI) command immediately before returning from the service routine. Or, if the AEOI (Automatic End of Interrupt) bit is set, this IS bit remains set until the trailing edge of the second INTA#. While the IS bit is set, all further interrupts of the same or lower priority are inhibited, while higher levels will generate an interrupt (which will be acknowledged only if the microprocessor internal interrupt enable flip-flop has been re-enabled through software).

After the initialization sequence, IRQ0 has the highest priority and IRQ7 the lowest. Priorities can be changed, as will be explained, in the rotating priority mode.

The Special Fully Nested Mode

This mode will be used in the case of a system where cascading is used, and the priority has to be conserved within each slave. In this case, the special fully nested mode will be programmed to the master (using ICW4). This mode is similar to the normal nested mode with the following exceptions:

When an interrupt request from a certain slave is in service, this slave is not locked out from the master's priority logic and further interrupt requests from higher priority IRQs within the slave will be recognized by the master and will initiate interrupts to the processor. (In the normal nested mode, a slave is masked out when its request is in service and no higher requests from the same slave can be serviced.)

When exiting the Interrupt Service routine, the software has to check whether the interrupt serviced was the only one from that slave. This is done by sending a Non-Specific End of Interrupt (EOI) command to the slave and then reading its In-Service Register and checking for zero. If it is empty, a Non-Specific EOI can be sent to the master too. If not, no EOI should be sent.

Automatic Rotation (Equal Priority Devices)

In some applications, there are a number of interrupting devices of equal priority. Automatic rotation mode provides for a sequential 8-way rotation. In this mode, a device receives the lowest priority after being serviced. In the worst case, a device requesting an interrupt will have to wait until each of seven other devices are serviced at most once.

There are two ways to accomplish automatic rotation using OCW2; the Rotation on Non-Specific EOI Command (R=1, SL=0, EOI=1) and the Rotate in Automatic EOI Mode which is set by (R=1, SL=0, EOI=0) and cleared by (R=0, SL=0, EOI=0).

Specific Rotation (Specific Priority)

The programmer can change priorities by programming the bottom priority and thus fixing all other priorities. For example, if IRQ5 is programmed as the bottom priority device, IRQ6 will be the highest priority device.

The Set Priority Command is issued in OCW2 where: R=1, SL=1; LO–L2 is the binary priority level code of the bottom priority device. See the register description for the bit definitions.

Note that, in this mode, internal status is updated by software control during OCW2. However, it is independent of the End of Interrupt (EOI) command (also executed by OCW2). Priority changes can be executed during an EOI command by using the Rotate on Specific EOI Command in OCW2 (R=1, SL=1, EOI=1 and LO–L2=IRQ level to receive bottom priority).

Poll Command

The Polled Mode can be used to conserve space in the interrupt vector table. Multiple interrupts that can be serviced by one interrupt service routine do not need separate vectors if the service routine uses the poll command.

The Polled Mode can also be used to expand the number of interrupts. The polling interrupt service routine can call the appropriate service routine, instead of providing the interrupt vectors in the vector table.

In this mode, the INTR output is not used and the microprocessor internal Interrupt Enable flip-flop is reset, disabling its interrupt input. Service to devices is achieved by software using a Poll Command.

The Poll command is issued by setting P=1 in OCW3. The Interrupt Controller treats the next I/O read pulse to the Interrupt Controller as an interrupt acknowledge, sets the appropriate IS bit if there is a request, and reads the priority level. Interrupts are frozen from the I/O write to the I/O read.

This mode is useful if there is a routine command common to several levels so that the INTA# sequence is not needed (saves ROM space).

8.6.4. CASCADE MODE

The Interrupt Controllers in PIIX4 are interconnected in a cascade configuration with one master and one slave. This configuration can handle up to 15 separate priority levels.

The master controls the slaves through a three line internal cascade bus. When the master drives 010b on the cascade bus, this bus acts like a chip select to the slave controller.

In a cascade configuration, the slave interrupt outputs are connected to the master interrupt request inputs. When a slave request line is activated and afterwards acknowledged, the master enables the corresponding slave to release the interrupt vector address during the second INTA# cycle of the interrupt acknowledge sequence.

Each Interrupt Controller in the cascaded system must follow a separate initialization sequence and can be programmed to work in a different mode. An EOI Command must be issued twice: once for the master and once for the slave.

8.6.5. EDGE AND LEVEL TRIGGERED MODE

In ISA systems this mode is programmed using bit 3 in ICW1. With PIIX4, 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 active until after the falling edge of the first INTA#. 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.

8.6.6. INTERRUPT MASKS

Masking on an Individual Interrupt Request Basis

Each interrupt request input can be masked individually by the Interrupt Mask Register (IMR). This register is programmed through OCW1. Each bit in the IMR masks one interrupt channel, if it is set to a 1. Bit 0 masks IRQ0, Bit 1 masks IRQ1 and so forth. Masking an IRQ channel does not affect the other channel's operation, with one exception. Masking IRQ2 on CNTRL-1 will mask off all requests for service from CNTRL-2. The CNTRL-2 INTR output is physically connected to the CNTRL-1 IRQ2 input.

Special Mask Mode

Some applications may require an interrupt service routine to dynamically alter the system priority structure during its execution under software control. For example, the routine may wish to inhibit lower priority requests for a portion of its execution but enable some of them for another portion.

The difficulty is that if an Interrupt Request is acknowledged and an End of Interrupt command did not reset its IS bit (i.e., while executing a service routine), the Interrupt Controller would have inhibited all lower priority requests with no easy way for the routine to enable them.

The Special Mask Mode enables all interrupts not masked by a bit set in the Mask Register. Interrupt service routines that require dynamic alteration of interrupt priorities can take advantage of the Special Mask Mode. For example, a service routine can inhibit lower priority requests during a part of the interrupt service, then enable some of them during another part.

In the Special Mask Mode, when a mask bit is set to 1 in OCW1, it inhibits further interrupts at that level and enables interrupts from all other levels (lower as well as higher) that are not masked.

Thus, any interrupts may be selectively enabled by loading the Mask Register with the appropriate pattern.

Without Special Mask Mode, if an interrupt service routine acknowledges an interrupt without issuing an EOI to clear the IS bit, the interrupt controller inhibits all lower priority requests. The Special Mask Mode provides an easy way for the interrupt service routine to selectively enable only the interrupts needed by loading the Mask register.

The special Mask Mode is set by OCW3 where: SSMM=1, SMM=1, and cleared where SSMM=1, SMM=0.

8.6.7. READING THE INTERRUPT CONTROLLER STATUS

The input status of several internal registers can be read to update the user information on the system. The Interrupt Request Register (IRR) and In-Service Register (ISR) can be read via OCW3. The Interrupt Mask Register (IMR) is read via a read of OCW1. Brief descriptions of the ISR, the IRR, and the IMR follow.

- **Interrupt Request Register (IRR):** 8-bit register which contains the status of each interrupt request line. Bits that are clear indicate interrupts that have not requested service. The Interrupt Controller clears the IRR's highest priority bit during an interrupt acknowledge cycle. (Not affected by IMR).
- **In-Service Register (ISR):** 8-bit register indicating the priority levels currently receiving service. Bits that are set indicate interrupts that have been acknowledged and their interrupt service routine started. Bits that are cleared indicate interrupt requests that have not been acknowledged, or interrupt request lines that have not been asserted. Only the highest priority interrupt service routine executes at any time. The lower priority interrupt services are suspended while higher priority interrupts are serviced. The ISR is updated when an End of Interrupt Command is issued.
- **Interrupt Mask Register (IMR):** 8-bit register indicating which interrupt request lines are masked.

The IRR can be read when, prior to the I/O read cycle, a Read Register Command is issued with OCW3 (RR=1, RIS=0). The ISR can be read when, prior to the I/O read cycle, a Read Register Command is issued with OCW3 (RR=1, RIS=1).

The interrupt controller retains the ISR/IRR status read selection following each write to OCW3. Therefore, there is no need to write an OCW3 before every status read operation, as long as the current status read corresponds to the previously selected register. For example, if the ISR is selected for status read by an OCW3 write, the ISR can be read over and over again without writing to OCW3 again. However, to read the IRR, OCW3 will have to be reprogrammed for this status read prior to the OCW3 read to check the IRR. This is not true when poll mode is used. Polling Mode overrides status read when P=1, RR=1 in OCW3.

After initialization the Interrupt Controller is set to read the IRR.

As stated, OCW1 is used for reading the IMR. The output data bus will contain the IMR status whenever I/O read is active the address is 021h or 061h (OCW1).

8.6.8. INTERRUPT STEERING

PIIX4 can be programmed to allow four PCI programmable interrupts (PIRQ[A:D]#) to be internally routed to one of 11 interrupts IRQ[15,14,12:9,7:3]. PCLK is used to synchronize the PIRQx# inputs. The PIRQx# lines are run through an internal multiplexer that assigns, or routes, an individual PIRQx# line to any one of 11 IRQ 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.

Bits [3:0] in each PIRQx Route Control register are used to route the associated PIRQx# line to an internal IRQ input. Bit 7 in each register is used to disable routing of the associated PIRQx#.

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 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.

8.7. Serial Interrupts

PIIX4 supports a serial IRQ scheme. This allows a single signal to be used to report ISA-style interrupt requests. Typically, it will be used in a mobile environment by docking bridges or Cardbus controllers.

Because more than one device may need to share the single serial IRQ signal, an Open Collector signaling scheme is used. Timing is based on the PCI Clock. If the PCI Clock is inactive when a device needs to signal an interrupt, the CLKRUN# signal must first be asserted by the device to restart the PCI Clock. The serial IRQ configuration is handled via the PCI configuration space. No other registers are associated with the scheme.

8.7.1. PROTOCOL

Serial interrupt information is transferred using three types of frames: a Start frame, one or more IRQ Data frames, and one Stop frame. There are also two modes of operation: Quiet Mode and Continuous Mode.

Quiet (Active) Mode

To indicate an interrupt, the peripheral brings the SERIRQ signal active for one clock, and then tri-states the signal. This brings all the state machines from IDLE to the ACTIVE states.

PIIX4 then takes control of the SERIRQ signal by driving it low on the next clock, and continues driving it low for 3–7 clocks more (programmable). Thus, the total number of clocks low will be 4–8. After those clocks, PIIX4 drives SERIRQ high for one clock and then tri-state the signal.

Continuous (Idle) Mode

In this mode, PIIX4 initiates the Start frame, rather than the peripherals. Typically, this is done to update IRQ status (acknowledges). PIIX4 drives SERIRQ low for 4–8 clocks. This is the default mode after reset, and can be used to enter the Quiet mode.

Data Frame

Once the Start frame has been initiated, all of the serial interrupt peripherals must start counting frames based on the rising edge of SERIRQ. Each of the IRQ/DATA frames has exactly three phases of one clock each: a Sample phase, a Recovery phase, and a Turn-around phase.

During the Sample phase, the device drives SERIRQ low if the corresponding interrupt signal should be active. If the corresponding interrupt is inactive, then the devices should not drive the SERIRQ signal. It will remain high due to pull-up resistors. During the other two phases (Turn around and Recovery), no device should drive the SERIRQ signal. The IRQ/DATA frames have a specific order and usage, as shown in Table 26.

If an SMI# is activated on frame 3, PIIX4 drives its EXTSMI# signal active. This then generates an SMI# to the microprocessor, if enabled.

Table 26. SERIRQ Frames

Data Frame Number	Usage	# Clocks Past Start
1	UNASSIGNED	2
2	IRQ1	5
3	SMI#	8
4	IRQ3	11
5	IRQ4	14
6	IRQ5	17
7	IRQ6	20
8	IRQ7	23
9	UNASSIGNED	26
10	IRQ9	29
11	IRQ10	32
12	IRQ11	35
13	IRQ12	38
14	UNASSIGNED	41
15	IRQ14	44
16	IRQ15	47
17	IOCHCK#	50
18	PCI INTA#	53
19	PCI INTB#	56
20	PCI INTC#	59
21	PCI INTD#	62
32:22	UNASSIGNED	96

Stop Frame

After all of the data frames, a Stop frame is performed done by PIIX4. This is accomplished by making SERIRQ low for 2–3 clocks. The number of clocks determines the next mode:

If SERIRQ is low for 2 clocks, the next mode is the Quite Mode. Any device may initiate a Start frame in the second clock (or more) after the rising edge of the Stop frame.

If SERIRQ is low for 3 clocks, the next mode is the Continuous mode. Only PIIX4 may initiate a Start frame in the second clock (or more) after the rising edge of the Stop frame.

8.8. Timer/Counters

PIIX4 contains three counters that are equivalent to those found in the 82C54 programmable interval timer. The three counters are contained in one PIIX4 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 one 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 (see NMI Status and Control ports).

8.8.1. PROGRAMMING THE INTERVAL TIMER

The counter/timers are programmed by I/O accesses and are addressed as though they are contained in one 82C54 interval timer. A single Control Word Register controls the operation of all three counters. The interval timer is an I/O-mapped device. Several commands are available:

The Control Word Command specifies:

- which counter to read or write
- the operating mode
- the count format (binary or BCD)

The Counter Latch Command latches the current count so that it can be read by the system. The countdown process continues. The Read Back Command reads the count value, programmed mode, the current state of the OUT pins, and the state of the Null Count Flag of the selected counter.

The Read/Write Logic selects the Control Word Register during an I/O write when address lines A[1:0]=11. This condition occurs during an I/O write to port address 043h, the address for the Control Word Register on Timer 1. If the CPU writes to port 043h, the data is stored in the Control Word Register and is interpreted as the Control Word used to define the operation of the Counters.

The Control Word Register is write only. Counter status information is available with the read back Command.

Because the timer counters wake up in an unknown state after power up, multiple refresh requests may be queued. To avoid possible multiple refresh cycles after power up, program the timer counter immediately after power up.

Write Operations

Programming the interval timer is a simple process:

1. Write a control word.
2. Write an initial count for each counter.
3. Load the least and/or most significant bytes (as required by Control Word bits 5, 4) of the 16-bit counter.

The programming procedure for the PIIX4 timer is very flexible. Only two conventions need to be observed. First, for each counter, the control word must be written before the initial count is written. Second, the initial count must follow the count format specified in the control word (least significant byte only, most significant byte only, or least significant byte and then most significant byte).

Since the Control Word Register and the three counters have separate addresses (selected by the A1, A0 inputs), and each control word specifies the counter it applies to (SC0, SC1 bits), no special instruction sequence is required. Any programming sequence that follows the conventions above is acceptable.

A new initial count may be written to a counter at any time without affecting the counter's programmed mode. Counting will be affected as described in the mode definitions. The new count must follow the programmed count format.

If a counter is programmed to read/write 2-byte counts, the following precaution applies: A program must not transfer control between writing the first and second byte to another routine which also writes into that same counter. Otherwise, the counter will be loaded with an incorrect count.

Interval Timer Control Word Format

The control word specifies the counter, the operating mode, the order and size of the count value, and whether it counts down in a 16-bit or binary-coded decimal (BCD) format. After writing the control word, a new count may be written at any time. The new value will take effect according to the programmed mode.

If a counter is programmed to read/write 2-byte counts, the following precaution applies: A program must not transfer control between writing the first and second byte to another routine which also writes into that same counter. Otherwise, the counter will be loaded with an incorrect count. The count must always be completely loaded with both bytes.

Read Operations

It is often desirable to read the value of a counter without disturbing the count in progress. There are three possible methods for reading the counters: a simple read operation, the Counter Latch Command, and the Read Back Command.

Counter I/O Port Read

The first method is to perform a simple read operation. To read the counter, which is selected with the A1, A0 inputs (port 040h, 041h, or 042h), the CLK input of the selected counter must be inhibited by using either the GATE input or external logic. Otherwise, the count may be in the process of changing when it is read, giving an undefined result. When reading the count value directly, follow the format programmed in the control register: read LSB, read MSB, or read LSB then MSB. Within the PIIX4 timer unit, the GATE input on Counter 0 and Counter 1 is tied high. Therefore, the direct register read should not be used on these two counters. The GATE input of Counter 2 is controlled through I/O port 061h. If the GATE is disabled through this register, direct I/O reads of port 042h will return the current count value.

Counter Latch Command

The Counter Latch Command latches the count at the time the command is received. This command is used to ensure that the count read from the counter is accurate (particularly when reading a 2-byte count). The count value is then read from each counter's Count Register as was programmed by the Control Register.

The selected counter's output latch (OL) latches the count at the time the Counter Latch Command is received. This count is held in the latch until it is read by the CPU (or until the Counter is reprogrammed). The count is then unlatched automatically and the OL returns to "following" the counting element (CE). This allows reading the contents of the counters "on the fly" without affecting counting in progress. Multiple Counter Latch Commands may be used to latch more than one counter. Each latched counter's OL holds its count until it is read. Counter Latch Commands do not affect the programmed mode of the counter in any way. The Counter Latch Command can be used for each counter in the PIIX4 timer unit.

If a Counter is latched 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.

With either method, the count must be read according to the programmed format; specifically, if the counter is programmed for 2-byte counts, 2 bytes must be read. The 2 bytes do not have to be read one right after the other. Read, write, or programming operations for other counters may be inserted between them.

Another feature of the PIIX4 timer is that reads and writes of the same counter may be interleaved. For example, if the Counter is programmed for 2-byte counts, the following sequence is valid:

- Read least significant byte.
- Write new least significant byte.
- Read most significant byte.
- Write new most significant byte.

If a counter is programmed to read/write 2-byte counts, a program must not transfer control between reading the first and second byte to another routine which also reads from that same counter. Otherwise, an incorrect count will be read.

Read Back Command

The third method uses the 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 Control Word Register, which causes the current states of the above mentioned variables to be latched. The value of the counter and its status may then be read by I/O access to the counter address.

The Read Back Command may be used to latch multiple counter output latches (OL) by setting the COUNT# bit D5=0 and selecting the desired counter(s). This single command is functionally equivalent to several counter latch commands, one for each counter latched. Each counter's latched count is held until it is read (or

the counter is reprogrammed). Once read, a counter is automatically unlatched. The other counters remain latched until they are read. If multiple count Read Back Commands are issued to the same counter without reading the count, all but the first are ignored (i.e. the count which will be read is the count at the time the first Read Back Command was issued).

The Read Back Command may also be used to latch status information of selected counter(s) by setting STATUS# bit D4=0. Status must be latched to be read. The status of a counter is accessed by a read from that counter's I/O port address.

If multiple counter status latch operations are performed without reading the status, all but the first are ignored. The status returned from the read is the counter status at the time the first status Read Back Command was issued.

Both count and status of the selected counter(s) may be latched simultaneously by setting both the COUNT# and STATUS# bits [5:4]=00. This is functionally the same as issuing two consecutive, separate Read Back Commands. The above discussions apply here also. Specifically, if multiple count and/or status Read Back Commands are issued to the same counter(s) without any intervening reads, all but the first are ignored.

If both count and status of a counter are latched, the first read operation from that counter will return the latched status, regardless of which was latched first. The next one or two reads (depending on whether the counter is programmed for one or two type counts) return the latched count. Subsequent reads return unlatched count.

8.9. Real Time Clock

The Real Time Clock (RTC) module provides a date-and-time keeping device with alarm features and battery backed-up operation. The RTC counts seconds, minutes, hours, days, day of the week, date, month, and year with leap year compensation. Daylight savings compensation is options. Three interrupt features are available: time of day alarm with once a second to once a month range, periodic rates of 122 μs to 500 ms, and end of update cycle notification.

The RTC contains 256 bytes of battery-backed static RAM in two banks, namely, the standard bank and the extended bank. The standard bank contains 10 bytes indicating time and date information, 4 bytes used as four Control Register (A,B,C,D), and 114 bytes used as general purpose RAM. The extended bank has 128 bytes used as general purpose RAM.

Time, calendar, and alarm can be represented in either binary or BCD format. The format is determined by bit 2 of Control Register B. The hour is represented in 12 or 24 hour format and the format is selected by bit 1 of Control Register B. Note that when changing the format, the time registers must be reinitialized to the corresponding data format. See Control Register B in the following section for more information on the configuration of the RTC functions.

The RTC module requires an external oscillating source of 32.768 KHz connected on the TRCX1 and RTCX2 pins. This clock signal is then divided down to 1 Hz signal. The divider chain is controlled by bits [6:4] of Control Register A. Bits [3:0] of the Control Register A select one of the 15 taps from the divider chain to be used as a periodic interrupt. See Control Register A in the following section for divider configuration and rate selections.

8.9.1. RTC REGISTERS AND RAM

The RTC internal registers and RAM are organized as two banks of 128 bytes each, called the standard and extended banks. The first 14 bytes of the standard bank contain the RTC time and date information along with four registers, A–D, that are used for configuration of the RTC. The extended bank contains a full 128 bytes of battery backed SRAM, and will be accessible even when the RTC module is disabled (via the RTC configuration register).

All data movement between the host CPU and RTC is done through registers mapped to the ISA I/O space at locations 70–73h.

I/O locations 70h and 71h are the standard ISA location for the RTC. The map for this bank is shown in Table 27. Locations 72h and 73h are for accessing the extended RAM, and may be disabled.

The first 10 bytes contain information about date and time. It is up to the programmer to make sure that data stored in these locations is within the reasonable values and represents a possible date and time. The exception to these ranges is to store a value of C0h–FFh in the alarm bytes to indicate a “don’t care” situation. Note that bit 7 (UIP) of Control Register A should be read as 0 before each access to these locations. Bit 7 (SET) of Control Register B should be 1 while programming these locations to avoid clashes with an update cycle.

The extended RAM bank is also accessed using an indexed scheme. ISA I/O address 72h is used as the address pointer and ISA I/O address 73h is used as the data register. Index addresses above 127h are not valid.

The internal RTC registers can only be accessed by PCI masters. ISA master access is not supported.

Table 27. RTC (Standard) RAM Bank

Index Address	Name
00h	Seconds
01h	Seconds Alarm
02h	Minutes
03h	Minutes Alarm
04h	Hours
05h	Hours Alarm
06h	Day of Week
07h	Date of Month
08h	Month
09h	Year
0Ah	Register A
0Bh	Register B
0Ch	Register C
0Dh	Register D
0Eh–7Fh	114 Bytes of User RAM

8.9.1.1. Control Register A

Address Offset: 0Ah
Default Value: NA (This register is not affected by any system reset signal.)
Attribute: Read/Write

This register is used for general configuration of the RTC functions.

<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>7</td><td>Update in Progress (UIP). This bit may be monitored as a status flag. 1=Signifies that the update of the timing registers is soon to occur or is in progress.
0=Signifies that the update cycle will not start for at least 244 µs. The time, calendar, and alarm information in RAM is always available when the UIP bit is 0.</td></tr>
<tr><td>6:4</td><td>Division Chain Select (DVx). These three bits control the divider chain for the oscillator.
<table>
<tr><th>DV2</th><th>DV1</th><th>DV0</th><th>Function</th></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Normal Operation</td></tr>
<tr><td>1</td><td>1</td><td>X</td><td>Divider Reset</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>Bypass 15 stages (test mode only)</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>Bypass 10 stages (test mode only)</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Bypass 5 stages (test mode only)</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Invalid</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Invalid</td></tr>
</table></td></tr>
<tr><td>3:0</td><td>Rate Select Bits (RSx). Selects one of 13 taps of the 15 stage divider chain. The selected tap can generate a periodic interrupt if the PIE bit is set in register B. Otherwise, this tap sets the PF flag of register C. If the periodic interrupt is not to be used, these bits should all be set to 0.
<table>
<tr><th>Bits[3:0]</th><th>Periodic Rate</th><th>Bits[3:0]</th><th>Periodic Rate</th></tr>
<tr><td>0000</td><td>none</td><td>1000</td><td>3.90625 ms</td></tr>
<tr><td>0001</td><td>3.90625 ms</td><td>1001</td><td>7.8125 ms</td></tr>
<tr><td>0010</td><td>7.8125 ms</td><td>1010</td><td>15.625 ms</td></tr>
<tr><td>0011</td><td>122.070 µs</td><td>1011</td><td>31.25 ms</td></tr>
<tr><td>0100</td><td>244.141 µs</td><td>1100</td><td>62.5 ms</td></tr>
<tr><td>0101</td><td>488.281 µs</td><td>1101</td><td>125 ms</td></tr>
<tr><td>0110</td><td>976.5625 µs</td><td>1110</td><td>250 ms</td></tr>
<tr><td>0111</td><td>1.953125 ms</td><td>1111</td><td>500 ms</td></tr>
</table></td></tr>
</table>

8.9.1.2. Control Register B

Address Offset: 0Bh
Default Value: X0000XXXb
Attribute: Read/Write

This register is used for general configuration of the RTC functions.

Bits	Description
7	**SET.** Enables the update cycles. 1=A current time update cycle will be aborted, and subsequent update cycles will not occur until SET is returned to 0. When SET=1, BIOS may initialize time and calendar bytes safely. 0=Time update cycle occurs normally once a second. This bit is not affected by RSMRST#.
6	**Periodic Interrupt Enable (PIE).** 1=Enables the generation of the Periodic interrupt. The rate of the Periodic interrupt is determined by bits 3:0 of Control Register A. 0=Disables the generation of the Periodic interrupt. This bit is cleared (set to 0) on active RSMRST#.
5	**Alarm Interrupt Enable (AIE).** 1=Enables the generation of the Alarm interrupt. The Alarm interrupt occurs immediately after a time update of the seconds, minutes, hours which match the alarm setting. An alarm can occur once a second, one an hour, once a day, or one a month. 0=Disables the generation of the Alarm interrupt. This bit is cleared on active RSMRST#.
4	**Update-ended Interrupt Enable (UIE).** 1=Enables the generation of the Update-ended Interrupt which occurs when the update cycle ends. 0=Disables the generation of the Updated-ended interrupt. This bit is cleared on active RSMRST#.
3	**Square Wave Enable (SQWE).** The Square Wave Enable bit serves no function in this device, yet is left in this register bank to provide compatibility with the Motorola 146818B. There is not SQW pin on this device. This bit is cleared on active RSMRST#.
2	**Data Mode (DM).** 1=Binary. 0=BCD. The Data Mode (DM) bit specifies either binary or BCD data representation. This bit is not affected by RSMRST#.
1	**Hour Format (HF).** 1=24-hour mode. 0=12-hour mode. This bit indicates the hour byte format. In 12 hour mode, the seventh bit represents AM as 0 and PM as 1. This bit is not affected by RSMRST#.
0	**Daylight Savings Enable (DSE).** 1=Enable. 0=Disable. When DSE=1, daylight savings is enabled on two special hour updates per year. One is on the first Sunday in April, where time increments from 1:59:59 AM to 3:00:00 AM. The other is the last Sunday in October when the time first reaches 1:59:59 AM, it is changed to 1:00:00 AM. The time must increment normally for at least two update cycles (seconds) previous to these conditions for the time change to occur properly. These special update conditions do not occur when the DSE bit is set to 0. The days for the hour adjustment are those specified in United States federal law as of 1987, which is different than previous years. This bit is not affected by RSMRST#.

8.9.1.3. Control Register C

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

This register is used for various flags. All flag bits are cleared upon active RSMRST# or a read of Register C.

Bits	Description
7	**Interrupt Request Flag (IRQF).** Interrupt Request Flag=PF * PIE + AF * AIE + UF *UFE. This also causes the CH_IRQ_B signal to be asserted.
6	**Periodic Interrupt Flag (PF).** Periodic interrupt Flag is 1 when the tap as specified by the RS bits of register A is 1. If no taps are specified, this flag bit will remain at 0.
5	**Alarm Flag (AF).** 1=All Alarm values match the current time.
4	**Update-ended Flag (UF).** UF=1 immediately following an update cycle for each second.
3:0	**Reserved.** Read as 0.

8.9.1.4. Control Register D

Address Offset: 0Dh
Default Value: NA (This register is not affected by any system reset signal.)
Attribute: Read/Write

This register is used for various flags.

Bits	Description
7	**Valid RAM and Time Bit (VRT).** This bit is set to 1 when the PWRGD signal is asserted. This feature is not typically used. This bit should always be set to 0 during a write to this register.
6	**Reserved.** This bit always returns a 0 and should be set to 0 during register writes.
5:0	**Date Alarm (DA).** These bits store the date of month alarm value. If set to 000000, a ‘don’t care’ state is assumed. The host must configure the date alarm for these bits to operate, even though the bits can be written at any time. If the date alarm is not enabled, these bits will return zeros to mimic the functionality of the Motorola* 146818B. These bits are not affected by RSMRST#.

8.9.2. RTC UPDATE CYCLE

An update cycle occurs once a second, if the SET bit of register B is not set to 1 and the divide chain is properly configured. During this procedure, the stored time and date are incremented, overflow checked, a matching alarm condition is checked, and the time and date are rewritten to the RAM locations. The update cycle will start at least 244 μs after the UIP bit of register A is set to 1, and the entire cycle does not take more than 1984 μs to complete. The time and date RAM locations (0–9) are disconnected from the external bus during this time. To avoid update and data conditions, external RAM access to these locations can safely occur at two times. When an updated-ended interrupt is detected (almost 999 ms is available to read and write valid time and date data). If the UIP bit of register A is detected to be low, there is at least 244 μs before the update cycle begins. Because the overflow conditions for leap years and daylight savings adjustments are based on more than one date or time item, the time before one of these conditions should be set (when adjusting) at least 2 seconds before one of these conditions to ensure proper operation.

8.9.3. RTC INTERRUPTS

The real-time clock interrupt is connected to ISA IRQ8#, and is internally routed within PIIX4. If the RTC module is not enabled, the GPI6 signal is available as the IRQ8# input.

8.9.4. LOCKABLE RAM RANGES

The real-time clock battery-backed RAM supports two 8-byte ranges that can be enabled via the configuration space. If the configuration bits are set, the corresponding range in the RAM will not be readable or writeable. A write cycle to these locations have no effect. A read cycle to these locations do not return the actual location value. Once enabled, this function can only be disabled by a hard reset.

8.9.5. RTC EXTERNAL CONNECTIONS

RTC Crystal

The RTC modules requires an externally connected crystal on the RX1 and RX2 pins.

RTC Battery

The RTC modules requires an external battery connection to maintain the RTC block while PIIX4 is not powered by the system.

Battery characteristics:

- Minimum Voltage: 2.0V
- Recommended Batteries: Duracell 2032, 2025, or 2016

The battery must also be connected to PIIX4 via isolation diodes. This is both a chip-design requirement, as well as a UL requirement. The diode circuit allows the RTC-well to be powered by the battery when system power is not available, but by the system power when it is available. This is done by setting the diode to be reverse biased when system power is not available.

8.10. X-Bus Support

PIIX4 provides positive decode (chip selects) and X-Bus buffer control (XDIR# and XOE#) for a real time clock, keyboard, BIOS, and two programmable IO ranges for PCI and ISA initiated cycles. PIIX4 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). PIIX4 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 PIIX4, an internal IRQ13 is generated and the INTR output from PIIX4 is driven active. When a write to I/O location F0h is detected, PIIX4 negates IRQ13 (internal to PIIX4) 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. PIIX4 informs the CPU of this interrupt via an 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.

8.11. Reset Support

PIIX4 integrates the system reset logic for the system. PIIX4 generates CPURST, PCIRST#, and RSTDRV during power up (PWROK) and when a hard reset is initiated through the RC register, as well as during certain power management resume operations.

The following PIIX4 signals interface directly to the processor:

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

These signals are open drain. Thus, external logic is not required for interfacing with the processors based on 2.5V technology which do not support 3.3V tolerant input buffers.

8.12. Stand-Alone I/O APIC Support

PIIX4 supports a stand-alone I/O APIC device on the ISA X-Bus. PIIX4 provides a chip select signal (APICCS#) for the I/O APIC. It also provides handshake signals to maintain buffer coherency in the I/O APIC 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, PIIX4 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 transceiver are also enabled during accesses to the I/O APIC.

9.0. IDE CONTROLLER FUNCTIONAL DESCRIPTION

PIIX4 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. PIIX4 provides an interface for either primary and secondary IDE connectors or primary drive 0 and primary drive 1 IDE connectors.

All the IDE data transfer command strobes, DMA request and grant signals, IORDY signal, address lines, and data lines interface directly to PIIX4. Two full sets of signals are provided to enhance electrical isolation capabilities.

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.

The following descriptions of PIIX4 IDE signals may sometimes omit the Primary (P) or Secondary (S) prefix. These descriptions apply to both sets of IDE signals. For example, references to IORDY apply to both PIORDY and SIORDY.

9.1. IDE Signal Configuration

PIIX4 has two features that support a number of physical configuration options for system designers.

1. PIIX4 has two completely separate IDE channels. There are no shared signals between the two interfaces. This improves signal timings and allows separate power management monitoring of both channels. Each channels signals can be completely tri-stated or isolated. This allows individual power down of IDE devices on separate channels. It can also be used to allow another device to drive the signal lines. This is useful for Swap-Bay implementations where a system designer may have different types of devices inserted into the bay (such as IDE device or floppy drive).
2. PIIX4 IDE channels can also be configured to support four devices in a Primary and Secondary channel configuration or to support two devices in a Primary Drive 0 and Primary Drive 1 channel configuration. The second configuration allows for the power management of individual devices and is very useful in a mobile environment.

Figure 6 and Figure 7 show these configuration options.

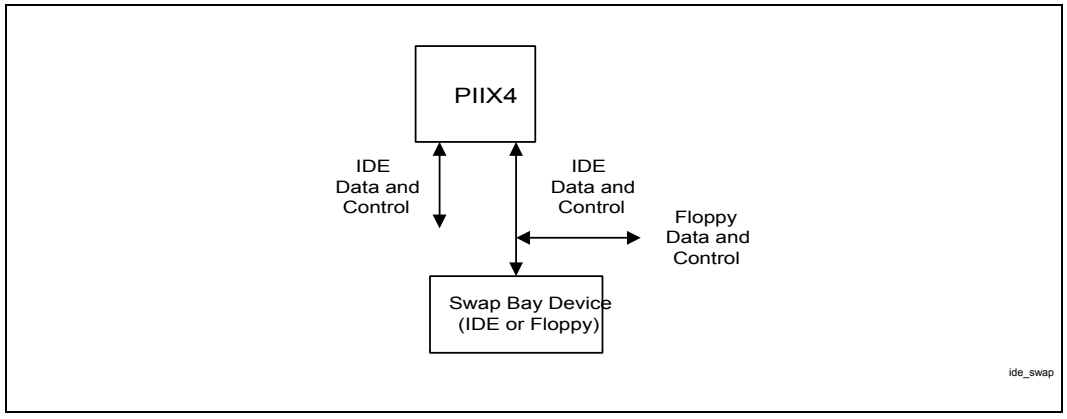


Figure 6. PIIX4 IDE Swap Bay Example

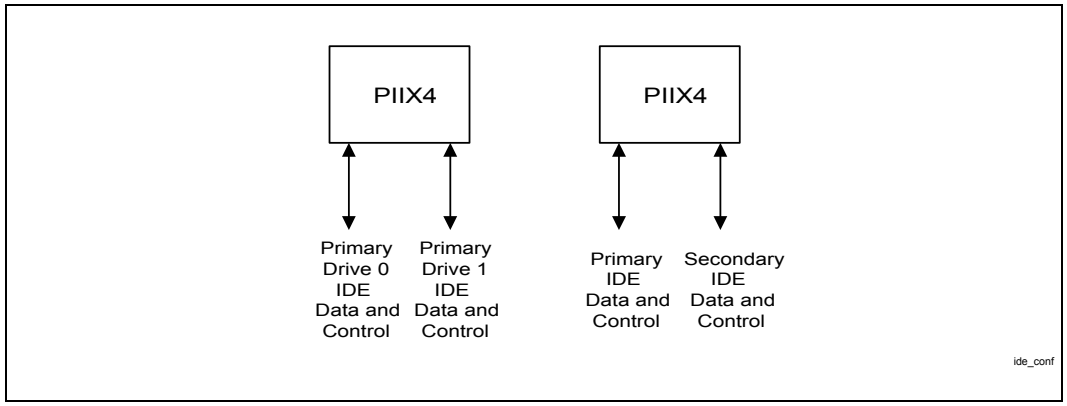


Figure 7. PIIX4 IDE Configurations

9.2. ATA Register Block Decode

The IDE ATA I/O ports are decoded by PIIX4 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 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 28 specifies the registers as they affect the PIIX4 hardware definition.

Table 28. 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 29. 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

PIIX4 claims all accesses to these ranges, if enabled. 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, PIIX4 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).

9.3. PIO IDE Transactions

The PIIX4 IDE controller includes both compatible and 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. The PIIX4 IDE signals are controlled with the granularity of the PCI clock.

Up to two IDE devices may be attached per IDE connector (drive 0 and drive 1). The IDETIM and SIDETIM Registers permit different timing modes to be programmed for drive 0 and drive 1 of the same connector.

The Ultra DMA/33 synchronous DMA timing modes can also be applied to each drive by programming the SDMACTL and SDMATIM registers. When a drive is enabled for synchronous DMA mode operation, the DMA transfers are executed with the synchronous DMA timings. The PIO transfers are executed using compatible timings or fast timings if also enabled.

PIO IDE Timing Modes

IDE data port transaction latency consists of startup latency, cycle latency, and shutdown latency. 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 prior to assertion of the read and write strobes (DIOR# and DIOW#).

Cycle latency consists of the I/O command 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. The command strobe assertion width for the enhanced timing mode 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.

If IORDY is asserted when the initial sample point is reached, no wait states are added to the command strobe assertion length. 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.

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#). Shutdown latency is 2 PCI clocks in duration.

The IDE timings for various transaction types are shown in Table 30. Note that bit 2 (16-bit I/O recovery enable) of the ISA I/O Recovery Timer Register does not add wait states to IDE data port read accesses when any of the fast timing modes are enabled.

Table 30. IDE Transaction Timings (PCI Clocks)

IDE Transaction Type	Startup Latency	ISP	RCT	Shutdown Latency
Non-Data Port Compatible	4	11	22	2
Data Port Compatible	3	6	14	2
Fast Timing Mode	2	2–5	1–4	2

IORDY Masking

The IORDY signal can be ignored and assumed asserted at the first IORDY Sample Point (ISP) on a drive by drive basis via the IDETIM Register.

PIO 32-Bit IDE Data Port Accesses

A 32-bit PCI transaction run to the IDE data address (01F0h primary, 0170h secondary) results in two back-to-back 16-bit transactions to the IDE data port. The 32-bit data port feature is enabled for all timings, not just enhanced timing. For compatible timings, a shutdown and startup latency is incurred between the two 16-bit halves of the IDE transaction. This guarantees that the chip selects will be negated for at least 2 PCI clocks between the two cycles.

PIO IDE Data Port Prefetching and Posting

PIIX4 can be programmed via the IDETIM registers to allow data to be posted to and prefetched from the IDE data ports. Data prefetching is initiated when a data port read occurs. The read prefetch eliminates latency to the IDE data ports and allows them to be performed back to back for the highest possible PIO data transfer rates. The first data port read of a sector is called the demand read. Subsequent data port reads from the

sector are called prefetch reads. The demand read and all prefetch reads must be of the same size (16 or 32 bits).

Data posting is performed for writes to the IDE data ports. The transaction is completed on the PCI bus after the data is received by PIIX4. PIIX4 then runs the IDE cycle to transfer the data to the drive. If the PIIX4 write buffer is non-empty and an unrelated (non-data or opposite channel) IDE transaction occurs, that transaction will be stalled until all current data in the write buffer is transferred to the drive.

9.4. Bus Master Function

PIIX4 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, PIIX4 off-loads the CPU and improves system performance in multitasking environments. Both devices attached to a connector can be programmed for bus master transfers, but only one channel per connector can be active at a time.

Physical Region Descriptor Format

The physical memory region to be transferred is described by a Physical Region Descriptor (PRD). The PRDs are stored sequentially in a Descriptor Table in memory (Figure 8). The data transfer proceeds until all regions described by the PRDs in the table have been transferred. Note that the PIIX4 bus master IDE function does not support memory regions or Descriptor tables located on ISA.

Descriptor Tables must be aligned on 64-Kbyte boundaries. Each PRD entry in the table is 8 bytes in length. The first 4 bytes specify the byte address of a physical memory region. This memory region must be DWord aligned and must not cross a 64-Kbyte boundary. The next 2 bytes specify the size or transfer count of the region in bytes (64-Kbyte limit per region). A value of 0 in these 2 bytes indicates 64 Kbytes (thus the minimum transfer count is 1). If bit 7 (EOT) of the last byte is a 1, it indicates that this is the final PRD in the Descriptor table. Bus master operation terminates when the last descriptor has been retired.

When the Bus Master IDE controller is reading data from the memory regions, bit 1 of the Base Address is masked and byte enables are asserted for all read transfers. The controller reads to a boundary of 64 bytes, regardless of byte count field of the PRD. However, only the byte count value is transferred to the drive. When writing data, bit 1 of the Base Address is not masked and if set, causes the lower Word byte enables to be negated for the first DWord transfer. The write to PCI typically consists of a 32-byte cache line. If valid data ends prior to end of the cache line, the byte enables are negated for invalid data.

The total sum of the byte counts in every PRD of the descriptor table 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.

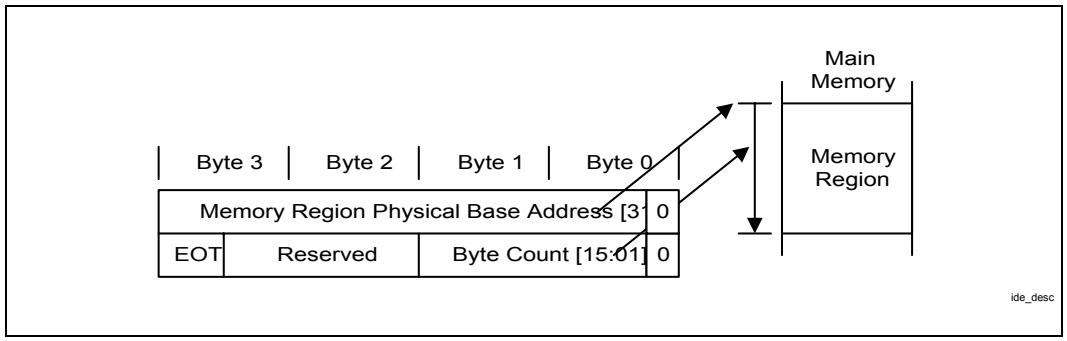


Figure 8. Physical Region Descriptor Table Entry

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. This includes the total amount of data to be transferred.
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 PIIX4. The channel remains masked until the first descriptor is loaded.
5. The controller transfers data to or from memory responding to the DMA requests from the IDE device. When the last data transfer for a memory region has been completed on the IDE interface, the next PRD is fetched from the table. The controller then begins transferring data to or from that PRD's memory region.
6. The IDE device signals an interrupt once its programmed data count has been transferred. The IDE device will also negate its DMA request signal, causing PIIX4 to stop transferring data. If PIIX4 has also transferred the final data from the last PRD memory region, it will reset the BMIDEA bit in the status register and mask the DMA request signal from the drive.
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. See the Bus Master IDE Status Register section for a description of controller transfer status.

Line Buffer

A single line buffer exists for the PIIX4 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. Memory writes are typically 4-DWord bursts and invalid DWords have C/BE[3:0]#=0Fh. The line buffer allows burst data transfers to proceed at peak transfer rates.

The Bus Master IDE Active bit in Bus Master IDE Status register is reset automatically when the controller has transferred all data associated with a Descriptor Table (as determined by EOT bit in last PRD). The IDE Interrupt Status bit is set when the IDE device generates an interrupt. These events may occur prior to line buffer emptying for memory writes. If either of these conditions exist, all PCI Master non-memory read accesses to PIIX4 are retried until all data in the line buffers has been transferred to memory.

Bus Master IDE Timings

The timing modes used for Bus Master IDE transfers are identical to those for PIO transfers. The DMA Timing Enable Only bits in IDE Timing register can be used to program fast timing mode for DMA transactions only. This is useful for IDE devices whose DMA transfer timings are faster that its PIO transfer timings.

The IDE device DMA request signal is sampled on same PCI clock that the IO strobe is negated. If inactive, the DMA Acknowledge signal is negated on the next PCI clock and no more transfers take place until DMA request is again asserted.

9.5. "Ultra DMA/33" Synchronous DMA Operation

Ultra DMA/33 is a new physical protocol used to transfer data between an Ultra DMA/33 capable IDE controller such as PIIX4 and one or more Ultra DMA/33 capable IDE devices. It utilizes the standard Bus Master IDE functionality and interface to initiate and control the transfer. Ultra DMA/33 utilizes a "source synchronous" signaling protocol to transfer data at rates up to 33 Mbytes/sec. The Ultra DMA/33 definition also incorporates a Cyclic Redundancy Checking (CRC-16) error checking protocol. CRC-16 only has the ability for detecting errors, not correcting them.

Signal Descriptions

The Ultra DMA/33 protocol requires no extra signal pins on the IDE connector. It does redefine a number of the standard IDE control signals when in Ultra DMA/33 mode. These redefinitions are shown in Table 31. Read cycles are defined as transferring data from the IDE device to PIIX4. Write cycles are defined as transferring data from PIIX4 to IDE device.

Table 31. Ultra DMA/33 Control Signal Redefinitions

Standard IDE Signal Definition	Ultra DMA/33 Read Cycle Definition	Ultra DMA/33 Write Cycle Definition	PIIX4 Primary Channel Signal	PIIX4 Secondary Channel Signal
DIOW#	STOP	STOP	PDIOW#	SDIOW#
DIOR#	DMARDY#	STROBE	PDIOR#	SDIOR#
IORDY	STROBE	DMARDY#	PIORDY	SIORDY

The DIOW# signal is redefined as STOP for both read and write transfers. This is always driven by PIIX4 and is used to request that a transfer be stopped or as an acknowledgment to stop a request from IDE device.

The DIOR# signal is redefined as DMARDY# for transferring data from the IDE device to PIIX4 (read). It is used by PIIX4 to signal when it is ready to transfer data and to add wait states to the current transaction. The DIOR# signal is redefined as STROBE for transferring data from PIIX4 to the IDE device (write). It is the data strobe signal driven by PIIX4 on which data is transferred during each rising and falling edge transition.

The IORDY signal is redefined as STROBE for transferring data from the IDE device to PIIX4 (read). It is the data strobe signal driven by the IDE device on which data is transferred during each rising and falling edge transition. The IORDY signal is redefined as DMARDY# for transferring data from PIIX4 to the IDE device (write). It is used by the IDE device to signal when it is ready to transfer data and to add wait states to the current transaction.

All other signals on the IDE connector retain their functional definitions during Ultra DMA/33 operation.

Operation

Initial setup programming consists of enabling and performing the proper configuration of PIIX4 and the IDE device for Ultra DMA/33 operation. For PIIX4, this consists of enabling Synchronous DMA mode and setting up appropriate Synchronous DMA timings.

When ready to transfer data to or from an IDE device, the Bus Master IDE programming model is followed. Once programmed, the drive and PIIX4 control the transfer of data via the Ultra DMA/33 protocol. The actual data transfer consists of three phases, a startup phase, a data transfer phase, and a burst termination phase.

The IDE device begins the startup phase by asserting DMARQ signal. When ready to begin the transfer, PIIX4 asserts DMACK# signal. When DMACK# signal is asserted, the host controller drives CS0# and CS1# inactive, DA0–DA2 low, and the IDE device drives IOCS16# inactive. For write cycles, PIIX4 negates STOP, waits for the IDE device to assert DMARDY#, and then drives the first data word and STROBE signal. For read cycles, PIIX4 tri-states the DD lines, negates STOP, and asserts DMARDY#. The IDE device then sends the first data word and STROBE.

The data transfer phase continues the burst transfers with the data transmitter (PIIX4 writes, IDE device reads) providing data and toggling STROBE. Data is transferred (latched by receiver) on each rising and falling edge of STROBE. The transmitter can pause the burst by holding STROBE high or low, resuming the burst by again toggling STROBE. The receiver can pause the burst by negating DMARDY# and resumes the transfers by asserting DMARDY#. PIIX4 pauses a burst transaction to prevent an internal line buffer over or under flow condition, resuming once the condition has cleared. It may also pause a transaction if the current PRD byte count has expired, resuming once it has fetched the next PRD.

The current burst can be terminated by either the transmitter or receiver. A burst termination consists of a Stop Request, Stop Acknowledge and transfer of CRC data. A Burst must first be paused as described above before it can be terminated. PIIX4 can then stop the burst by asserting STOP, with the IDE device acknowledging by negating DMARQ. The IDE device can then stop the burst by negating DMARQ and PIIX4 acknowledges by asserting STOP. The transmitter then drives the STROBE signal to a high level. PIIX4 then drives the CRC value onto the DD lines and negate DMACK#. The IDE device latches the CRC value on rising edge of DMACK#. PIIX4 terminates a burst transfer if it needs to service the opposite IDE channel, if a Programmed I/O (PIO) cycle is executed to the IDE channel currently running the burst, or upon transferring the last data from the final PRD.

CRC Calculation

Cyclic Redundancy Checking (CRC-16) is used for error checking on Ultra DMA/33 transfers. The CRC value is calculated for all data by both PIIX4 and the IDE device over the duration of the Ultra DMA/33 burst transfer segment. This segment is defined as all data transferred with a valid STROBE edge from DDACK# assertion to DDACK# negation. At the end of the transfer burst segment, PIIX4 drives the CRC value onto the DD[15:0] signals. It is then latched by the IDE device on negation of DDACK#. The IDE device compares the PIIX4 CRC value to its own and reports an error if their is a mismatch.

Synchronous DMA Timings

The timings for Ultra DMA/33 are programmed into the Ultra DMA/33 Timing Register. The programmable timings include Cycle Time (CT) and Ready to Pause (RP) time. The Cycle Time represents the minimum pulse width of active data strobe (STROBE) signal. The Ready to Pause time represents the number of PCI clocks PIIX4 waits from negation of DMARDY# to the assertion of STOP when it desires to stop a burst read transaction.

10.0. USB HOST CONTROLLER FUNCTIONAL DESCRIPTION

PIIX4 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 PIIX4 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 PIIX4 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 9 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 PIIX4'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 PIIX4'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 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.

For additional information on the functionality of PIIX4 USB Host Controller, refer to the Universal Host Controller Interface (UHCI) Design Guide, Revision 1.1 available from Intel Literature Center with order number 297650-002. Note that the UHCI Design Guide refers to USB ports 1 and 2. The PIIX4 USB ports are ports 0 and 1 respectively.

Additions to the PIIX4 USB interface beyond UHCI, revision 1.1 include support for over-current detection on USB ports 0 and 1. If an over-current condition is detected on a USB port, that port will be disabled. Bits 10:11 in each Port Status and Control register are used to report over-current conditions.

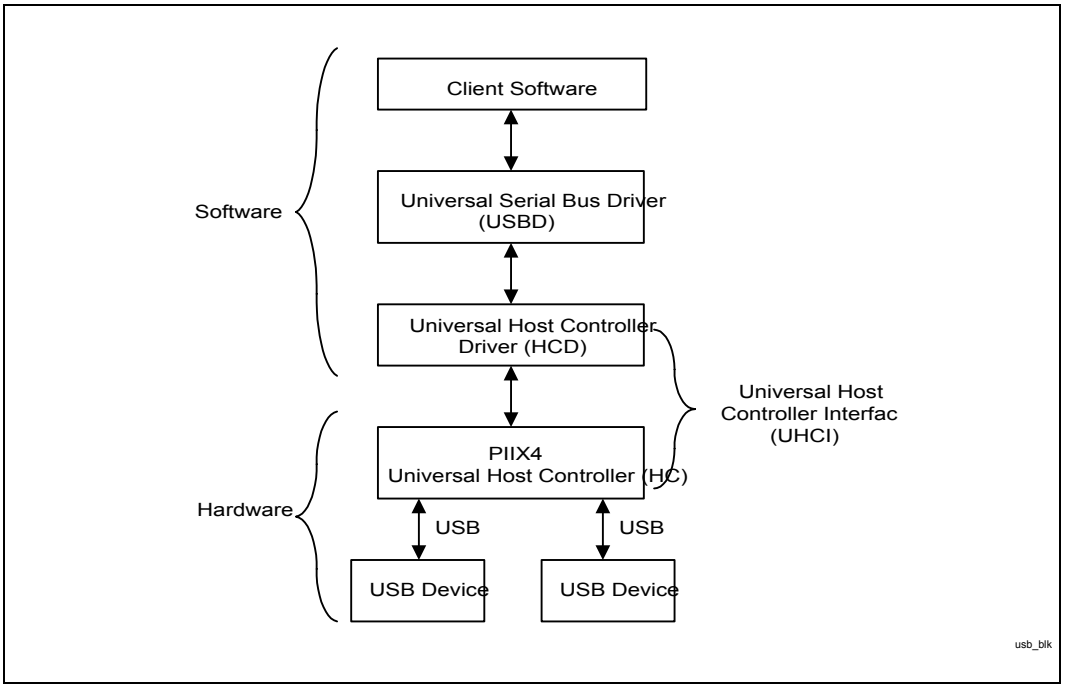


Figure 9. USB System Conceptual View

11.0. POWER MANAGEMENT FUNCTIONAL DESCRIPTION

11.1. Power Management Overview

The PIIX4 power management function provides a wide range of capabilities and configuration options allowing a system designer to implement a wide range of power saving modes.

PIIX4 provides for four main areas of power management:

- Clock Control and Processor Complex Management
- Peripheral Device Management
- System Management (SMI Generation, System Management Bus)
- System Suspend and Resume

PIIX4 uses several mechanisms to help the power management software initiate and manage the transitions between the power managed states. These include system-wide and peripheral event monitors to identify idle and wake-up conditions, Intel's System Management Interrupt (SMI#) support, Advanced Power Management (APM) 1.2 interface, Pentium and Pentium II processor STPCLK# and SLP# Clock Control, Suspend/Resume Hardware, and System Management Bus. PIIX4 supports the Advanced Configuration and Power Interface (ACPI) specification.

System power management functions under a combination of both hardware and software control. The software consists of System Management Mode (SMM) BIOS for legacy control and Operating System (OS) for ACPI. The basic operation consists of software setting up the desired configuration and power management mode and corresponding power savings level. The hardware then performs the necessary actions to maintain the power mode. PIIX4 also monitors the system for events which may require changing the system power mode. When one of these events is detected, PIIX4 informs the software, which makes the decision to change power modes. This is done by a System Management Interrupt (SMI#) for legacy SMM BIOS or a System Control Interrupt (SCI) for ACPI OS.

Brief descriptions of primary power management functions follow. More complete descriptions are provided later in this section.

Clock Control: When the operating system, application program, or system software is not performing useful work, the processor complex (Processor, Host Bridge, DRAM, L2 Cache) does not need to be executing cycles and, therefore, can be placed in a Standby mode.

- Flexible STPCLK# Mechanism for Host Clock Control
  - Throttling: STPCLK# Duty Cycle Control for Low Frequency Emulation
  - Stop Grant State: With Processor Clock input RUNNING
  - Stop Clock State: With Processor Clock input STOPPED
- Clock Wake-up (Clock Break Events) from Interrupts, Device Monitors, Bus Activity.
- Burst Mechanism for Hardware-Controlled Return to Standby
  - Fast and Slow Burst timers
- PCI Clock Control (CLKRUN#) is independent from Host Clock Control.
- SRAM ZZ mode for L2 power management during Standby.
- Thermal Management Input for Clock Throttling during critical thermal condition.

Peripheral Device Management: Peripheral device resources are monitored to detect when a specific device is idle. PIIX4 then informs system power management software, which can place that individual device into a

power managed condition (such as Local Standby or Powered Off). Accesses targeting that device are then monitored. When detected, an SMI# is generated to allow the software to restore the device to operation.

- 14 distinct device monitors and idle timers for various system devices
  - Four generic device monitors
  - Can monitor devices on PCI or ISA bus
  - Can monitor General Purpose Inputs
- I/O SMI# traps with I/O cycle restart timing
- Device address ranges can be used to forward cycles to ISA bus

System Management: In addition to individual devices, PIIX4 provides capabilities to monitor other events within the system, including an external power button, notebook lid or other type of switch, global system activity, thermal alarm input, countdown timers, and SMBus message generation and receipt. These events can generate an SMI to the processor allowing the software to manage system as necessary.

System Suspend: Once the power management software has determined that the system is fully idle or that a critical system event has occurred, it can place the system into a suspend state, which allows for increased power savings. The software configures PIIX4 for the type of suspend, types of resume or wake-up events, and then PIIX4 automatically transitions the system into suspend. When an enabled resume event is detected, PIIX4 automatically restores the system to operation.

- Three suspend states
  - Power-on-Suspend (POS) with three system reset options
  - Suspend-to-RAM (STR)
  - Suspend-to-Disk (STD) or Soff Off (SOff)
- Global Standby Timer (also active during suspend) to monitor for overall system idleness and as a resume timer
- Power Button Input (PWRBTN#)
  - Override feature forcing immediate transition to Soft Off
- Battery Low indication pin (BATLOW#)
- Shadow registers for standard AT write only registers to save and restore system state information
- “Resume Well” to monitor wake-up events during suspend
- Resume power and reset sequencing

11.2. Clock Control

PIIX4 provides the ability to separately control the system Host clocks and PCI clocks. The Host Clock Control primarily uses the processor clock control features, but adds unique capabilities to allow for more flexible and robust power management. It supports both the Pentium processor Stop Grant and Stop Clock states, as well as the Pentium II processor Stop Grant and Sleep states or Quick Start and Deep Sleep states. The PCI Clock Control follows the Clock Run mechanism as described in the PCI Mobile Design Guide. An example system configuration is shown in Figure 10.

11.2.1. HOST CLOCK CONTROL MECHANISMS

PIIX4 support four primary Host Clock Control Mechanisms, with three types of variations. System events can be monitored to break out of clock control modes or to generate burst execution. Software enables clock control by setting [CC_EN] bit along with other optional control bits. Table 32 details register bit settings required to place the PIIX4 clock control into the various modes of operation. Figure 11 and Figure 12 give examples of various clock control conditions.

- Primary Mechanisms
  - Stop Grant or Quick Start
  - Sleep
  - Stop Clock
  - Deep Sleep
- Variations
  - Manual Throttle
  - Thermal Throttle
  - Stop Break and Burst Execution

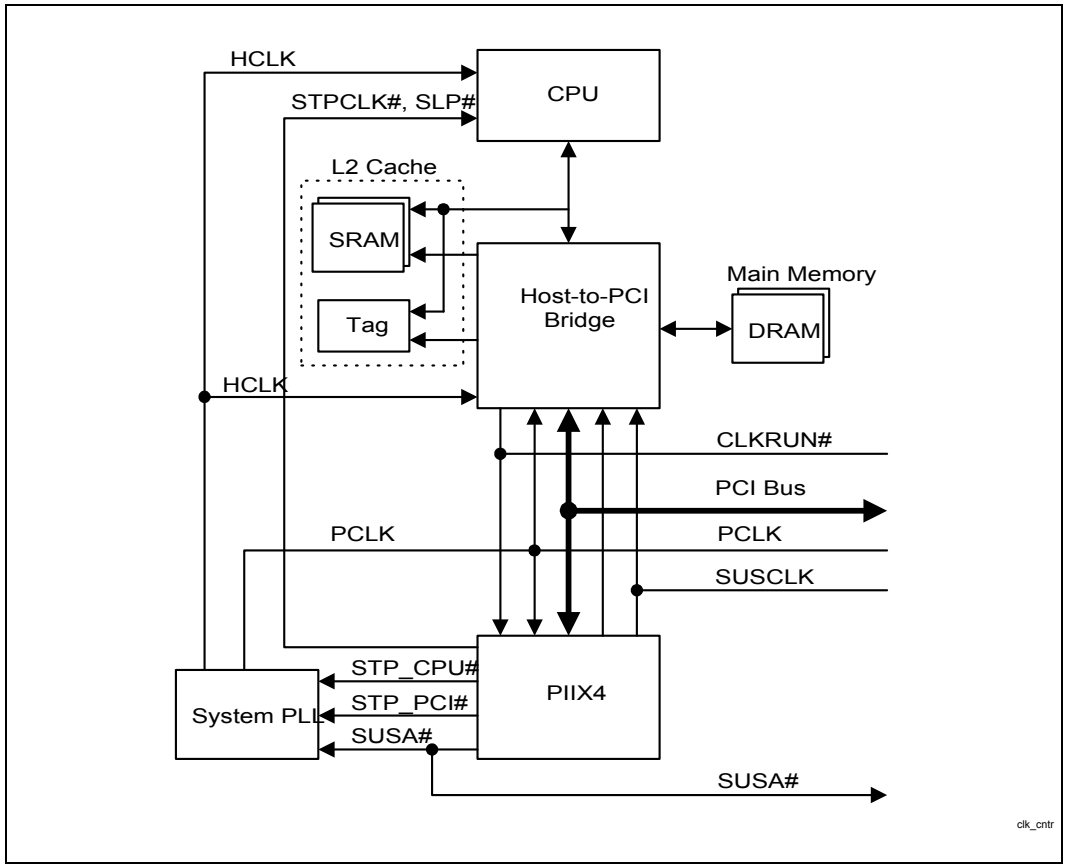


Figure 10. Clock Control

Table 32. Clock Control Programming Modes

Clock Control Mode	Register Read	CC_EN	STP_CLK_EN	SLEEP_EN	THT_EN
Thermal Throttle		X	X	X	X
Disable Clock Control		0	X	X	X
Stop Grant/Quick Start without Throttle	LVL 2	1	X	X	0
Stop Grant/Quick Start with Throttle	LVL 2	1	X	X	1
Reserved	LVL 3	1	0	0	X
Sleep	LVL 3	1	0	1	X
Stop Clock	LVL 3	1	1	0	X
Deep Sleep	LVL 3	1	1	1	X
Enable Burst Execution[1]	LVL 3	1	X	X	X

NOTES:

1. Burst execution is enabled for any of the above modes (except disabled and thermal throttle) if BRST_EN bit is set.

Stop GRANT or Quick Start State: Initiated by a read to the LVL2 register, the STPCLK# signal is asserted and PIIX4 waits for the processor to issue a Stop Grant bus cycle. When Stop Grant cycle is terminated, PIIX4 asserts the ZZ pin to the L2 SRAM, if the [ZZ_EN] bit is set. PIIX4 does not assert the CPU_STP# signal and the Host clocks remain running in this state. In this state, the processor disables clocks to portions of its internal logic, but is able to snoop host bus cycles in order to maintain cache coherency. To exit this state, PIIX4 negates the ZZ signal (if applicable) and then negates STPCLK#.

Sleep State: Initiated by a read to the LVL3 register, the STPCLK# signal is asserted and the processor issues a Stop Grant bus cycle. When Stop Grant cycle is terminated, PIIX4 asserts the ZZ pin to the L2 SRAM if the [ZZ_EN] bit is set and after 50 PCI clocks asserts the SLP# signal. PIIX4 does not assert the CPU_STP# signal and the Host clocks remain running in this state. In this state, the processor disables clocks to portions of its internal logic. The processor does not snoop host bus cycles and system designers must ensure that no host cycles to main memory are executed by other system masters. Disabling of the PCI arbiter is a method used by the Host Controller, for example. To exit this state, PIIX4 negates the SLP# signal, waits approximately 32 μs and then negates the ZZ signal (if applicable); two PCI clocks later STPCLK# is negated.

Stop CLOCK State (Pentium II processor only): Initiated by a read to the LVL3 register, the STPCLK# signal is asserted and the processor issues a Stop Grant Bus Cycle. When Stop Grant cycle is terminated, PIIX4 asserts the ZZ pin to the L2 SRAM if the [ZZ_EN] bit is set, asserts the SUS_STAT1# signal to Host Bridge to enable Suspend Refresh for the DRAM, and then asserts the CPU_STP# signal to the clock synthesizer. The Host clocks stop running in this state. The processor does not snoop host bus cycles and system designers must ensure that no host cycles to main memory are executed by other system masters. To exit this state, PIIX4 negates the CPU_STP# signal. At this time PIIX4 loads the Fast Burst Timer (see below) with the [CPU_LCK] value and count down allowing time for the processor PLL to lock. After the timer expires, PIIX4 negates the SUS_STAT1# signal, the ZZ signal (if applicable), and finally STPCLK#.

Deep Sleep State (Pentium II processor only): Initiated by a read to the LVL3 register, the STPCLK# signal is asserted and the processor issues a Stop Grant Bus Cycle. When Stop Grant cycle is terminated, PIIX4 asserts the ZZ pin to the L2 SRAM if the [ZZ_EN] bit is set, asserts the SLP# signal, asserts the SUS_STAT1# signal to Host Bridge to enable Suspend Refresh for the DRAM and then asserts the CPU_STP# signal to the clock synthesizer. The Host clocks stop running in this state. The processor does not snoop host bus cycles

and system designers must ensure that no host cycles to main memory are executed by other system masters. To exit this state, PIIX4 negates the CPU_STP# signal. Again, PIIX4 loads the Fast Burst Timer with the [CPU_LCK] value and count down allowing time for the processor PLL to lock. After the timer expires, PIIX4 negates the SUS_STAT1# signal, the SLP# signal, and ZZ signal
(if applicable), and STP_CLK#.

System Throttle Control: If the system has been placed into the Stop Grant or Quick Start states and [THT_EN] bit is set, PIIX4 toggles the STPCLK# signal and ZZ signal (if [ZZ_EN] set) with a period of 244 μs (approximately eight 32-kHz clock periods) and a programmable duty cycle. This system toggles between full-speed operation and the Stop Grant or Quick Start state. The duty cycle can be set in 12.5% increments by programming the [THTL_DTY] bits in the Processor Control (P_CNTRL) register. This emulates a reduced frequency Host clock, resulting in associated power savings.

Thermal Throttle Control: If the THRM# signal is asserted for greater than 2 seconds and the system is not in a Stop Clock, Sleep, or Deep Sleep state, PIIX4 automatically starts toggling the STPCLK# signal and ZZ signal (if [ZZ_EN] set) with a period of 244 μs and a programmable Duty Cycle. This system toggles between full-speed operation and the Stop Grant State. The Duty Cycle can be set in 12.5% increments by programming the [THRM_DTY] bits in the Count B (CNTB) register. The functionality of thermal throttling is independent of [THRM_EN] bit, which is used to enable events for other power management functions. The [THRM_DTY] field must be programmed by the BIOS. This emulates a reduced frequency Host clock, resulting in reduced power and thermal output. When the THRM# signal is negated, the system returns to clock control previously in use.

Stop Break and Burst Execution: Once the hardware has been placed into a clock control state, it can be restored to full operation by system hardware or software. Software can restore the system to full operation by clearing the [CC_EN] bit. Hardware events can be enabled to return the system to a non-clock controlled condition. If the [BRST_EN] bit is reset, these events are called Stop Break Events. Alternatively, if the [BRST_EN] bit is set, these events are called Burst Events.

Stop Break events completely return the system to non-clock controlled state. To restore clock control, software must set the desired clock control configuration and again perform a read from LVL2 or LVL3 registers to initiate the control.

Burst events cause the reload of a Burst timer, which begins to count down from its loaded value. While the timer is counting, the system returns to full clock operation. Once the burst timer expires, the system automatically returns to the clock controlled state. PIIX4 provides two different burst timers, a fast burst timer (which generates a short count) and a slow burst timer (which generates a longer count).

Care must be taken before placing the system into a LVL2 or LVL3 state when the [BRST_EN] bit is set. Prior to LVL2 or LVL3 register read, software must ensure that no external burst events are active and only
Device 3 idle timer should be enabled as a burst event. The Device 3 idle timer is then enabled with all reload events disabled. The LVL2 or LVL3 register read is performed placing the system into a LVL2 or LVL3 clock control condition. The Device 3 idle timer will then generate a burst event upon expiration. During this first burst, the desired burst events are then enabled. Failure to follow this procedure may cause the PIIX4 to miss LVL2 or LVL3 register reads.

The thermal throttle state is not affected by [CC_EN] bit settings, nor are any hardware Stop Break or Burst events.

Stop Break events are the logical “OR” of Fast Burst and Slow Burst Events. If the [BRST_EN] bit is high, the Burst events reload their associated burst timer. When the [BRST_EN] bit is low, these events generate
a Stop Break event. The Fast Burst and Slow Burst timer and burst event programming information is
shown on next page.

Fast Burst Timer Programming Information:

Resolution: 1 ms

Count: 5 bit [FB_CNT]

Slow Burst Timer Programming Information:

Resolution: 1 sec

Count: 4 bit [SB_CNT]

Fast Burst Timer Programming Information (for PLL Lock):

Resolution: 1µ or 1 ms [CPU_SEL]

Count: 5 bit [CPU_LCK]

Fast Burst Events

IRQ0: [BRLD_EN_IRQ0]

IRQ8: [BRLD_EN_IRQ8]

NMI, INIT, IRQ[1,3:7,9:15]: [BRLD_EN_IRQ]

PCI Bus Master Activity: [BRLD_EN_BM]

Device 0–13 Monitors: [BRLD_EN_DEVx] x=1–13

Slow/Fast Burst Select: [BRLD_SEL_DEVx] x=1–3, 5

PCI Activity (FRAME# Assertion): [BRLD_EN_PCI]

GPI1 Asserted: [BRLD_EN_PME]

LID Asserted: [BRLD_EN_PME]

— Polarity Select: [LID_POL]

PWRBTN# Asserted: [BRLD_EN_PME]

SMI# Event: [BRLD_EN_PME]

Slow Burst Events

Device 0–13 Monitors: [BRLD_EN_DEVx] x=1–3, 5

Slow/Fast Burst Select: [BRLD_SEL_DEVx] x=1–3, 5

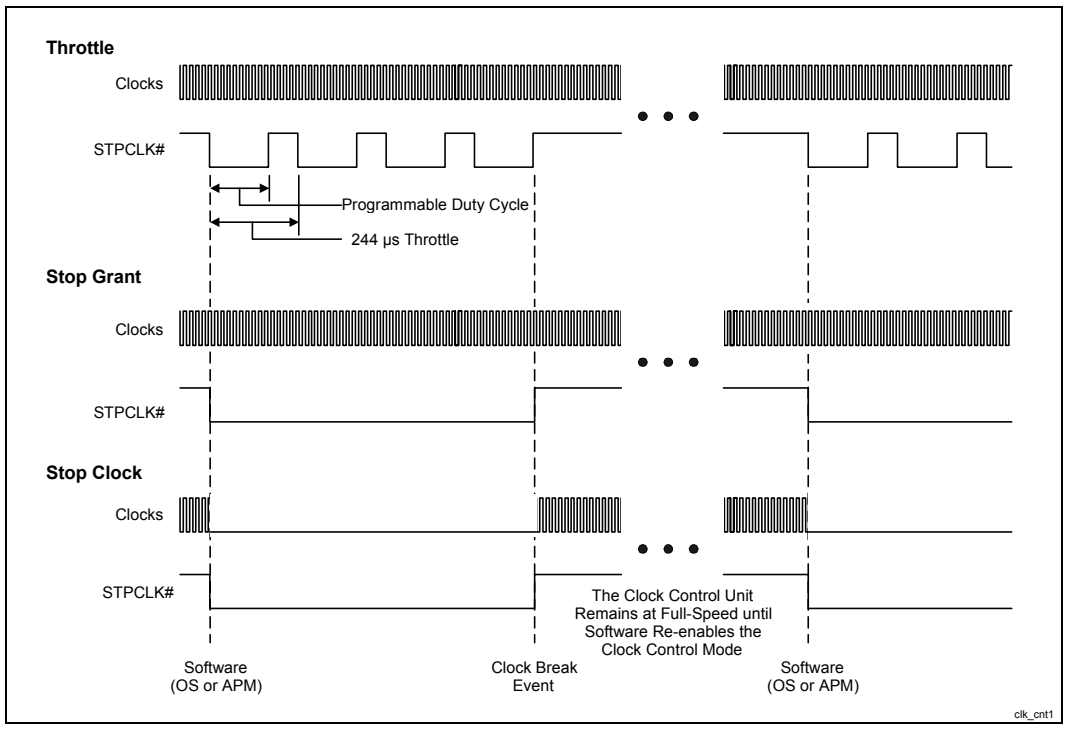


Figure 11. Clock Control Mechanisms (NON-Burst)

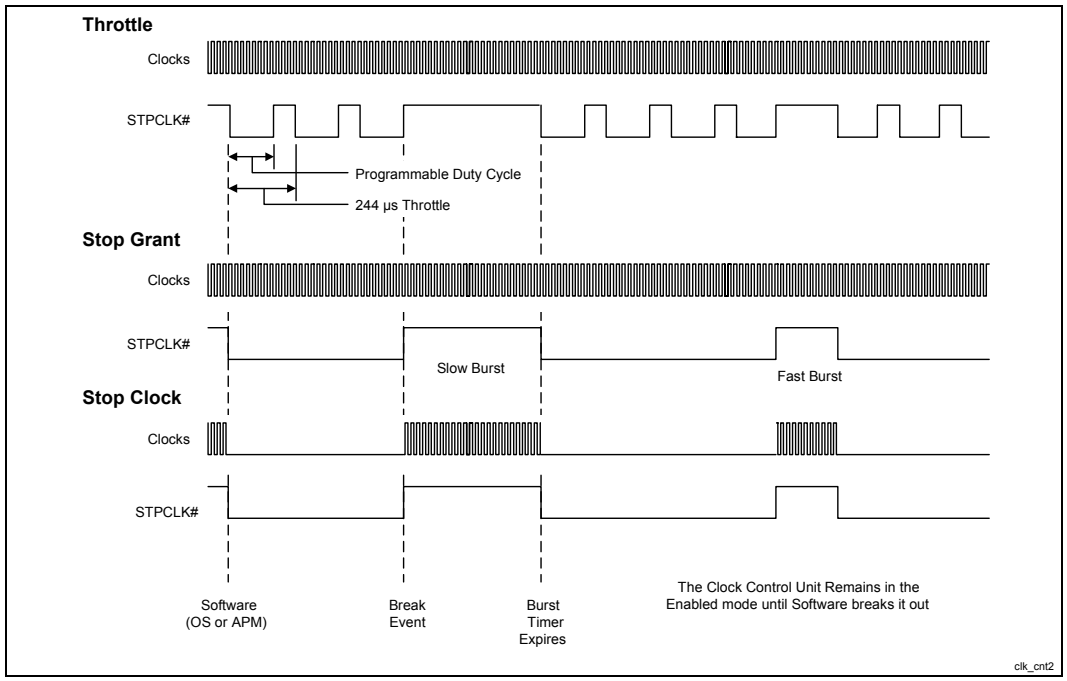


Figure 12. Clock Control Mechanisms (with BURST Enabled)

11.2.2. STOP CLOCK AND DEEP SLEEP STATE EXAMPLE SEQUENCE

The Stop Clock Mode requires special consideration to allow the processor PLL to stabilize before starting any activity that would require the processor to snoop its internal cache. The following is an example of system transition into and out of Stop Clock. Figure 13 shows an example timing diagram. The numbers shown in braces {} below correspond to the numbers shown in Figure 13.

Entering the processor **Stop Clock State:**

- Software sets PIIX4 for the appropriate Clock Control Mechanism.
- Software disables the PCI arbiter in the Host Bridge.
- Software sets PIIX4 to enable the Stop Clock Mode by reading LVL3 Register.
- PIIX4 asserts STPCLK# pin.
- Processor accepts STPCLK# interrupt, flushes buffers, sends the STOP GRANT bus cycle.
- The Host Bridge forwards Stop Grant bus cycle to PCI bus and does PCI Master Abort.
- The Host bridge completes the Stop Grant bus cycle by returning an RDY# (BRDY#) to the processor.
- Processor gates the internal clocks to the processor core and enters the Stop Grant state. PIIX4 asserts the ZZ pin to the SRAM if the [ZZ_EN] bit is set.
- PIIX4 waits up to 2–32-kHz clock periods after receiving the Stop Grant Bus Cycle to allow the Host Bridge to complete pending cycles to DRAM. PIIX4 then asserts the SUSTAT1# signal to the Host Bridge {1}.

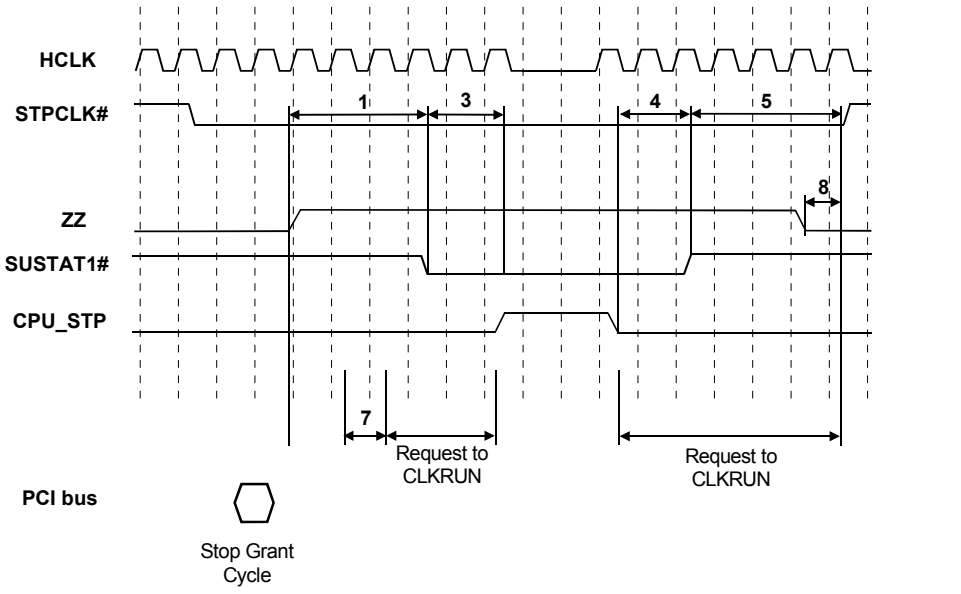
- PIIX4 waits an additional 32-khz clock period while the Host Bridge switches from Normal Refresh to Suspend Refresh, then asserts the CPU_STP# signal to the clock synthesizer which stops the Host clock to the processor, host bridge, L2 SRAM, and SDRAM. The processor is now in the Stop Clock State {3}.
- The processor will stay in this state until a Stop Break or Burst Event occurs.

Leaving the processor **Stop Clock State:**

- A Stop Break or Burst Event occurs.
- PIIX4 negates the CPU_STP# signal to the clock synthesizer to start the Host clocks.
- PIIX4 waits for the processor PLL to start and lock (about 1 ms + 1 32-khz period) then negates the SUS_STAT1# signal {4}. The Host Bridge will switch from Suspend Refresh to Normal Refresh after the SUS_STAT1# signal is negated.
- PIIX4 waits up to 2**–**32-kHz periods and then negates the STPCLK# signal {5}. If the ZZ signal was enabled, PIIX4 will negate the ZZ signal a minimum of 2 PCI clocks before the STPCLK# signal is negated {8}.
- Processor returns to the On state and resumes normal execution.
- Software re-enables the Host Bridge PCI Arbiter.

NOTE

PCI masters must not be granted the bus before the processor is ready to snoop the PCI cycles (STPCLK# negated). Therefore, when the system enters the Stop Clock State the PCI Arbiter (Host Bridge) must be disabled. When the PCI arbiter is disabled any request to the arbiter (REQ# or PHOLD# from PIIX4) should generate an SMI# so that power management software can re-enable the arbiter. These requests can be trapped using the Device 8 (LPT) Peripheral Device Monitor.



NOTES:
The numbers shown in this figure are referenced in the bullets on previous page for entering and leaving the Stop Clock State.

Figure 13. Stop Clock Example

11.2.3. PCI CLOCK CONTROL

PIIX4 manages the PCI Clock Control through the CLKRUN# protocol as specified by the PCI Mobile Design Guide Rev 1.0. PIIX4 acts as the CLKRUN# Central Resource.

If the [CLKRUN_EN] bit is set in the Clock Control Register, PIIX4 requests to stop the PCI Clock if the bus has been idle for 26 PCI Clocks. PIIX4 asserts the PCI CLKRUN# signal high for four clocks. If no other device in the system denies the request to stop before the 5th PCI clock, then PIIX4 asserts the PCI_STP# signal to the clock synthesizer to gate the PCI Clocks to the system.

PIIX4 should always receive a PCI clock even after the clocks have been stopped to the rest of the system. The clock synthesizer must have one non-gated PCI clock signal routed to PIIX4. The clock synthesizer must follow the timing diagrams shown in Figure 18 and Figure 19 for stopping and starting the PCI clocks.

11.3. Peripheral Device Management

The PIIX4 Peripheral Device Management mechanisms provide means to detect an idle peripheral device and to trap accesses to a peripheral device that has been powered down. Device activity can also reload the Global Standby Timer or can generate a Clock Control Stop Break or Burst Event. Device Accesses (I/O or Memory) are monitored from the PCI bus. For devices that sit on the ISA Bus, these accesses can be

forwarded to the ISA Bus. PIIX4 contains 14 separate device monitors, each capable of detecting activity for a different type of device. Figure 14 illustrates the logic associated with each device monitor.

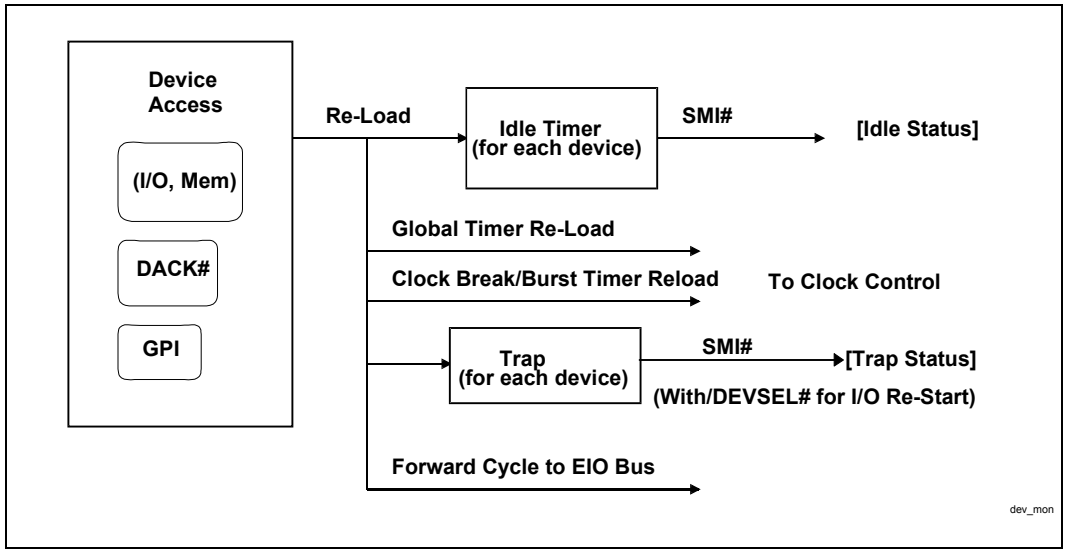


Figure 14. Device Monitoring Logic

11.3.1. DEVICE IDLE TIMER

Each Device has an Idle Timer (except Device 12 and 13) that can be reloaded by activity on that device. Individual device monitors are configured to monitor a specific type of device, such as an IDE hard drive or the audio subsystem. Activity indication is specific to each device and can include the following:

- **Device Access.** Specific I/O or Memory ranges associated with that device are monitored on the PCI bus. Most devices have multiple options to allow for a wide range of system configurations.
- **DMA Acknowledge.** DACK# used for DMA transfers by the device, if applicable (Audio, Floppy, LPT).
- **General Purpose Input.** Most device monitor can watch for assertion of a specific General Purpose Input (GPI) pin. Each GPI signal can have its assertion polarity modified to be high or low. Two GPI signals (device 12 and 13) can also be enabled for edge transition detection.
- **System Activity.** Miscellaneous activity such as Keyboard or Mouse interrupt, PCI bus Master activity, or PCI bus utilization (used to monitor for graphics activity) may be monitored for specific devices.

A device access, DACK# assertion, or GPI signal can be enabled to reload the device's Idle Timer as well as to reload the Global Standby Timer or the Fast Burst or Slow Burst Timers.

Some of the device monitors can serve multiple functions. For example, the Device 3 IDE Secondary IDE Drive 1 monitor can also be enabled as a programmable Software Timer. The Device 8 LPT monitor can be enabled to monitor parallel port activity or PCI Bus Master activity.

When the Idle Timer expires (due to no reload activity), an Idle Status bit is set and an SMI# is generated if enabled. The power management software can then place the idle device into a power managed condition. The idle timers stop counting when the [SM_FREEZE] bit is set. This can be used to keep the idle timers from counting down when the system is executing an SMI routine.

11.3.2. DEVICE TRAP

Each device monitor can enable an IO Trap so that when software makes an access to the enabled I/O or memory range a Trap Status bit is set and an SMI# is generated if enabled. The I/O trap SMI# is synchronous to the completion of the I/O instruction in the processor. The I/O instruction is completed when the BRDY# is returned to the processor. PIIX4 coordinates the assertion of SMI# to the processor with the generation of BRDY# to the processor from the Host Bridge such that the SMI# is generated at least
3 HCLKs before Ready is generated. This allows the processor to perform an IO restart cycle. If the device to be trapped is a PCI device, PIIX4 must be enabled to claim the cycle so that the SMI# can be generated synchronously. The device should be programmed to send the I/O access cycle to the ISA bus where it is terminated normally (although read cycles will return unknown data).

11.3.3. PERIPHERAL DEVICE MANAGEMENT SEQUENCE

Following is a brief description of the peripheral device power management process.

- **Setup:** The system's power management software initializes the access I/O address ranges and the Idle Timer counter for each peripheral device.
- **On-to-Off Transition**: When power management software enables the Idle Timer for that device, the Idle Timer begins to count down. Any access to a peripheral device's I/O address (also DACK# or GPI assertion if enabled) reloads that device's Idle Timer. If the peripheral device's Idle Timer expires, the Idle Status bit is set, and an SMI# is generated. The power management software (SMI# handler) identifies the device by the status bit, then puts the peripheral device into a low power state, disables the Idle Timer hardware, and enables the I/O Trap hardware.
- **Off-to-On Transition**: When the system requires an I/O access to that device range, the access is trapped, an SMI# is generated, and the corresponding I/O Trap SMI status bit is set. The power management software determines which device was accessed, restores the peripheral device to the "on" state, clears the Trap SMI status bits, and then enables the Idle Timer hardware. The processor then issues an I/O restart to access the device again.

11.3.4. DEVICE LOCATION ON PCI BUS OR ISA BUS

Most of the peripheral devices have the capability to exist on the PCI Bus or the ISA/EIO Bus. However, PIIX4 does not support RTC or Keyboard Controller to be resided on the PCI Bus. The Device Activity Monitor is watching cycles on the PCI bus to generate activity events. The device monitors also can be enabled to forward cycles to the device's enabled addresses to the ISA bus. Devices that reside on the ISA Bus must have both address ranges selected and enabled AND the ISA/EIO forwarding enabled.

PCI accesses to external IDE devices on the PCI bus do not generate power management events (Idle timer reloads, global standby timer reloads, burst timer reloads, I/O traps). Power management of external
PCI-based IDE devices must use other means to monitor the activity of those devices. On next page are the following methods for system BIOS to use to monitor external PCI-based IDE devices:

1. If there is a need to monitor accesses to the IDE controller to keep the global standby timer from expiring, then the IRQs should be enabled (GRLD_EN_IRQ) as a reload event for the global standby timer.
2. If there is a need to monitor an external IDE controller for idleness, use the following algorithm
   a. Disable the external IDE controller. Set the PIIX4 to trap on the IDE access and enable the internal IDE controller.
   b. When the SMI is generated, the idle timer can be started, the internal IDE controller disabled, and the instruction redone to the external IDE controller. The IDE device is then assumed to be active during idle timer count down.
   c. When the idle timer times out, an SMI is generated and the PIIX4 should again be set to trap, the external IDE device disabled, and the idle timer started.
   d. If the idle timer times out before the trap occurs, then the external IDE controller is idle and can be put into a lower power mode. The PIIX4 is then set up to trap as in number 3, below.
   e. If the trap occurs first, the IDE device is not idle. The BIOS then returns to step b, above.
3. If there is a need to perform I/O trapping on an external IDE controller, set the PIIX4 to trap on the IDE access and enable the PIIX4 internal IDE controller. When the SMI is generated, the internal IDE controller can be disabled, the external controller enabled, and the I/O cycle restarted.

Table 33. Peripheral Device Overview

	Device Access			Reload			
Peripheral Device	**Access Address**	**DACK#**	**GPI**	**Idle Timer Count**	**Global Standby**	**Fast Burst**	**Slow Burst**
0. Primary IDE Drive 0	1F0h–1F7h 3F6h	IDE PDDACK#		CNT-A	x		x
1. Primary IDE Drive 1	1F0h–1F7h 3F6h	IDE PDDACK#	GPI5	CNT-A	x	x	x
2. Secondary IDE Drive 0	170h–177h 376h	IDE SDDACK#	GPI6	CNT-A	x	x	x
3. Secondary IDE Drive 1 (Software SMI Timer)	170h–177h 376h	IDE SDDACK#	GPI0	SWCNT	x	x	x
4. Audio	300h–303h MIDI 310h–313h MIDI 320h–323h MIDI 330h–333h MIDI 200h–207h GAME 388h–38Bh ADLIB 220h–233h SB-8/16 240h–253h SB-8/16 260h–273h SB-8/16 280h–293h SB-8/16 530h–537h MSS 604h–60Bh MSS E80h–E87h MSS F40h–F47h MSS	any/all: DACK[x]# 0,1,3,5,6,7	GPI13	CNT-B	x	x	
5. FDD	3F0h–3F5h, 3F7h 370h–375h, 3F7h	DACK2#	GPI14	CNT-B	x	x	x

Table 33. Peripheral Device Overview

	Device Access			Reload			
Peripheral Device	**Access Address**	**DACK#**	**GPI**	**Idle Timer Count**	**Global Standby**	**Fast Burst**	**Slow Burst**
6. Serial A (Modem)	3F8h–3FFh COM1 2F8h–2FFh COM2 220h–227h 228h–22Fh 238h–23Fh 2E8h–2EFh COM4 338h–33Fh 3E8h–3EFh COM3		GPI15	CNT-B	x	x	
7. Serial B (IR)	3F8h–3FFh COM1 2F8h–2FFh COM2 220h–227h 228h–22Fh 238h–23Fh 2E8h–2EFh COM4 338h–33Fh 3E8h–3EFh COM3		GPI16	CNT-B	x	x	
8. LPT	LPT_DEC_SEL:3 0,0=3BCh–3BFh, 7BCh–7BEh, 0,1=378h–37Fh, 778h–77Ah, 1,0=278h–27Fh, 678h–67Ah,	one of: DACK[x]# 0,1,3	GPI17	BM_CNT	x	x	
(8.) Bus Master Activity		PCIREQ[A:D]#, PHOLD#		Dev8 Timer	x	x	
9. Generic I/O Range 0	16-byte I/O range		GPI4	CNT-C	x	x	
10. Generic I/O Range 1	16-byte I/O range		GPI18	CNT-C	x	x	
11. User Interface Graphics Keyboard Mouse PCI Utilization	1M to 8M Mem range A0000h–BFFFFh 3B0h–3DFh VGA 60h, 64h IRQ0, IRQ12/M		GPI19	CNT-D	x	x	
12. CARDBUS 0	16-byte I/O range 32-KB–4-MB Mem range		GPI20		x	x	
13. CARDBUS 1	16-byte I/O range 32-KB–4-MB Mem range		GPI21		x	x	

11.3.5. DEVICE SPECIFIC DETAILS

For each device monitor, the system events which can cause actions such as timer reloads or IO traps are listed. The power management resources affected by these events are also listed. The names of register bits which are programmed to enable power management resources or status bits set when events occur are shown in brackets for each item. More specific information can be found in the register bit descriptions.

11.3.5.1. Device 0: IDE Primary Drive 0

Device 0 monitors only the Primary IDE device, drive 0. The IDE device DRV bit (bit 4 of port 1F6h) is shadowed to determine if drive 0 is active on the primary connector.

Device 0 System Events:

- PCI accesses to IO address 1F0–1F7h, 3F6h, independent of IDE enable in PCI function 1, if IDE drive 0 is active. This allows monitoring of devices on PCI or ISA bus. This can cause idle, burst, or global standby timer reloads or IO trap SMI#.
- PDDACK# assertion if primary IDE drive 0 is active and BMIDE is active for primary connector. This can cause only idle, burst and global standby timer reloads.
- There are no General Purpose Inputs associated with Device 0.

Device 0 Idle Timer:

Resolution: 1 second or 8 second	[IDL_SEL_DEV0]
Count: 4 bit	[IDL_CNTA]

Device 0 Idle Timer Reload:	[IDL_EN_DEV0]	
Global Standby Timer Reload:	[GRLD_EN_DEV0]	
Burst Timer Reload (Slow Burst Only):	[BRLD_EN_DEV0]	
Idle Timer Expiration SMI#:	[IDL_EN_DEV0]	[IDL_STS_DEV0]
IO Trap SMI#:	[TRP_EN_DEV0]	[TRP_STS_DEV0]

11.3.5.2. Device 1: IDE Primary Drive 1

Device 1 monitors the Primary IDE device, drive 1 and GPI5. The IDE device DRV bit (bit 4 of port 1F6h) is shadowed to determine if drive 1 is active on the primary connector.

Device 1 System Events:

- PCI accesses to IO address 1F0–1F7h, 3F6h, independent of IDE enable in PCI function 1, if IDE drive 1 is active. This can cause idle, burst or global standby timer reloads or IO trap SMI#.
- PDDACK# assertion if primary IDE drive 1 is active, the IDE interface is configured as primary and secondary and BMIDE is active for primary channel. This can cause only idle, burst, and global standby timer reloads.
- Assertion of GPI5. The polarity of active signal (high or low) is selectable. This can cause idle, burst, or global standby timer reloads or IO trap SMI#.

Device 1 Idle Timer:

Resolution: 1 second or 8 second	[IDL_SEL_DEV1]
Count: 4 bit	[IDL_CNTA]
GPI Enable:	[GPI_EN_DEV1]
GPI Polarity Select:	[GPI_POL_DEV1]
Device 1 Idle Timer Reload:	[IDL_EN_DEV1]
Global Standby Timer Reload:	[GRLD_EN_DEV1]
Burst Timer Reload:	[BRLD_EN_DEV1]
Fast or Slow Burst Select:	[BRLD_SEL_DEV1]
Idle Timer Expiration SMI#:	[IDL_EN_DEV1] [IDL_STS_DEV1]
Trap SMI#:	[TRP_EN_DEV1] [TRP_STS_DEV1]

11.3.5.3. Device 2: IDE Secondary Drive 0

Device 2 monitors the Secondary IDE device, drive 0 and GPI6. The IDE device DRV bit (bit 4 of port 176h) is shadowed to determine if drive 0 is active on the secondary connector.

Device 2 System Events:

- PCI accesses to IO address 170–177h, 376h, independent of IDE enable in PCI function 1, if secondary IDE drive 0 is active. This can cause idle, burst, or global standby timer reloads or IO trap SMI#.
- SDDACK# assertion if secondary IDE drive 0 is active, the IDE interface is configured as primary and secondary and BMIDE is active for secondary channel. This can cause only idle, burst, and global standby timer reloads.
- Assertion of GPI6. The polarity of active signal (high or low) is selectable. This can cause idle, burst, or global standby timer reloads or IO trap SMI#.

Device 2 Idle Timer:

Resolution: 1 second or 8 second	[IDL_SEL_DEV2]
Count: 4 bit	[IDL_CNTA]
GPI Enable:	[GPI_EN_DEV2]
GPI Polarity Select:	[GPI_POL_DEV2]
Device 2 Idle Timer Reload:	[IDL_EN_DEV2]
Global Standby Timer Reload:	[GRLD_EN_DEV2]
Burst Timer Reload:	[BRLD_EN_DEV2]
Fast or Slow Burst Select:	[BRLD_SEL_DEV2]
Idle Timer Expiration SMI#:	[IDL_EN_DEV2] [IDL_STS_DEV2]
Trap SMI#:	[TRP_EN_DEV2] [TRP_STS_DEV2]

11.3.5.4. Device 3: IDE Secondary Drive 1

Device 3 monitors the Secondary IDE device, drive 1 and GPI0. The IDE device DRV bit (bit 4 of port 176h) is shadowed to determine if drive 1 is active on the secondary connector. Device 3 can also be used as a Software SMI# Timer. It has a configuration bit to disable the Idle Timer Reload so that the timer can be allowed to expire based only on the timer count.

Device 3 System Events:

- PCI accesses to IO address 170–177h, 376h, independent of IDE enable in PCI function 1, if secondary IDE drive 1 is active. This can cause idle, burst, or global standby timer reloads or IO trap SMI#.
- SDDACK# assertion if secondary IDE drive 1 is active, the IDE interface is configured as primary and secondary and BMIDE is active for secondary channel. This can cause only idle, burst, and global standby timer reloads.
- Assertion of GPI0. The polarity of active signal (high or low) is selectable. This can cause idle, burst, or global standby timer reloads or IO trap SMI#.

Device 3 Idle Timer:		
Resolution: 1 msec or 8 sec	[IDL_SEL_DEV3]	
Count: 4 bit	[SW_CNT]	
GPI Enable:	[GPI_EN_DEV3]	
GPI Polarity Select:	[GPI_POL_DEV3]	
Device 3 Idle Timer Reload:	[IDL_EN_DEV3]	
Idle Timer Reload Disable (SW):	[IDL_RLD_EN_DEV3]	
Global Standby Timer Reload:	[GRLD_EN_DEV3]	
Burst Timer Reload:	[BRLD_EN_DEV3]	
Fast or Slow Burst Select:	[BRLD_SEL_DEV3]	
Idle Timer Expiration SMI#:	[IDL_EN_DEV3]	[IDL_STS_DEV3]
Trap SMI#:	[TRP_EN_DEV3]	[TRP_STS_DEV3]

11.3.5.5. Device 4: Audio

Device 4 monitors an audio subsystem and GPI13. The available address ranges encompass the following types of audio devices, 8/16-bit Sound Blaster, standard Game Port, ADLIB music synthesizer, Microsoft Sound System, and MIDI. The actual address ranges selectable for each type is shown below.

Device 4 System Events:

- PCI accesses to any of the enabled IO addresses (see below). This can cause idle, burst, or global standby timer reloads, IO trap SMI#, or forwarding of the cycle from PCI to ISA.
- DACKx# assertion (x=0,1,3,5,6,7) if enabled (see below). This can cause only idle, burst, and global standby timer reloads.
- Assertion of GPI13. The polarity of active signal (high or low) is selectable. This can cause idle, burst, or global standby timer reloads.

Device 4 Address Ranges:

Sound Blaster 8/16:	220–22Fh, 230–233h or 240–24Fh, 250–253h or 260–26Fh, 270–273h or 280–28Fh, 290–293h		[SB_EN] [SB_SEL]
Game Port:	200–207h		[GAME_EN]
ADLIB Synthesizer:	388–38Bh	[SB_EN]	
Microsoft Sound System:	530–537h or 604–60Bh or E80–E87h or F40–F47h		[MSS_EN] [MSS_SEL]
MIDI:	300–303h or 310–313h or 320–323h or 330–333h		[MIDI_EN] [MIDI_SEL]

Device 4 Idle Timer:
Resolution: 1 second
Count: 5 bit [IDL_CNTB]

GPI Enable: [GPI_EN_DEV4]
GPI Polarity Select: [GPI_POL_DEV4]
DACKx# Enables: [DACKx_EN_DEV4] where x=0,1,3,5,6,7
Device 4 ISA Forwarding Enable: [EIO_EN_DEV4]
Device 4 Idle Timer Reload: [IDL_EN_DEV4]
Global Standby Timer Reload: [GRLD_EN_DEV4]
Burst Timer Reload (Fast Burst Only): [BRLD_EN_DEV4]
Idle Timer Expiration SMI#: [IDL_EN_DEV4] [IDL_STS_DEV4]
Trap SMI#: [TRP_EN_DEV4] [TRP_STS_DEV4]

11.3.5.6. Device 5: Floppy Disk Drive

Device 5 monitors accesses to Floppy Drive Controller or GPI14.

Device 5 System Events:

- PCI accesses to IO addresses for the floppy drive, selectable below. This can cause idle, burst, or global standby timer reloads, IO trap SMI#, or forwarding of the cycle from PCI to ISA.
- DACK2# assertion if enabled (see below). This can cause only idle, burst, and global standby timer reloads.
- Assertion of GPI14. The polarity of active signal (high or low) is selectable. This can cause idle, burst, or global standby timer reloads.

Device 5 Address Ranges:

Floppy Drive:	3F0–3F5h, 3F7h or 370–375h, 377h	[FDC_MON_EN] [FDC_DEC_SEL]

Device 5 Idle Timer:

Resolution: 1 second	
Count: 5 bit	[IDL_CNTB]
GPI Enable:	[GPI_EN_DEV5]
GPI Polarity Select:	[GPI_POL_DEV5]
DACK2# Enable:	[RES_EN_DEV5]
Device 5 ISA Forwarding Enable:	[EIO_EN_DEV5]
Device 5 Idle Timer Reload:	[IDL_EN_DEV5]
Global Standby Timer Reload:	[GRLD_EN_DEV5]
Burst Timer Reload:	[BRLD_EN_DEV5]
Fast or Slow Burst Select:	[BRLD_SEL_DEV5]
Idle Timer Expiration SMI#:	[IDL_EN_DEV5] [IDL_STS_DEV5]
Trap SMI#:	[TRP_EN_DEV5] [TRP_STS_DEV5]

11.3.5.7. Device 6: Serial Port A

Device 6 monitors accesses to Serial Port A or GPI15. Device 7 also monitors serial port resources. This gives the capability to monitor two separate serial ports in a system.

Device 6 System Events:

- PCI accesses to IO addresses for a serial port, selectable below. This can cause idle, burst, or global standby timer reloads, IO trap SMI#, or forwarding of the cycle from PCI to ISA.
- Assertion of GPI15. The polarity of active signal (high or low) is selectable. This can cause idle, burst, or global standby timer reloads.

Device 6 Address Ranges:

Serial Port A:	3F8–3FFh or 2F8–2FFh or 220–227h or 228–22Fh or 238–23Fh or 2E8–2EFh or 338–33Fh or 3E8–3EFh	[SA_MON_EN] [COMA_DEC_SEL]
Device 6 Idle Timer:		
Resolution: 1 second		
Count: 5 bit	[IDL_CNTB]	
GPI Enable:	[GPI_EN_DEV6]	
GPI Polarity Select:	[GPI_POL_DEV6]	
Device 6 ISA Forwarding Enable:	[EIO_EN_DEV6]	
Device 6 Idle Timer Reload:	[IDL_EN_DEV6]	
Global Standby Timer Reload:	[GRLD_EN_DEV6]	
Burst Timer Reload (Fast Burst Only):	[BRLD_EN_DEV6]	
Idle Timer Expiration SMI#:	[IDL_EN_DEV6]	[IDL_STS_DEV6]
Trap SMI#:	[TRP_EN_DEV6]	[TRP_STS_DEV6]

11.3.5.8. Device 7: Serial Port B

Device 7 monitors accesses to Serial Port B or GPI16. Device 6 also monitors serial port resources. This gives the capability to monitor two separate serial ports in a system.

Device 7 System Events:

- PCI accesses to IO addresses for a serial port, selectable below. This can cause idle, burst, or global standby timer reloads, IO trap SMI#, or forwarding of the cycle from PCI to ISA.
- Assertion of GPI16. The polarity of active signal (high or low) is selectable. This can cause idle, burst, or global standby timer reloads.

Device 7 Address Ranges:

Serial Port B:	3F8–3FFh or 2F8–2FFh or 220–227h or 228–22Fh or 238–23Fh or 2E8–2EFh or 338–33Fh or 3E8–3EFh	[SB_MON_EN] [COMB_DEC_SEL]
Device 7 Idle Timer:		
Resolution: 1 second Count: 5 bit	[IDL_CNTB]	
GPI Enable:	[GPI_EN_DEV7]	
GPI Polarity Select:	[GPI_POL_DEV7]	
Device 7 ISA Forwarding Enable:	[EIO_EN_DEV7]	
Device 7 Idle Timer Reload:	[IDL_EN_DEV7]	
Global Standby Timer Reload:	[GRLD_EN_DEV7]	
Burst Timer Reload (Fast Burst Only):	[BRLD_EN_DEV7]	
Idle Timer Expiration SMI#:	[IDL_EN_DEV7]	[IDL_STS_DEV7]
Trap SMI#:	[TRP_EN_DEV7]	[TRP_STS_DEV7]

11.3.5.9. Device 8: LPT (Parallel Port)

Device 8 monitors accesses to Parallel Port or GPI17. It can also be used to monitor for PCI Bus Master activity (PIIX4 or any other PCI Master).

Device 8 System Events:

- PCI accesses to IO addresses for a parallel port, selectable below. This can cause idle, burst, or global standby timer reloads, IO trap SMI#, or forwarding of the cycle from PCI to ISA.
- DACKx# assertion (x=0,1,3) if enabled (see below). This can cause idle, burst, and global standby timer reloads.
- Assertion of GPI17. The polarity of active signal (high or low) is selectable. This can cause idle, burst, or global standby timer reloads.
- Assertion of PCIREQ[0:3] or PHOLD#, signifying PCI Master activity. This can cause idle, burst, or global standby timer reloads or IO trap SMI#. The Bus Master activity can be programmed to cause a Trap SMI# independently of IO address accesses.

Device 8 Address Ranges:		
LPT (Parallel Port):	378–37Fh, 778–77Ah or 278–27Fh, 678–67Ah or 3BC–3BFh, 7BC–7BEh	[LPT_MON_EN] [LPT_DEC_SEL]
Device 8 Idle Timer:		
Resolution: 1 msec or 1 sec	[IDL_SEL_DEV8]	
Count: 5 bit	[BM_CNT]	
GPI Enable:	[GPI_EN_DEV8]	
GPI Polarity Select:	[GPI_POL_DEV8]	
DACKx# Enables:	[RES_EN_DEV8]	
DACKx# Select (DACKx#=0,1,3):	[LPT_DMA_SEL]	
Device 8 ISA Forwarding Enable:	[EIO_EN_DEV8]	
Device 8 Idle Timer Reload:	[IDL_EN_DEV8]	
Global Standby Timer Reload:	[GRLD_EN_DEV8]	
Burst Timer Reload (Fast Burst Only):		
Decode, DACK GPI:	[BRLD_EN_DEV8]	
Above and Bus Master:	[BM_RLD_DEV8]	
Bus Master Only:	[BRLD_EN_BM]	
Idle Timer Expiration SMI#:	[IDL_EN_DEV8]	[IDL_STS_DEV8]
Trap SMI# (LPT or GPI only):	[TRP_EN_DEV8]	[TRP_STS_DEV8]
Trap SMI# (Bus Master only):	[BM_TRP_EN]	[BM_STS]

11.3.5.10. Device 9: Generic I/O Device 0

Device 9 monitors a device on the PCI bus with a programmable IO address or GPI4.

Device 9 System Events:

- PCI accesses to programmable IO addresses, selectable below. This can cause idle, burst, or global standby timer reloads, IO trap SMI#, or forwarding of the cycle from PCI to ISA. It can optionally generate a Chip Select signal (PCS0#). The address consists of a 16-bit base address and 4-bit mask, allowing an address range from 1 to 16 bytes.
- Assertion of GPI4. The polarity of active signal (high or low) is selectable. This can cause idle, burst, or global standby timer reloads.

Device 9 Address Ranges: [GDEC_MON_DEV9]
Programmable Base Address: [BASE_DEV9]
Programmable Mask: [MASK_DEV9]
Device 9 Idle Timer:
Resolution: 1 second
Count: 5 bit [IDL_CNTC]
GPI Enable: [GPI_EN_DEV9]
GPI Polarity Select: [GPI_POL_DEV9]
Device 9 ISA Forwarding Enable: [EIO_EN_DEV9]
Chip Select (PCS0#) Enable: [CS_EN_DEV9]
Device 9 Idle Timer Reload: [IDL_EN_DEV9]
Global Standby Timer Reload: [GRLD_EN_DEV9]
Burst Timer Reload (Fast Burst Only): [BRLD_EN_DEV9]
Idle Timer Expiration SMI#: [IDL_EN_DEV9] [IDL_STS_DEV9]
Trap SMI#: [TRP_EN_DEV9] [TRP_STS_DEV9]

11.3.5.11. Device 10: Generic I/O Device 1

Device 10 monitors a generic I/O device with a programmable IO address or GPI18.

Device 10 System Events:

- PCI accesses to programmable IO addresses, selectable below. This can cause idle, burst, or global standby timer reloads, IO trap SMI#, or forwarding of the cycle from PCI to ISA. It can optionally generate a Chip Select signal (PCS1#). The address consists of a 16-bit base address and 4-bit mask, allowing an address range from 1 to 16 bytes.
- Assertion of GPI18. The polarity of active signal (high or low) is selectable. This can cause idle, burst, or global standby timer reloads.

Device 10 Address Ranges: [GDEC_MON_DEV10]
Programmable Base Address: [BASE_DEV10]
Programmable Mask: [MASK_DEV10]
Device 10 Idle Timer:
Resolution: 1 second
Count: 5 bit [IDL_CNTC]
GPI Enable: [GPI_EN_DEV10]
GPI Polarity Select: [GPI_POL_DEV10]
Device 10 ISA Forwarding Enable: [EIO_EN_DEV10]
Chip Select (PCS1#) Enable: [CS_EN_DEV10]
Device 10 Idle Timer Reload: [IDL_EN_DEV10]
Global Standby Timer Reload: [GRLD_EN_DEV10]
Burst Timer Reload (Fast Burst Only): [BRLD_EN_DEV10]
Idle Timer Expiration SMI#: [IDL_EN_DEV10] [IDL_STS_DEV10]
Trap SMI#: [TRP_EN_DEV10] [TRP_STS_DEV10]

11.3.5.12. Device 11: User Interface (Keyboard, Mouse, Video)

Device 11 monitors the system's primary user interfaces, the keyboard, a PS/2 mouse, or the video subsystem. It contains special logic to monitor the PCI bus utilization in order to detect video activity. This will allow a system to playback video without power managing the video subsystem due to user inactivity (no keyboard or mouse movement).

Device 11 System Events:

- PCI accesses to programmable linear frame buffer addresses, selectable below. The linear frame buffer address consists of a 12-bit base address and 2-bit mask, allowing a frame buffer address range from 1 Mbyte to 4 Mbytes. This can cause burst timer reloads only.
- PCI accesses to VGA I/O addresses (3B0–3DFh) or the A and B segment video memory ranges (A000–BFFFh). This can cause burst timer reloads only.
- PCI accesses to Keyboard Controller I/O addresses (60h, 64h). This can cause idle, burst, or global standby timer reloads, IO trap SMI#, or forwarding KBC cycles to ISA.
- PCI bus utilization is monitored and a system event is generated if the number of PCI data phases (as measured by FRAME# assertion) exceeds a set limit. This can cause idle or global standby timer reloads.
- Assertion of GPI19. The polarity of active signal (high or low) is selectable. This can cause idle, burst, or global standby timer reloads or IO trap SMI#.
- Assertion of IRQ1 or IRQ12/M. These can cause idle, burst, or global standby timer reloads or IO trap SMI#.

Device 11 Linear Frame Buffer Ranges:	[LFB_DEC_EN]	
Programmable Base Address:	[LFBASE_DEV11]	
Programmable Mask:	[LFMASK_DEV11]	
Device 11 VGA Decode:	[GRAPH_IO_EN]	
Device 11 A,B Segment Decode:	[GRAPH_AB_EN]	
Device 11 Idle Timer:		
Resolution: 1 second or 1 minute	[IDL_SEL_DEV11]	
Count: 5 bit	[IDL_CNTD]	
GPI Enable:	[GPI_EN_DEV11]	
GPI Polarity Select:	[GPI_POL_DEV11]	
IRQ1 Enable:	[IRQ1_EN_DEV11]	
IRQ12/M Enable:	[IRQ12_EN_DEV11]	
KBC Decode Enable:	[KBC_EN_DEV11]	
Device 11 ISA Forwarding Enable:	[KBC_EIO_EN]	
PCI Bus Utilization Enable:	[VIDEO_EN]	[VIDEO_STS]
Threshold:	[BUS_UTIL]	
Percent Active:	[%BUS_UTIL]	
Device 11 Idle Timer Reload:	[IDL_EN_DEV11]	
Global Standby Timer Reload:	[GRLD_EN_DEV11]	
Burst Timer Reload (Fast Burst Only):	[BRLD_EN_DEV11]	
Idle Timer Expiration SMI#:	[IDL_EN_DEV11]	[IDL_STS_DEV11]
Trap SMI#:	[TRP_EN_DEV11]	[TRP_STS_DEV11]

11.3.5.13. Device 12: Cardbus Slot (or Generic I/O and MEM Device)

Device 12 monitors a generic I/O device or Memory device with a programmable IO or memory address or GPI20. Its operation is the same as Device 13.

Device 12 System Events:

- PCI accesses to programmable IO addresses and memory addresses, selectable below. This can cause burst, or global standby timer reloads, IO trap SMI#, or forwarding of the cycle from PCI to ISA. The IO address consists of a 16-bit base address and 4-bit mask, allowing an IO address range from 1 to 16 bytes. The memory address consists of a 17-bit base address (AD[31:15]) and a 7-bit mask (AD[21:15]). This provides memory ranges from 32 Kbytes to 4 Mbytes in size.
- Assertion of GPI20. The polarity of active signal (high or low) is selectable. It can also be enabled as edge-triggered. This can cause burst, or global standby timer reloads or IO trap SMI#.
- There is no idle timer associated with Device 12.

Device 12 IO Address Range: [IO_EN_DEV12]
Programmable IO Base Address: [IBASE_DEV12]
Programmable IO Mask: [IMASK_DEV12]
Device 12 Memory Address Range: [MEM_EN_DEV12]
Programmable Mem Base Addr: [MBASE_DEV12]
Programmable Memory Mask: [MMASK_DEV12]
Device 12 Idle Timer: NONE
GPI Enable: [GPI_EN_DEV12]
GPI Polarity Select: [GPI_POL_DEV12]
GPI Edge Detect Enable: [GPI_EDG_DEV12]
Device 12 ISA Forwarding Enable: [EIO_EN_DEV12]
Global Standby Timer Reload: [GRLD_EN_DEV12]
Burst Timer Reload (Fast Burst Only): [BRLD_EN_DEV12]
Trap SMI#: [TRP_EN_DEV12] [TRP_STS_DEV12]

11.3.5.14. Device 13: Cardbus Slot (or Generic I/O and MEM Device)

Device 13 monitors a generic I/O device or Memory device with a programmable IO or memory address or GPI21. Its operation is the same as Device 12.

Device 13 System Events:

- PCI accesses to programmable IO addresses and memory addresses, selectable below. This can cause burst, or global standby timer reloads, IO trap SMI#, or forwarding of the cycle from PCI to ISA. The IO address consists of a 16-bit base address and 4-bit mask, allowing an IO address range from 1 to 16 bytes. The memory address consists of a 17-bit base address (AD[31:15]) and a 7-bit mask (AD[21:15]). This provides memory ranges from 32 Kbytes to 4 Mbytes in size.
- Assertion of GPI21. The polarity of active signal (high or low) is selectable. It can also be enabled as edge-triggered. This can cause burst, or global standby timer reloads or IO trap SMI#.
- There is no idle timer associated with Device 13.

Device 13 IO Address Range: [IO_EN_DEV13]
Programmable IO Base Address: [IBASE_DEV13]
Programmable IO Mask: [IMASK_DEV13]
Device 13 Memory Address Range: [MEM_EN_DEV13]
Programmable Mem Base Addr: [MBASE_DEV13]
Programmable Memory Mask: [MMASK_DEV13]
Device 13 Idle Timer: NONE
GPI Enable: [GPI_EN_DEV13]
GPI Polarity Select: [GPI_POL_DEV13]
GPI Edge Detect Enable: [GPI_EDG_DEV13]
Device 13 ISA Forwarding Enable: [EIO_EN_DEV13]
Global Standby Timer Reload: [GRLD_EN_DEV13]
Burst Timer Reload (Fast Burst Only): [BRLD_EN_DEV13]
Trap SMI#: [TRP_EN_DEV13] [TRP_STS_DEV13]

11.4. Suspend/Resume and Power Plane Control

11.4.1. SYSTEM SUSPEND

PIIX4 supports three different Suspend modes. The PIIX4 power management architecture is designed to allow a single system to support multiple suspend modes and to switch between those modes as required. A suspended system can be resumed via a number of different events. It will then return to full operation where it can continue processing or be placed into another suspend mode (potentially a lower power mode than it resumed from).

The standard system usage model for these modes are described here and include Power On Suspend (POS), Suspend to RAM (STR), and Suspend to Disk (STD). This definition allows for other system usage models which use the PIIX4 suspend/resume control signals in other ways. The standard system mode names are used throughout this document. Table 34 summarizes these modes along with their target system power.

Power On Suspend (POS) System Model

All devices are powered up except for the clock synthesizer. The Host and PCI clocks are inactive and PIIX4 provides control signals and 32-kHz Suspend Clock (SUSCLK) to allow for DRAM refresh and to turn off the clock synthesizer. The only power consumed in the system is due to DRAM Refresh and leakage current of the powered devices.

When the system resumes from POS, PIIX4 can optionally resume without resetting the system, can reset the processor only, or can reset the entire system. When no reset is performed, PIIX4 only needs to wait for the clock synthesizer and processor PLLs to lock before the system is resumed. This takes typically 20 ms.

Suspend to RAM (STR)

Power is removed from most of the system components during STR, except the DRAM. Power is supplied to Suspend Refresh logic in the Host Controller, and RTC and Suspend Well logic in PIIX4. PIIX4 provides control signals and 32-kHz Suspend Clock (SUSCLK) to allow for DRAM refresh and to turn off the clock synthesizer and other power planes.

PIIX4 will reset the system on resume from STR.

Suspend to DISK (STD)

Power is removed from most of the system components during STD. Power is maintained to the RTC and Suspend Well logic in PIIX4.

PIIX4 will reset the system on resume from STD.

This state is also called the Soft Off (SOff) state. The difference depends on whether the system state is restored by software to a pre-suspend condition or if the system is rebooted.

Mechanical Off (MOff)

This is not a suspend state. This is a condition where all power except the RTC battery has been removed from the system. It is typically controlled by a mechanical switch turning off AC power to a power supply.
It could be used as a condition in which a mobile system’s main battery has been removed.

Table 34. Different Power Management Modes Supported (Standard System Model)

PM Mode	System Strategy	System Target Power	System Target Resume Latency
Global Standby	All monitored devices are powered off, and the processor's clock is stopped.	Variable	Variable
Powered-on Suspend (POS)	Same as Global Standby, but power is removed from the clock generators.	<250 mW	~20 ms
Suspend to RAM (STR)	Power is removed everywhere in the system except: power management section of PIIX4, slow refresh logic in the memory controller and graphics chips, and the graphics and DRAM memory.	<20 mW	~1 sec
Suspend to Disk/ Soft Off (STD/Soff)	Power is removed everywhere except the power management sections of PIIX4.	<300 uW	< 30 sec

PIIX4 controls the system entering the various suspend states through the suspend control signals listed in Table 35. Upon initiation of suspend, PIIX4 will assert the SUS_STAT[1:2]#, SUSA#, SUSB#, and SUSC# signals in a well defined sequence to switch the system into the desired power state. The SUSA#, SUSB#, and SUSC# signals can be used to control various power planes in the system. The SUS_STAT1# signal is a status signal that indicates to the host bridge when to enter or exit a suspend state, or when to enter or exit a stop clock state (when the system is still running). This is typically used to place the DRAM controller into a Suspend Refresh mode of operation. The SUS_STAT2# signal is a status signal that can be used to indicate to other system devices when to enter or exit a suspend state (like the graphics and Cardbus controllers). See "System Suspend And Resume Control Signaling" section for sequencing details. Note that these signals are associated with a particular type of suspend mode and power plane for descriptive purposes in this section. The system designer can, however, use these signals to control any type of function desired.

The system is placed in a suspend mode by programming the Power Management Control register. The Suspend Type is first programmed and then the Suspend Enable bit is set. This causes PIIX4 to automatically sequence into the programmed suspend mode.

Table 35. Power State Decode

Power State	RSMRST#	SUS_STAT1#	SUS_STAT2#	SUSA#	SUSB#	SUSC#
On	1	x[1]	1	1	1	1
POS	1	0	0	0	1	1
STR	1	0	0	0	0	1
STD/SOFF	1	0	0	0	0	0
Mechanical Off	0	0	0	0	0	0

NOTES:

1. SUS_STAT1# is also used when the system is running. It indicates to the Host-to-PCI bridge when to switch between the normal and suspend refresh mode for DRAM Stop Clock support. In the Stop Clock condition, HCLK is stopped and the Host-to-PCI bridge must run DRAM refresh off the SUSCLK input.

11.4.2. SYSTEM RESUME

PIIX4 can be resumed from either a Suspend or Soft Off state. Depending on the system's suspend state, different features can be enabled to resume the system. There are two classes of resume events, those whose logic resides in the PIIX4 main power well and those whose logic resides in the PIIX4 Suspend well. Those in the Suspend well can resume the system from any Suspend or Soft Off state. Those in the main power well can only resume the system from a Powered On Suspend state. Table 36 lists the suspend states that can be enabled for a particular resume event.

Upon detection of an enabled resume event, PIIX4 will set appropriate status signals and automatically transition its suspend control signals bringing the system into a "full on" condition. The sequencing is shown in the "System Suspend And Resume Control Signaling" section.

Table 36. Resume Events Supported In Different Power States

Resume Event	Suspend States			
	POS	STR	STD/SOff	MOff
RTC Alarm (IRQ8)*	x	x	x	
SMBus Resume Event (Slave Port Match)	x	x	x	
Serial A Ring (RI)	x	x	x	
Power Button (PWRBTN#)	x	x	x	
EXTSMI (EXTSMI#)	x	x	x	
LID (LID)	x	x	x	
GPI1	x	x	x	
GSTBY Timer Expiration	x	x	x	
Interrupt (IRQ[1,3:15])	x			
USB	x			

The various resume events and their programming model are shown here.

System Resume Events

PWRBTN# Asserted:	[PWRBTN_EN]
LID Asserted:	[LID_EN]
— Polarity Select:	[LID_POL]
GPI1 Asserted:	[GPI_EN]
EXTSMI# Asserted:	[EXTSMI_EN]
SMBus Events:	[ALERT_EN]
	[SLV_EN]
	[SHDW1_EN]
	[SHDW2_EN]
Global Standby Timer Expiration:	[GSTBY_EN]
Ring Indicate Assertion (RI#):	[RI_EN]
RTC Alarm (IRQ8):	[RTC_EN]
USB Resume Signaling: (POS Only)	[USB_EN]
IRQ[1,3:7,9:15]: (POS Only)	[IRQ_RSM_EN]

RTC Alarm only supports internal RTC. For external RTC implementations, the IRQ8 must be tied to one of the other resume input signals (GPI1, LID, EXTSMI#,RI#) for the resume functionality.

The PWRBTN# and LID signals are internally debounced for 16 ms. They must be active for greater than this debounce time for the resume to be recognized.

The GPI1, EXTSMI#, IRQ[15:9,7:3,1], and USB resume events must be active for a minimum of 64 μs (approximately 2 RTC clock periods) for the resume to be recognized.

Global Standby Timer Resume

The Global Standby Timer is used to monitor for global system activity during normal operation and can be reloaded by system activity events. Upon expiration, it generates an SMI# (see “System Management” section for additional information on Global Standby Timer). When the system is placed in a Suspend Mode, the Global Standby Timer can be used to generate a resume event. The Global Standby Timer can enable four different timer resolutions for wake-up times from approximately 30 seconds to 8.5 hours. This allows the system to transition into a lower power suspend state.

11.4.3. SYSTEM SUSPEND AND RESUME CONTROL SIGNALING

PIIX4 automatically controls the signals required to transition the system between the various power states. It provides control for Host and PCI clocks, main memory and video memory refresh, system power plane control, and system reset.

Figure 15–Figure 17 show the system timings for changing the power states of a system using the standard POS/STR/STD models. The table notes provide information on how these signals operate in non-standard modes of operation (e.g., Suspend to RAM mode without removing Core well power).

11.4.3.1. Power Supply Timings

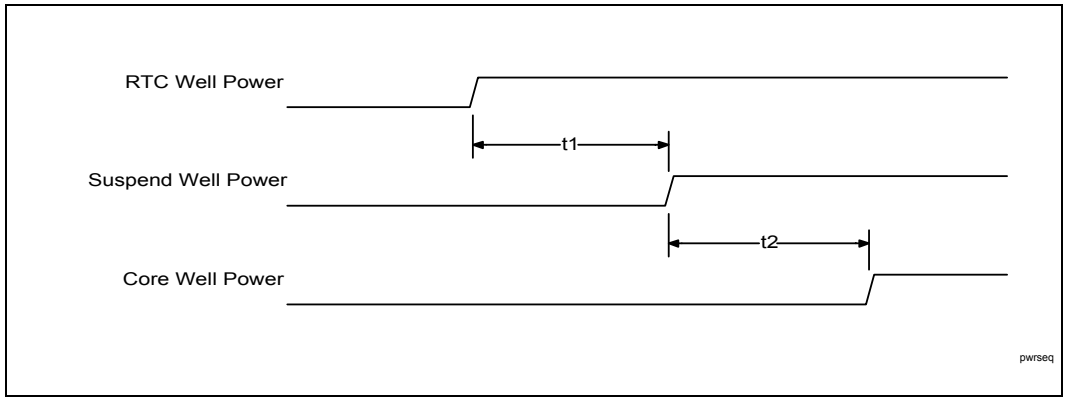


Figure 15. PIIX4 Power Well

Table 37. PIIX4 Power Well Timings

Sym	Parameter	Min	Max	Unit	Notes
t1	RTC Well Power to Suspend Well Power	0		ns	
t2	Suspend Well Power to Core Well Power	0		ns	

11.4.3.2. Power Level Active Status Signal Timings

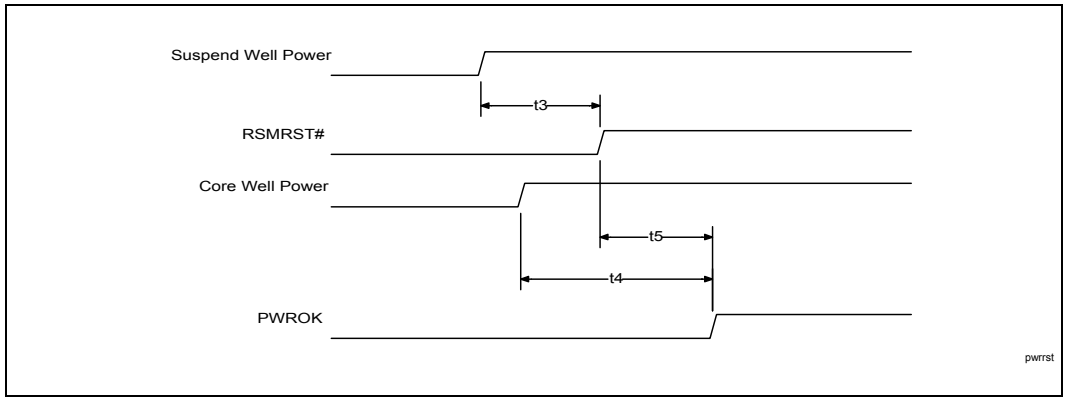


Figure 16. RSMRST# and PWROK Timings

Table 38. Power Supply Timings

Sym	Parameter	Min	Max	Unit	Notes
t3	Suspend Well Power to RSMRST# Inactive	1		ms	
t4	Core Well Power to PWROK Active	1		ms	
t5	RSMRST# Inactive to PWROK Active	0		ns	

11.4.3.3. Power Management Signal Timings (Powered From Suspend Power Well)

This section shows the timing relationships for PIIX4 power management signals that are powered from the Suspend Power well. These timings hold independent of the condition of Core Well power or the PWROK signal.

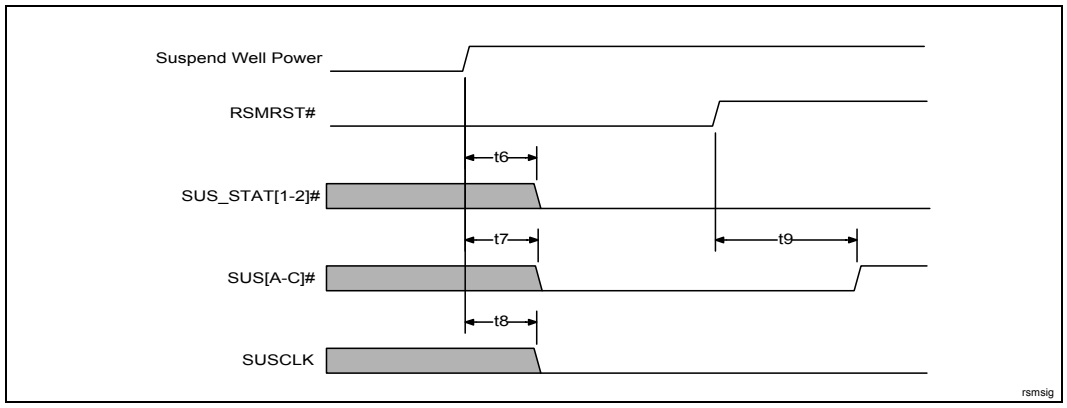


Figure 17. Suspend Well Power and RSMRST# Activated Signals

Table 39. Suspend Well Power and RSMRST# Activated Signals Timings

Sym	Parameter	Min	Max	Unit	Notes
t6	Resume Well Power and RSMRST# Active to SUS_STAT[1:2]# Active		1	RTC	1
t7	Resume Well Power and RSMRST# Active to SUS[A:C]# Active		1	RTC	1
t8	Resume Well Power and RSMRST# Active to SUSCLK Low		1	RTC	1
t9	RSMRST# Inactive to SUS[A:C]# Inactive	1	2	RTC	1

NOTES:
1. These signals are controlled off an internal RTC clock. 1 RTC unit is approximately 32 µs.

11.4.3.4. PCI Clock Stop and Start Timing Relationships

Figure 18 and Figure 19 show the timing relationship for the control of system PCICLK. The system PCICLK timings must be followed exactly for proper operation of PC/PCI DMA or Serial IRQ logic. If PC/PCI DMA and Serial IRQs are not used in the system, the system PCICLK stop timings must meet the system developer's requirements.

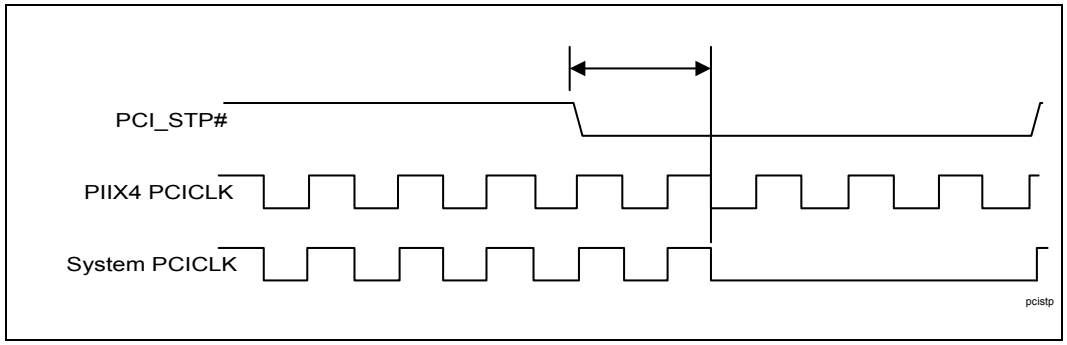


Figure 18. PCI Clock Stop Timing

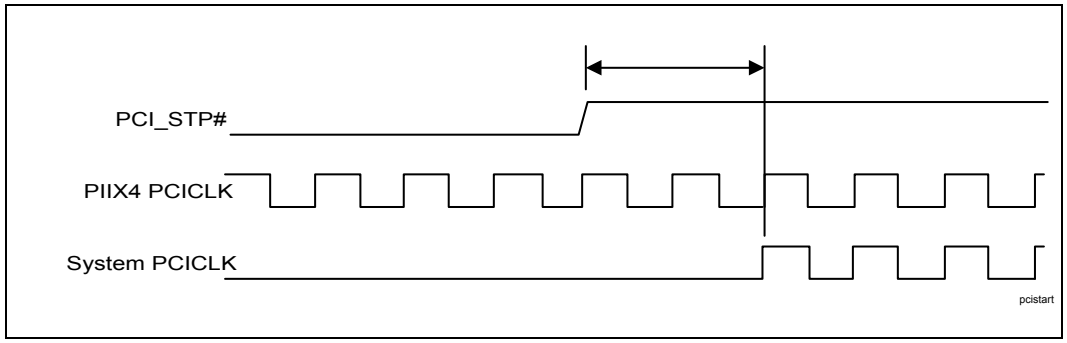


Figure 19. PCI Clock Start Timing

11.4.3.5. Power Management Signal Timings (Powered From PIIX4 Main Core Well)

Figure 20 shows the timing relations for Power Management signals powered from the PIIX4 Core well. Here the power active status signals (RSMRST# and PWROK) transition after the application of all power to PIIX4. It can be applied to situations where two or more of the PIIX4 power planes are connected together. It also shows timings when RSMRST# and PWROK are connected together.

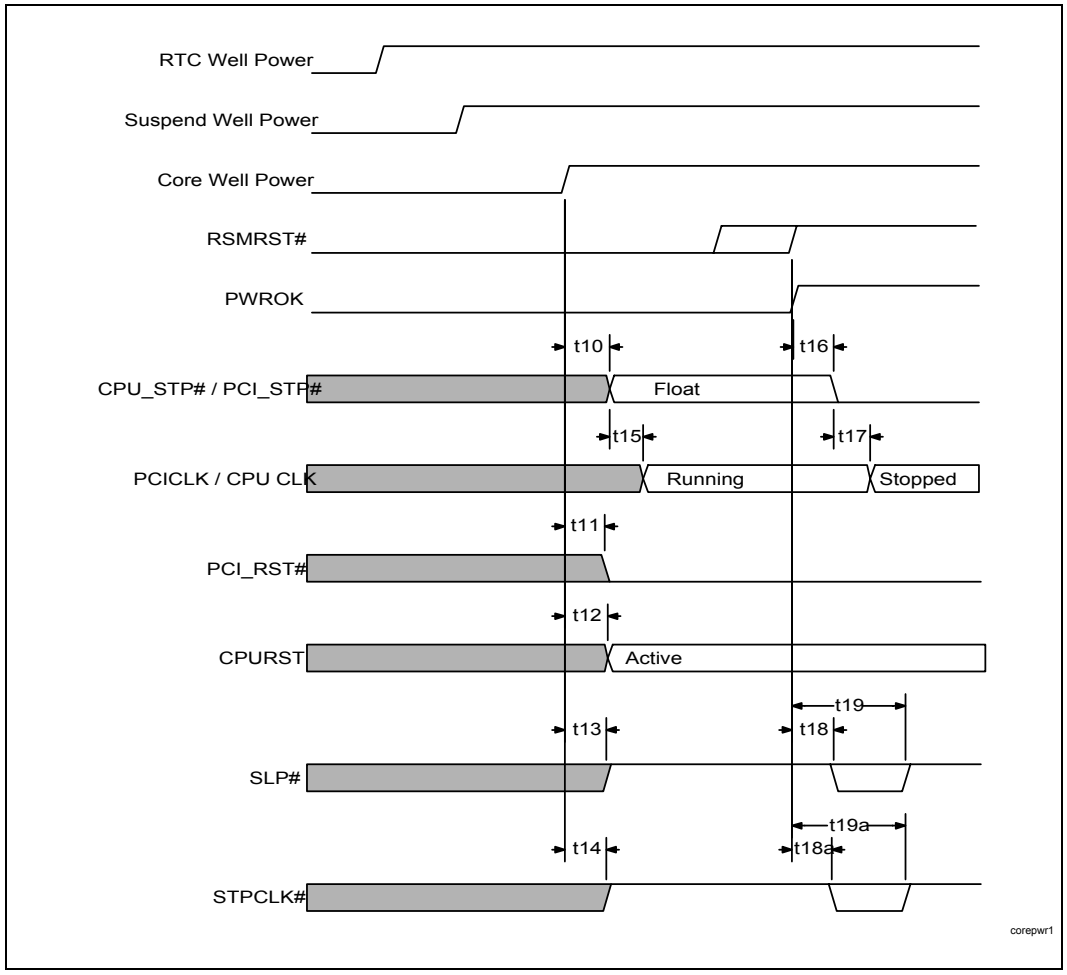


Figure 20. Core Well Power and PWROK Activated Signals (Core Well Power Applied Before RSMRST# Inactive)

Table 40. Core Well Power and PWROK Activated Signals Timings

Sym	Parameter	Min	Max	Unit	Notes
t10	Core Well Power and PWROK Inactive to CPU_STP# and PCI_STP# Float		1	RTC	1
t11	Core Well Power and PWROK Inactive to PCIRST# Active		1	RTC	1
t12	Core Well Power and PWROK Inactive to CPURST Active		1	RTC	1
t13	Core Well Power and PWROK Inactive to SLP# Inactive		1	RTC	1
t14	Core Well Power and PWROK Inactive to STPCLK# Inactive		1	RTC	1
t15	CPU_STP# and PCI_STP# Float to Clocks Running				2
t16	PWROK Active to CPU_STP# and PCI_STP# Active		1	RTC	1
t17	CPU_STP# and PCI_STP# Active to Clocks Stopped				2
t18	PWROK Active to SLP# Active	0		ns	3
t18a	PWROK Active to STPCLK# Active	0		ns	3
t19	PWROK Active to SLP# Inactive	1	2	RTC	1, 3
t19a	PWROK Active to STPCLK# Inactive	1	2	RTC	1, 3

NOTES:

1. These signals are controlled off an internal RTC clock. 1 RTC unit is approximately 32 µs.
2. There are no specific requirements for these timings related to PIIX4. The system manufacturer should make sure that the clocks on power up meet any other system specifications. As a minimum, the clocks must be available and stable after time t30 shown in Figure 22.
3. These timings depend on the relative timings between RSMRST# and PWROK. If RSMRST# goes inactive 2 RTC periods before PWROK active, then SLP# and STPCLK# will remain inactive. If RSMRST# goes inactive less than 2 RTC periods before PWROK active, then an active pulse will be seen on SLP# and STPCLK#.

11.4.3.6. Power Management Signal Timings (Powered From PIIX4 Main Core Well)

Figure 21 shows the timing relations for Power Management signals powered from the PIIX4 Main Core well. Here the Suspend well power active status signals (RSMRST#) transition before the application of Core well power to PIIX4.

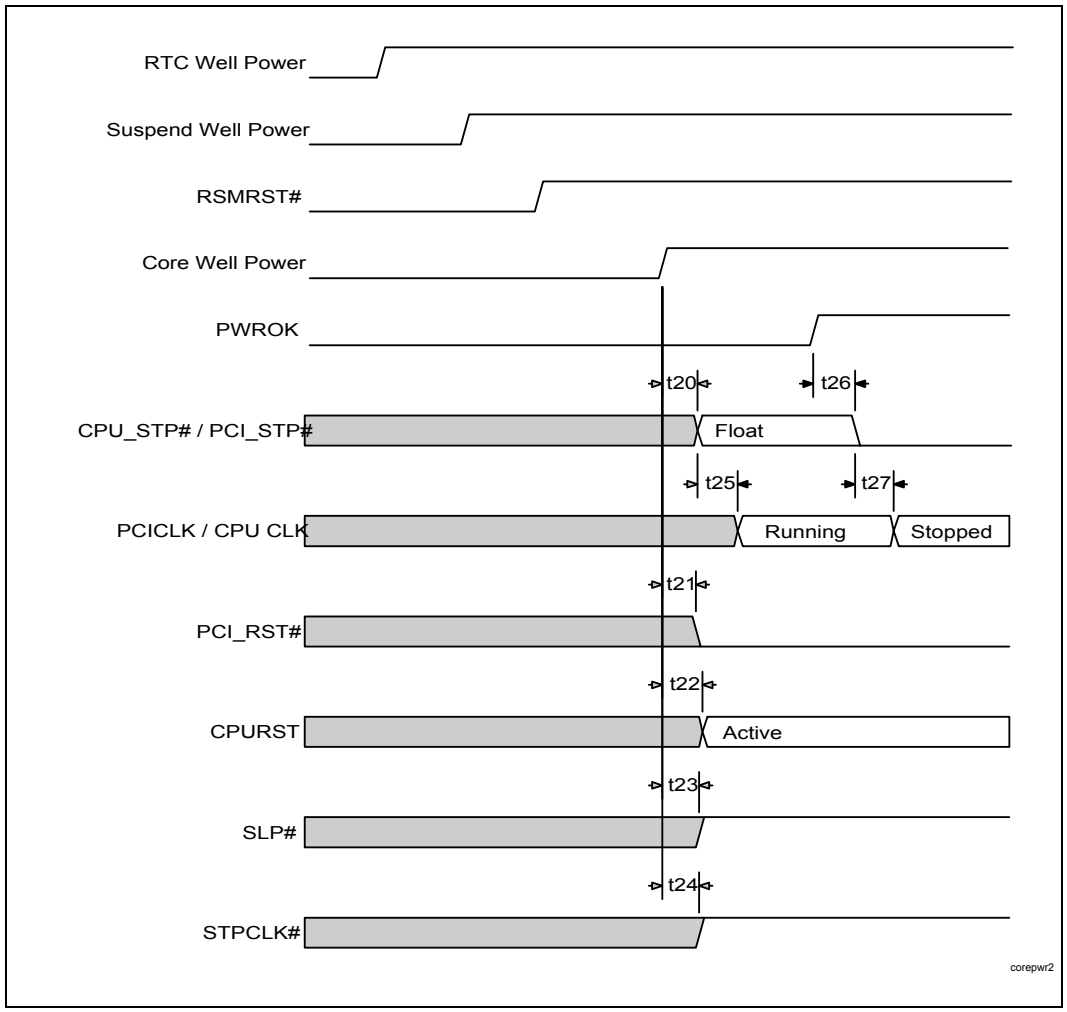


Figure 21. Core Well Power and PWROK Activated Signals (RSMRST# Inactive Before Core Well Power Applied)

Table 41. Core Well Power and PWROK Activated Signals Timings

Sym	Parameter	Min	Max	Unit	Notes
t20	Core Well Power and PWROK Inactive to CPU_STP# and PCI_STP# Float		1	RTC	1
t21	Core Well Power and PWROK Inactive to PCIRST# Active		1	RTC	1
t22	Core Well Power and PWROK Inactive to CPURST Active		1	RTC	1
t23	Core Well Power and PWROK Inactive to SLP# Active		1	RTC	1
t24	Core Well Power and PWROK Inactive to STPCLK# Active		1	RTC	1
t25	CPU_STP# and PCI_STP# Float to Clocks Running				2
t26	PWROK Active to CPU_STP# and PCI_STP# Active		1	RTC	1
t27	CPU_STP# and PCI_STP# Active to Clocks Stopped				2

NOTES:

1. These signals are controlled off an internal RTC clock. 1 RTC unit is approximately 32 µs.
2. There are no specific requirements for these timings related to PIIX4. The system manufacturer should make sure that the clocks on power up meet any other system specifications. As a minimum, the clocks must be available and stable after time t30 shown in Figure 22.

11.4.3.7. Mechanical Off to On Condition Timings

Figure 22 shows the transition from a Mechanical Off condition to the On condition. The signal states begin at the final states shown in Figure 20 and Figure 21.

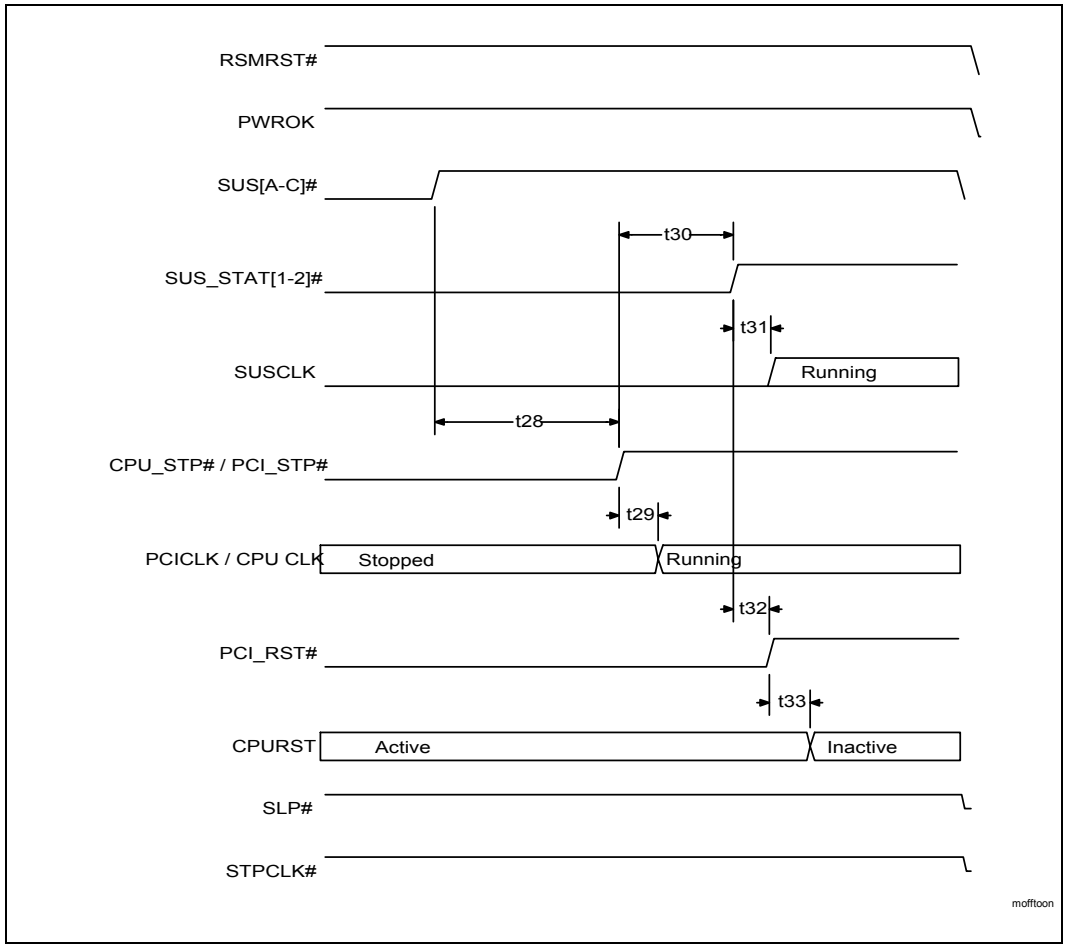


Figure 22. Mechanical Off to On

Table 42. Mechanical Off to On Timings

Sym	Parameter	Min	Max	Unit	Notes
t28	SUS[A:C]# Inactive to CPU_STP# and PCI_STP# Inactive	16		ms	1
t29	CPU_STP# and PCI_STP# Inactive to Clocks Running		2	PCICLK	2
t30	CPU_STP# and PCI_STP# Inactive to SUS_STAT[1:2]# Inactive	1		ms	
t31	SUS_STAT[1:2]# Inactive to SUSCLK Running		1	RTC	3
t32	SUS_STAT[1:2]# Inactive to PCI_RST# Inactive		1	RTC	3
t33	PCI_RST# Inactive to CPURST Inactive		1	RTC	3

NOTES:

1. This transition requires both a minimum of 16-ms wait for clock synthesizer PLL lock and PWROK to be active. If PWROK goes active after 16 ms from SUS[A:C]# inactive, the transition will occur a minimum of 1 RTC period from PWROK active.
2. See Figure 18 and Figure 19 for exact PCICLK requirements for use with PC/PCI DMA and Serial IRQs.
3. These signals are controlled off an internal RTC clock. 1 RTC unit is approximately 32 µs.

11.4.3.8. On State to Power On Suspend State Timing

This section describes the signal transitions from On state to Power On Suspend state.

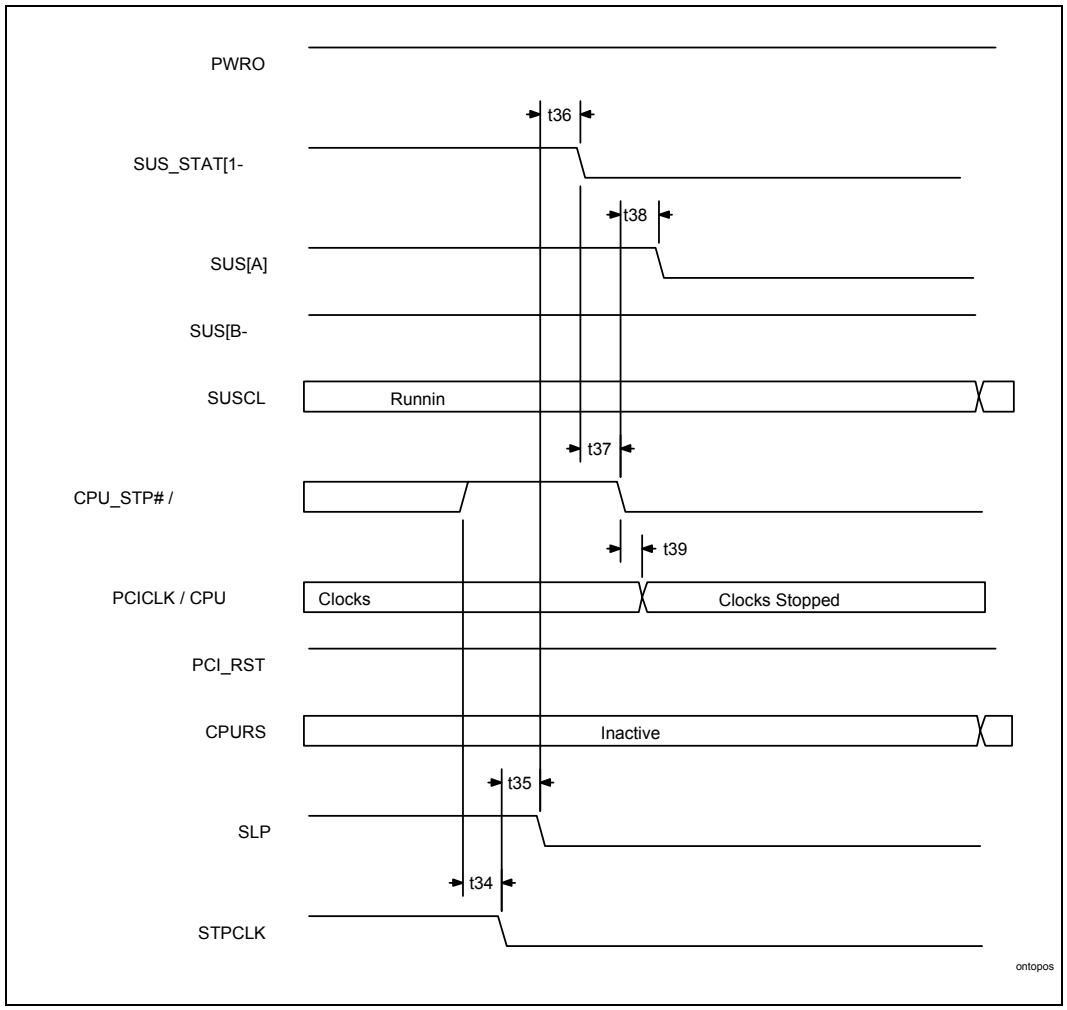


Figure 23. On to POS

Table 43. On to POS Timings

Sym	Parameter	Min	Max	Unit	Notes
t34	CPU_STP# and PCI_STP# Inactive to STPCLK# Active	1		RTC	1, 2
t35	STPCLK# Active to SLP# Active	1		RTC	1, 3
t36	SLP# Active to SUS_STAT[1:2]# Active		1	RTC	1
t37	SUS_STAT[1:2]# Active to CPU_STP# and PCI_STP# Active		1	RTC	1
t38	CPU_STP# and PCI_STP# Active to SUS[A]# Active		1	RTC	1
t39	CPU_STP# and PCI_STP# Active to Clocks Stopped (if applicable)		2	PCICLK	4, 5

NOTES:

1. These signals are controlled off an internal RTC clock. 1 RTC unit is approximately 32 µs.
2. CPU_STP# and PCI_STP# will only be active if system is under clock control.
3. This transition will also wait for the Stop Grant cycle to execute.
4. It is up to the system vendor to determine if CPU_STP# and PCI_STP# signals are used to control system clocks.
5. See Figure 18 and Figure 19 for exact PCICLK requirements for use with PC/PCI DMA and Serial IRQs.

11.4.3.9. Power On Suspend to On Timing (With a Full System Reset)

This section describes the system transition from Power On Suspend to On with a full system reset.

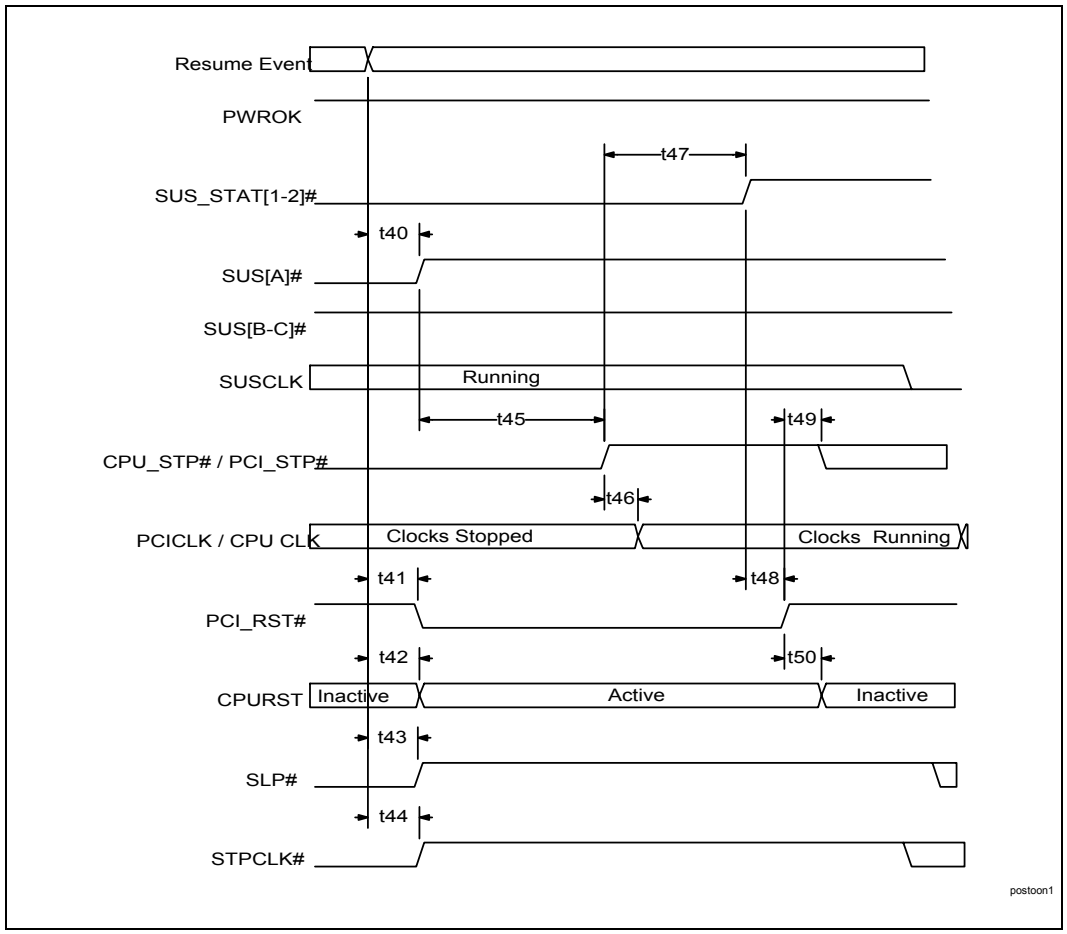


Figure 24. POS to On (w/Processor & PCI Reset)

Table 44. POS to On (w/Processor & PCI Reset) Timings

Sym	Parameter	Min	Max	Unit	Notes
t40	Resume Event to SUS[A]# Inactive	1		RTC	1
t41	Resume Event to PCI_RST# Active	1		RTC	1
t42	Resume Event to CPURST Active	1		RTC	1
t43	Resume Event to SLP# Inactive	1		RTC	1
t44	Resume Event to STPCLK# Inactive	1		RTC	1
t45	SUS[A]# Inactive to PCI_STP# and CPU_STP# Inactive	16		ms	2
t46	PCI_STP# and CPU_STP# Inactive to Clocks Running		2	PCICLK	3
t47	PCI_STP# and CPU_STP# Inactive to SUS_STAT[1:2]# Inactive	1		ms	
t48	SUS_STAT[1:2]# Inactive to PCI_RST# Inactive		1	RTC	1
t49	PCI_RST# Inactive to PCI_STP#; CPU_STP# allowed to change		1	RTC	1
t50	PCI_RST# Inactive to CPURST Inactive		1	RTC	1

NOTES:

1. These signals are controlled off an internal RTC clock. 1 RTC unit is approximately 32 µs.
2. This transition requires both a minimum of 16-ms wait for clock synthesizer PLL lock and PWROK to be active. If PWROK goes active after 16 ms from SUS[A:C]# inactive, the transition will occur a minimum of 1 RTC period from PWROK active. PWROK remains active throughout standard POS system usage.
3. See Figure 18 and Figure 19 for exact PCICLK requirements for use with PC/PCI DMA and Serial IRQs.

11.4.3.10. System Transition From Power On Suspend to On (With Only Processor Reset)

This section describes the system transition from Power On Suspend to On with only a processor reset.

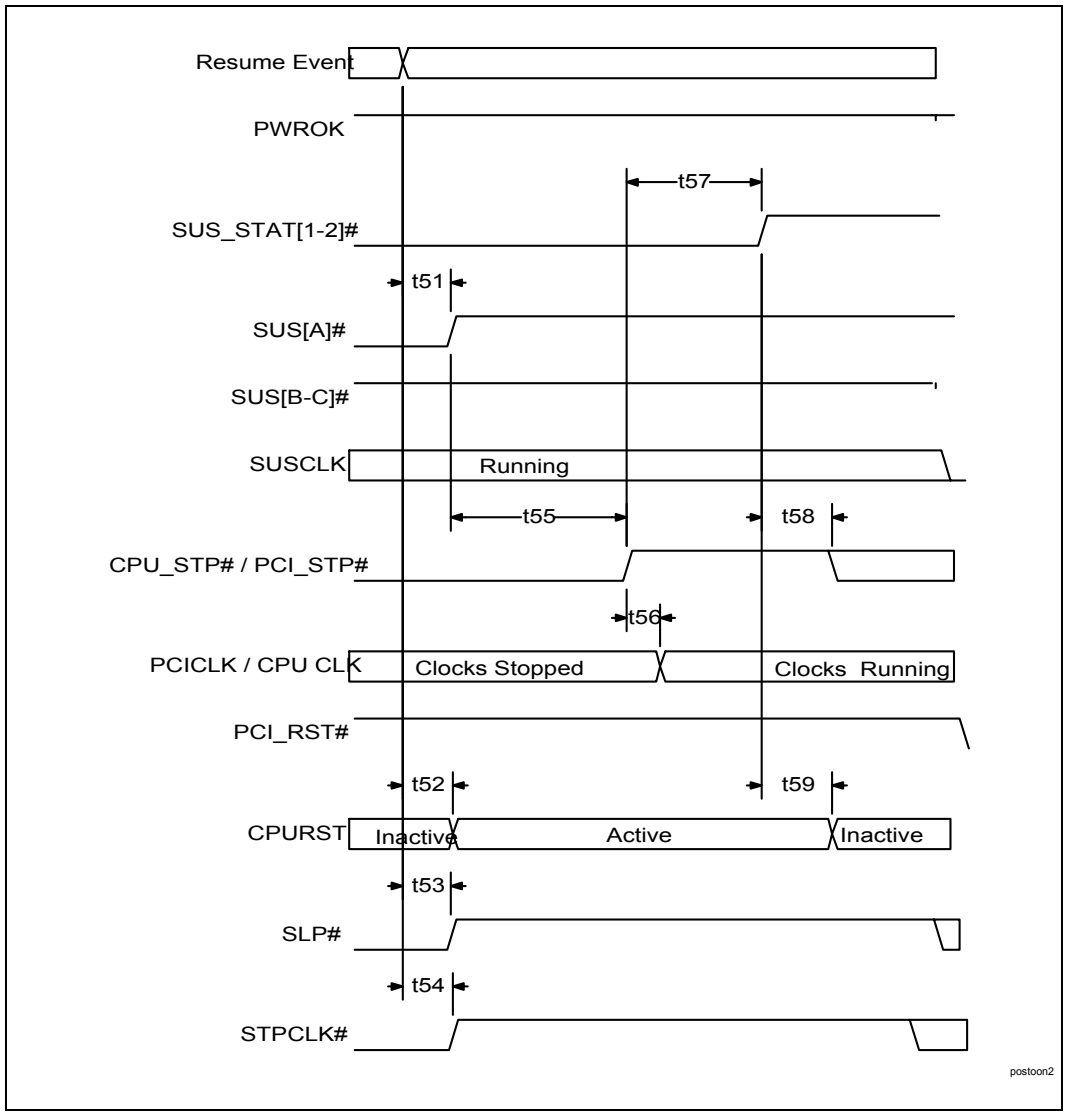


Figure 25. POS to On (w/Processor Reset)

Table 45. POS to On (w/Processor Reset) Timings

Sym	Parameter	Min	Max	Unit	Notes
t51	Resume Event to SUS[A]# Inactive	1		RTC	1
t52	Resume Event to CPURST Active	1		RTC	1
t53	Resume Event to SLP# Inactive	1		RTC	1
t54	Resume Event to STPCLK# Inactive	1		RTC	1
t55	SUS[A]# Inactive to PCI_STP# and CPU_STP# Inactive	16		ms	2
t56	PCI_STP# and CPU_STP# Inactive to Clocks Running		2	PCICLK	3
t57	PCI_STP# and CPU_STP# Inactive to SUS_STAT[1:2]# Inactive	1		ms	
t58	SUS_STAT[1:2]# Inactive to PCI_STP# and CPU_STP# allowed to change		2	RTC	1
t59	SUS_STAT[1:2]# Inactive to CPURST Inactive		2	RTC	1

NOTES:

1. These signals are controlled off an internal RTC clock. 1 RTC unit is approximately 32 µs.
2. This transition requires both a minimum of 16-ms wait for clock synthesizer PLL lock and PWROK to be active. If PWROK goes active after 16 ms from SUS[A:C]# inactive, the transition will occur a minimum of 1 RTC period from PWROK active. PWROK remains active throughout standard POS system usage.
3. See Figure 18 and Figure 19 for exact PCICLK requirements for use with PC/PCI DMA and Serial IRQs.

11.4.3.11. Power On Suspend to On Timing (With No Resets)

This section describes the system transition from Power On Suspend to On with no resets performed.

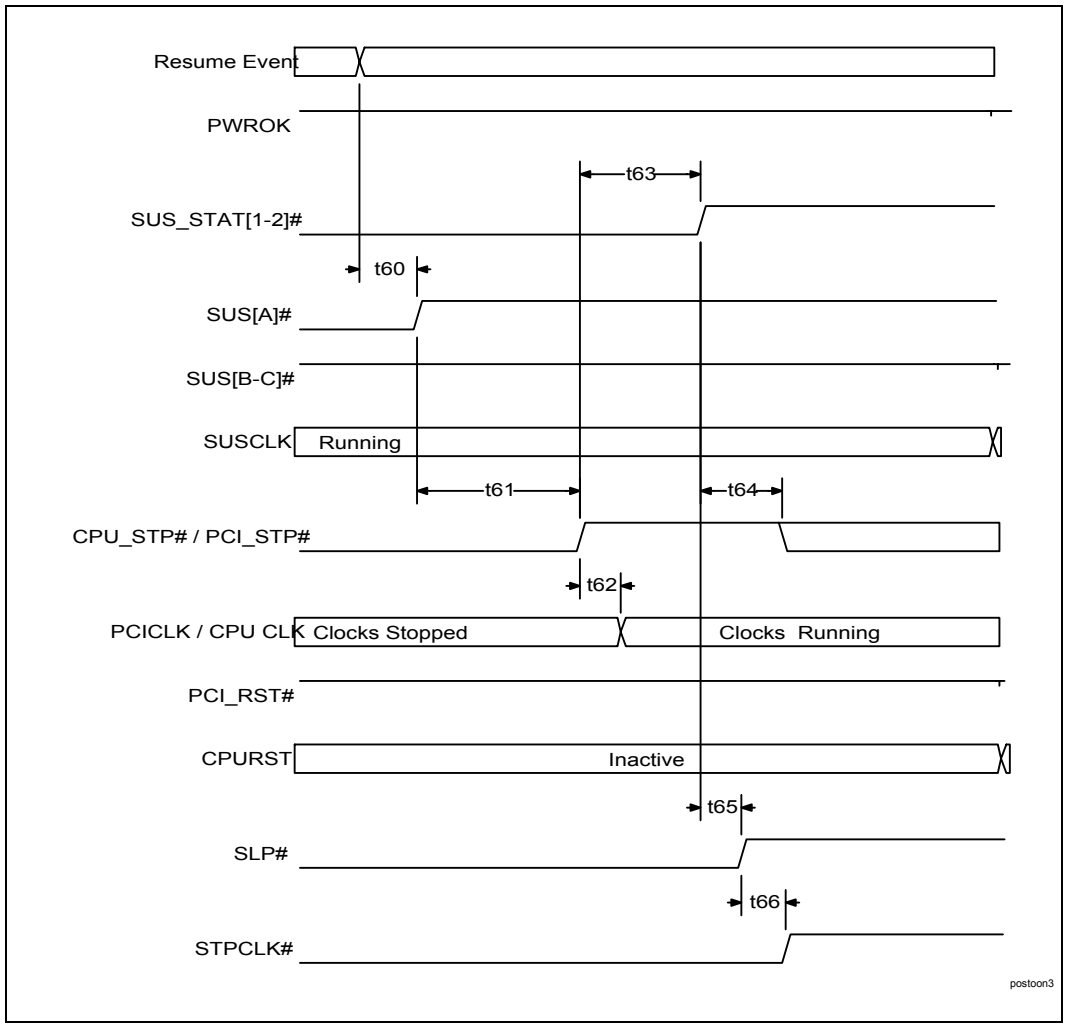


Figure 26. POS to On (No Reset)

Table 46. POS to On (w/Processor Reset) Timings

Sym	Parameter	Min	Max	Unit	Notes
t60	Resume Event to SUS[A]# Inactive	1		RTC	1
t61	SUS[A]# Inactive to PCI_STP# and CPU_STP# Inactive	16		ms	2
t62	PCI_STP# and CPU_STP# Inactive to Clocks Running		2	PCICLK	3
t63	PCI_STP# and CPU_STP# Inactive to SUS_STAT[1:2]# Inactive	1		ms	
t64	SUS_STAT[1:2]# Inactive to PCI_STP# and CPU_STP# allowed to change		2	RTC	1
t65	SUS_STAT[1:2]# Inactive to SLP# Inactive		1	RTC	1
t66	SLP# Inactive to STPCLK# Inactive		1	RTC	1

NOTES:

1. These signals are controlled off the internal RTC clock. 1 RTC is approximately 32 µs.
2. This transition requires both a minimum of 16-ms wait for clock synthesizer PLL lock and PWROK to be active. If PWROK goes active after 16 ms from SUS[A:C]# inactive, the transition will occur a minimum of 1 RTC period from PWROK active. PWROK remains active throughout standard POS system usage.
3. See Figure 18 and Figure 19 for exact PCICLK requirements for use with PC/PCI DMA and Serial IRQs.

11.4.3.12. On State to Suspend to RAM State Timing

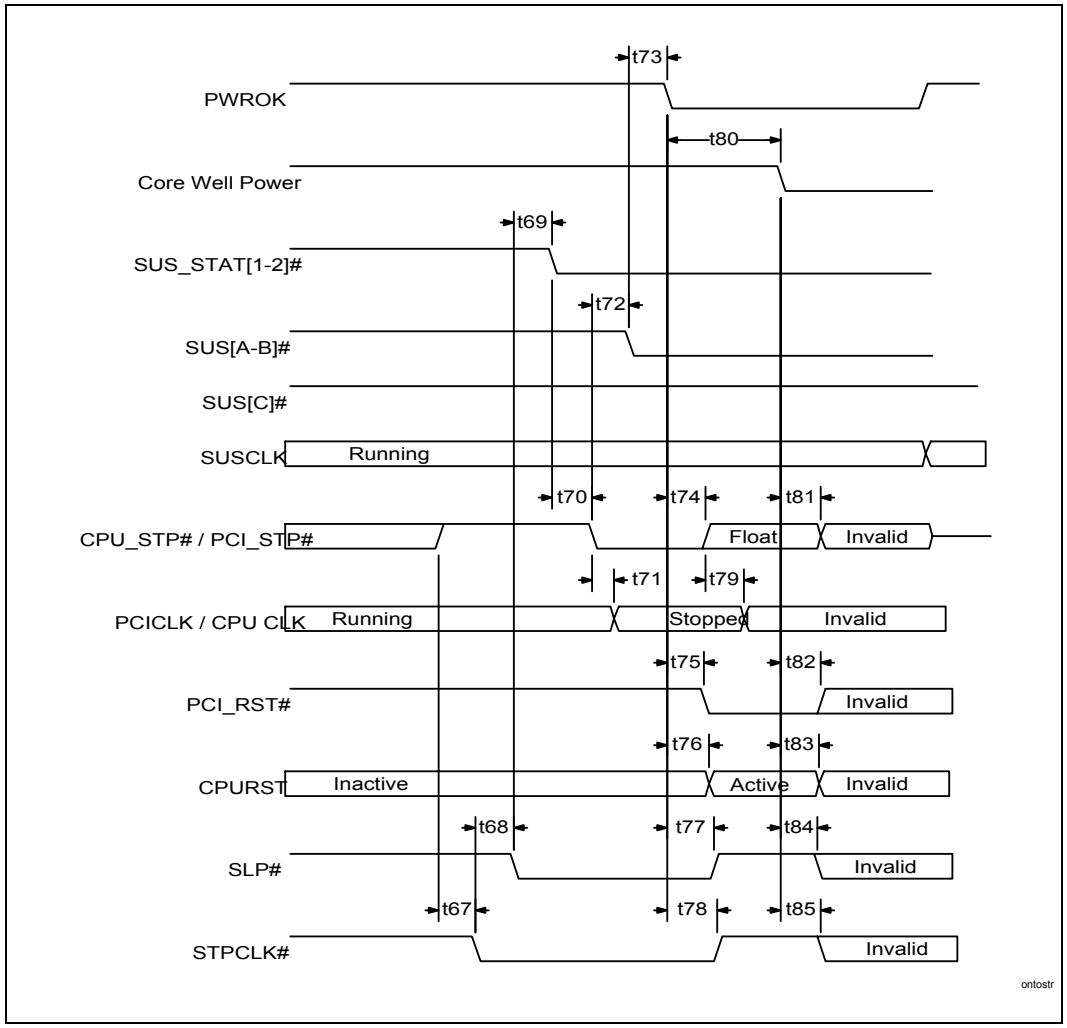


Figure 27. On to STR

Table 47. On to STR Timings

Sym	Parameter	Min	Max	Unit	Notes
t67	CPU_STP# and PCI_STP# Inactive to STPCLK# Active	1		RTC	1, 2
t68	STPCLK# Active to SLP# Active	1		RTC	1, 3
t69	SLP# Active to SUS_STAT[1:2]# Active		1	RTC	1
t70	SUS_STAT[1:2]# Active to CPU_STP# and PCI_STP# Active		1	RTC	1
t71	CPU_STP# and PCI_STP# Active to Clocks Stopped		2	PCICLK	4, 5
t72	CPU_STP# and PCI_STP# Inactive to SUS[A:B]# Active		1	RTC	1
t73	SUS[A:B]# Active to PWROK Inactive	0		ns	6
t74	PWROK Inactive to CPU_STP# and PCI_STP# Float		1	RTC	1
t75	PWROK Inactive to PCI_RST# Active		1	RTC	1
t76	PWROK Inactive to CPURST Active		1	RTC	1
t77	PWROK Inactive to SLP# Inactive		1	RTC	1
t78	PWROK Inactive to STPCLK# Inactive		1	RTC	1
t79	CPU_STP# and PCI_STP# Float to Clocks Invalid	0		ns	7
t80	PWROK Inactive to Core Well Power Removed	0		ns	
t81	Core Well Power Removed to PCI_STP# and CPU_STP# Invalid	0		ns	
t82	Core Well Power Removed to PCIRST# Invalid	0		ns	
t83	Core Well Power Removed to CPURST Invalid	0		ns	
t84	Core Well Power Removed to SLP# Invalid	0		ns	
t85	Core Well Power Removed to STPCLK# Invalid	0		ns	

NOTES:

1. These signals are controlled off the internal RTC clock. 1 RTC is approximately 32 µs.
2. CPU_STP# and PCI_STP# will only be active if system is under clock control.
3. This transition will also wait for the Stop Grant cycle to execute.
4. It is up to the system vendor to determine if CPU_STP# and PCI_STP# signals are used to control system clocks.
5. See Figure 18 and Figure 19 for exact PCICLK requirements for use with PC/PCI DMA and Serial IRQs.
6. It is up to the system vendor to determine if SUS[A:B]# signals are used to control system power planes. If power remains applied to system board and PWROK stays active during STR, the PIIX4 signals remain in the states shown after t73.
7. Clocks may or may not be running depending on condition of Power Supply voltages.

11.4.3.13. Suspend-To-RAM to On Timing (With Full System Reset)

This section describes the system transition from Suspend To RAM to On with a full system reset.

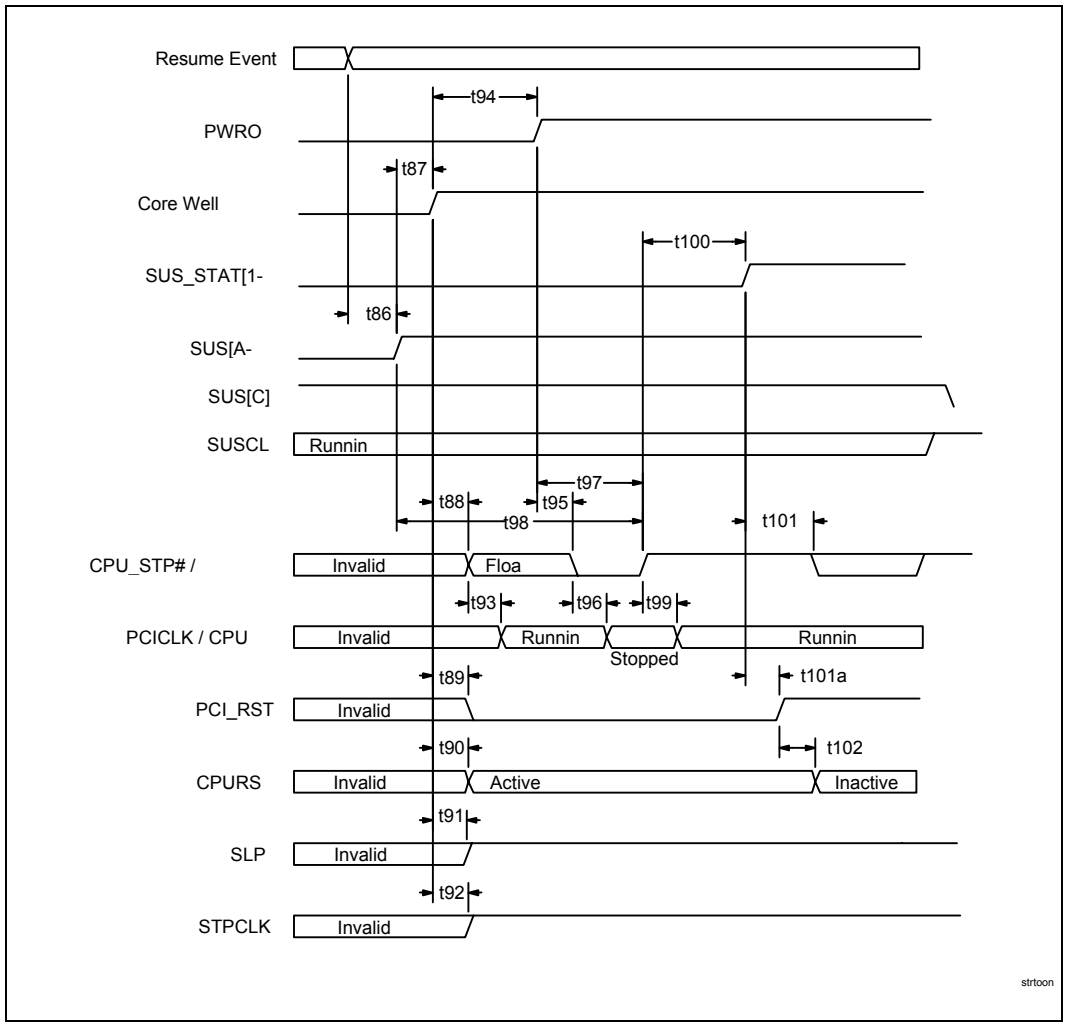


Figure 28. STR to On

Table 48. STR to On Timings

Sym	Parameter	Min	Max	Unit	Notes
t86	Resume Event to SUS[A:B]# Inactive	1		RTC	1
t87	SUS[A:B]# Inactive to Core Well Power Applied	0		ns	
t88	Core Well Power Applied to PCI_STP# and CPU_STP# Float	0		ns	
t89	Core Well Power Applied to PCI_RST# Active	0		ns	
t90	Core Well Power Applied to CPURST Active	0		ns	
t91	Core Well Power Applied to SLP# Inactive	0		ns	
t92	Core Well Power Applied to STPCLK# Inactive	0		ns	
t93	PCI_STP# and CPU_STP# Float to Clocks Running				2
t94	Core Well Power Applied to PWROK Active	1		ms	
t95	PWROK Active to CPU_STP# and PCI_STP# Active	0		ns	
t96	PCI_STP# and CPU_STP# Active to Clocks Stopped		2	PCICLK	3
t97	PWROK Active to CPU_STP# and PCI_STP# Inactive	1		RTC	1
t98	SUS[A:B]# Inactive to CPU_STP# and PCI_STP# Inactive	16		ms	
t99	CPU_STP# and PCI_STP# Inactive to Clocks Running		2	PCICLK	3
t100	CPU_STP# and PCI_STP# Inactive to SUS_STAT[1:2]# Inactive	1		ms	
t101	SUS_STAT[1:2]# Inactive to CPU_STP# and PCI_STP# allowed to change	2		RTC	1
t101a	SUS_STAT[1:2]# Inactive to PCI_RST# Inactive	1		RTC	1
t102	PCI_RST# Inactive to CPURST Inactive	1		RTC	1

NOTES:

1. These signals are controlled off the internal RTC clock. 1 RTC is approximately 32 µs.
2. There are no specific requirements for these timings related to PIIX4. The system manufacturer should make sure that the clocks on power up meet any other system specifications. As a minimum, the clocks must be available and stable after time t99.
3. See Figure 18 and Figure 19 for exact PCICLK requirements for use with PC/PCI DMA and Serial IRQs.

11.4.3.14. On State to Suspend to Disk/Soft Off State Timings

This section describes the signal transitions from On state to Suspend to Disk/Soft Off state.

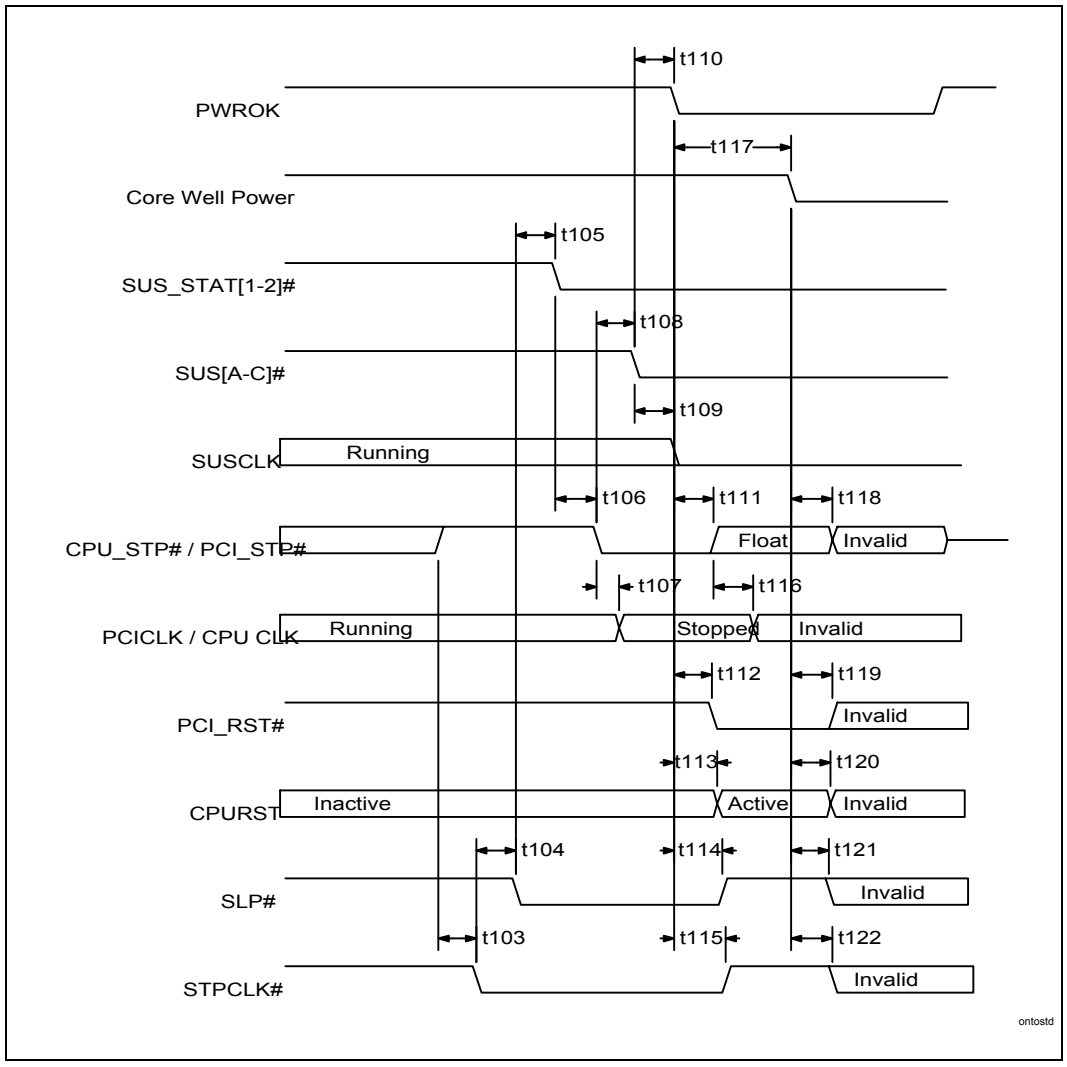


Figure 29. On to STD/SOff

Table 49. On to STD/SOff Timings

Sym	Parameter	Min	Max	Unit	Notes
t103	CPU_STP# and PCI_STP# Inactive to STPCLK# Active	1		RTC	1, 2
t104	STPCLK# Active to SLP# Active	1		RTC	1, 3
t105	SLP# Active to SUS_STAT[1:2]# Active		1	RTC	1
t106	SUS_STAT[1:2]# Active to CPU_STP# and PCI_STP# Active		1	RTC	1
t107	CPU_STP# and PCI_STP# Inactive to Clocks Stopped		2	PCICLK	1, 4, 5
t108	CPU_STP# and PCI_STP# Inactive to SUS[A:C]# Active		1	RTC	1
t109	SUS[A:C]# Active to SUSCLK Low		1	RTC	1
t110	SUS[A:C]# Active to PWROK Inactive	0		ns	6
t111	PWROK Inactive to CPU_STP# and PCI_STP# Float		1	RTC	1
t112	PWROK Inactive to PCI_RST# Active		1	RTC	1
t113	PWROK Inactive to CPURST Active		1	RTC	1
t114	PWROK Inactive to SLP# Inactive		1	RTC	1
t115	PWROK Inactive to STPCLK# Inactive		1	RTC	1
t116	CPU_STP# and PCI_STP# Float to Clocks Invalid	0		ns	1
t117	PWROK Inactive to Core Well Power Removed	0		ns	
t118	Core Well Power Removed to PCI_STP# and CPU_STP# Invalid	0		ns	
t119	Core Well Power Removed to PCIRST# Invalid	0		ns	
t120	Core Well Power Removed to CPURST Invalid	0		ns	
t121	Core Well Power Removed to SLP# Invalid	0		ns	
t122	Core Well Power Removed to STPCLK# Invalid	0		ns	

NOTES:

1. These signals are controlled off the internal RTC clock. 1 RTC is approximately 32 µs.
2. CPU_STP# and PCI_STP# will only be active if system is under clock control.
3. This transition will also wait for the Stop Grant cycle to execute.
4. It is up to the system vendor to determine if CPU_STP# and PCI_STP# signals are used to control system clocks.
5. See Figure 18 and Figure 19 for exact PCICLK requirements for use with PC/PCI DMA and Serial IRQs.
6. It is up to the system vendor to determine if SUS[A:C]# signals are used to control system power planes. If power remains applied to system board and PWROK stays active during STD, the PIIX4 signals remain in the states shown after t110.

11.4.3.15. Suspend-To-Disk to On (With Full System Reset)

This section describes the system transition from Suspend To Disk to On with a full system reset.

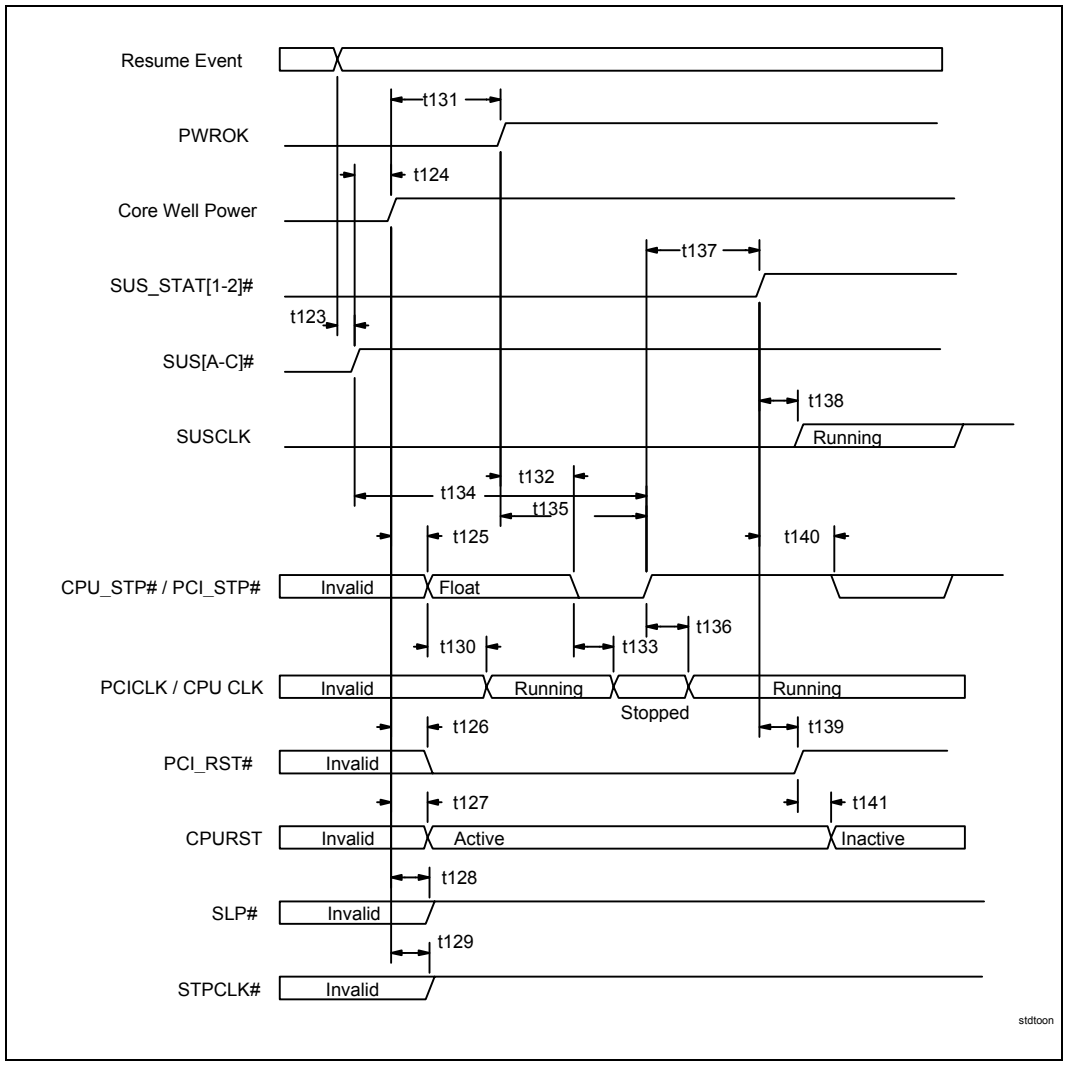


Figure 30. STD/SOff to On

Table 50. STD/SOff to On Timings

Sym	Parameter	Min	Max	Unit	Notes
t123	Resume Event to SUS[A:C]# Inactive	1		RTC	1
t124	SUS[A:C]# Inactive to Core Well Power Applied	0		ns	
t125	Core Well Power Applied to PCI_STP# and CPU_STP# Float	0		ns	
t126	Core Well Power Applied to PCI_RST# Active	0		ns	
t127	Core Well Power Applied to CPURST Active	0		ns	
t128	Core Well Power Applied to SLP# Inactive	0		ns	
t129	Core Well Power Applied to STPCLK# Inactive	0		ns	
t130	PCI_STP# and CPU_STP# Float to Clocks Running				2
t131	Core Well Power Applied to PWROK Active	1		ms	
t132	PWROK Active to CPU_STP# and PCI_STP# Active	0		ns	
t133	PCI_STP# and CPU_STP# Active to Clocks Stopped		2	PCICLK	3
t134	SUS[A:C]# Inactive to CPU_STP# and PCI_STP# Inactive	16		ms	
t135	PWROK Active to CPU_STP# and PCI_STP# Inactive	1		RTC	1
t136	PCI_STP# and CPU_STP# Active to Clocks Running	1	2	PCICLK	3
t137	CPU_STP# and PCI_STP# Inactive to SUS_STAT[1:2]# Inactive	1		ms	
t138	SUS_STAT[1:2]# Inactive to SUSCLK Running	1		RTC	1
t139	SUS_STAT[1:2]# Inactive to PCI_RST# Inactive	1		RTC	1
t140	SUS_STAT[1:2]# Inactive to CPU_STP# and PCI_STP# allowed to change	2		RTC	1
t141	PCI_RST# Inactive to CPURST Inactive	1		RTC	1

NOTES:

1. These signals are controlled off the internal RTC clock. 1 RTC is approximately 32 µs.
2. There are no specific requirements for these timings related to PIIX4. The system manufacturer should make sure that the clocks on power up meet any other system specifications. As a minimum, the clocks must be available and stable after time t136.
3. See Figure 18 and Figure 19 for exact PCICLK requirements for use with PC/PCI DMA and Serial IRQs.

11.4.4. SHADOW REGISTERS

PIIX4 includes a shadow mechanism for storing the data written to the standard AT write only registers. In the transition to Suspend mode, the contents of these registers can be read and saved so the system state can be restored when resumed.

Once placed in the “Alternate Access” mode, various registers that would otherwise be inaccessible can be read and written. This allows a system to restore the configuration that was present in PIIX4 before going into a suspend state.

To enable the Alt Access mode, set the following bit to 1: PCI Function 0, Register B0h, Bit 5.

NOTE

No provisions are made for stopping events from occurring while the BIOS is reading or restoring register values. The BIOS should exercise great care in using this feature. For example, when reading the status of the DMA controller, all the DMA channels should be temporarily masked.

The following tables show the changes to the read and write accesses associated with the various modules. When the Alt Access mode is enabled, some read and write cycles will cause alternative registers to be accessed.

See the description for Miscellaneous Support Register, PCI Function 2 (offset FFh) for details on saving and restoring the RTC Index value.

NOTE

It is assumed that no other accesses to the module will be permitted once in ALT Access Mode.

Table 51. DMA Controller Registers in Alternate Access Mode

I/O Location	R/W	Standard Mode Usage	ALT Access Mode
0000h	W	Base Address for CH0	Current Address for CH0
0000h	R	Current Address for CH0	Base Address for CH0
0001h	W	Base Byte Count for CH0	Current Byte Count for CH0
0001h	R	Current Byte Count for CH0	Base Byte Count for CH0
0002h	W	Base Address for CH1	Current Address for CH1
0002h	R	Current Address for CH1	Base Address for CH1
0003h	W	Base Byte Count for CH1	Current Byte Count for CH1
0003h	R	Current Byte Count for CH1	Base Byte Count for CH1
0004h	W	Base Address for CH2	Current Address for CH2
0004h	R	Current Address for CH2	Base Address for CH2
0005h	W	Base Byte Count for CH2	Current Byte Count for CH2
0005h	R	Current Byte Count for CH2	Base Byte Count for CH2
0006h	W	Base Address for CH3	Current Address for CH3
0006h	R	Current Address for CH3	Base Address for CH3

Table 51. DMA Controller Registers in Alternate Access Mode

I/O Location	R/W	Standard Mode Usage	ALT Access Mode
0007h	W	Base Byte Count for CH3	Current Byte Count for CH3
0007h	R	Current Byte Count for CH3	Base Byte Count for CH3
0008h	W	Command Register (CH[0:3])	Status Register (CH[0:3])
0008h	R	Status Register (CH[0:3])	1st Read: Command Register (CH[0:3]) 2nd Read: Request Register (CH[0:3]) 3rd Read: Mode Register (CH0) 4th Read: Mode Register (CH1) 5th Read: Mode Register (CH2) 6th Read: Mode Register (CH3)
0009h	W	Request Register (CH[0:3])	Reserved
0009h	R	Reserved	Reserved
000Ah	W	Write Single Mask (CH[0:3])	Reserved
000Ah	R	Reserved	Reserved
000Bh	W	Mode Register (CH[0:3])	Reserved
000Bh	R	Reserved	Reserved
000Ch	W	Clear Byte Pointer	Clear Byte Pointer
000Ch	R	Reserved	Reserved
000Dh	W	Master Clear	Master Clear
000Dh	R	Reserved	Reserved
000Eh	W	Clear Masks	Clear Masks
000Eh	R	Reserved	Reserved
000Fh	W	Write All Mask (0–3)	Write All Mask (0–3)
000Fh	R	Reserved	Read All Masks (0–3)
00C0h	W	Base Address for CH4	Current Address for CH4
00C0h	R	Current Address for CH4	Base Address for CH4
00C2h	W	Base Word Count for CH4	Current Word Count for CH4
00C2h	R	Current Word Count for CH4	Base Word Count for CH4
00C4h	W	Base Address for CH5	Current Address for CH5
00C4h	R	Current Address for CH5	Base Address for CH5
00C6h	W	Base Word Count for CH5	Current Word Count for CH5
00C6h	R	Current Word Count for CH5	Base Word Count for CH5
00C8h	W	Base Address for CH6	Current Address for CH6

Table 51. DMA Controller Registers in Alternate Access Mode

I/O Location	R/W	Standard Mode Usage	ALT Access Mode
00C8h	R	Current Address for CH6	Base Address for CH6
00CAh	W	Base Word Count for CH6	Current Word Count for CH6
00CAh	R	Current Word Count for CH6	Base Word Count for CH6
00CCh	W	Base Address for CH7	Current Address for CH7
00CCh	R	Current Address for CH7	Base Address for CH7
00CEh	W	Base Word Count for CH7	Current Word Count for CH7
00CEh	R	Current Word Count for CH7	Base Word Count for CH7
00D0h	W	Command Register (CH[4:7])	Status Register (CH[4:7])
00D0h	R	Status Register (CH[4:7])	1st Read: Command Register (CH[4:7]) 2nd Read: Request Register (CH[4:7]) 3rd Read: Mode Register (Ch 4) 4th Read: Mode Register (Ch 5) 5th Read: Mode Register (Ch 6) 6th Read: Mode Register (Ch 7)
00D2h	W	Request Register (CH[4:7])	Reserved
00D2h	R	Reserved	Reserved
00D4h	W	Write Single Mask (CH[4:7])	Reserved
00D4h	R	Reserved	Reserved
00D6h	W	Mode Register (CH[4:7])	Reserved
00D6h	R	Reserved	Reserved
00D8h	W	Clear Byte Pointer	Clear Byte Pointer
00D8h	R	Reserved	Reserved
00DAh	W	Master Clear	Master Clear
00DAh	R	Reserved	Reserved
00DCh	W	Clear Masks	Clear Masks
00DCh	R	Reserved	Reserved
00DEh	W	Write All Mask (0–3)	Write All Mask (4–7)
00DEh	R	Reserved	Read All Masks (4–7)

NOTES:

1. This allows reading and restoring all of the initial base address and byte/word counts. Also makes it possible to read command, mode, and mask registers, as well as restore status, mode, and mask register.

Table 52. NMI Enable Bit Changes in Alternate Access Mode

I/O Location	R/W	Standard Mode Usage	ALT Access Mode
0070h (bit 7 only)	R	Write Only for NMI Enable Bit	Read Bit Enable Bit

Table 53. Programmable Interval Timer Changes in Alternate Access Mode

I/O Location	R/W	Standard Mode Usage	ALT Access Mode
0040h	R	Status Byte Counter 0	1st Read: Status Byte Counter 0 2nd Read: CR_L for Counter 0 3rd Read: CR_M for Counter 0 4th Read: CR_L for Counter 1 5th Read: CR_M for Counter 1 6th Read: CR_L for Counter 2 7th Read: CR_M for Counter 2
0041h	R	Status Byte Counter 1	Status Byte Counter 1
0042h	R	Status Byte Counter 2	Status Byte Counter 2

The BIOS must perform seven consecutive reads from port 40h in alternate access mode. If BIOS deviates from this it may get inaccurate data. It also allows BIOS to set the alt access bit and still read the status of all the counters. Setting the Alt Access Mode automatically clears the high/low flip flop. When the ALT Access Mode is entered the timers do not stop counting, hence the current values will change from the time the initial value is read.

Table 54. Programmable Interrupt Controller

I/O Location	R/W	Standard Mode Usage	ALT Access Mode
0020h	R	Interrupt Request Register for Cntrl 1	1st Read: ICW1 for Cntrl 1 2nd Read: ICW2 for Cntrl 1 3rd Read: ICW3 for Cntrl 1 4th Read: ICW4 for Cntrl 1 5th Read: OCW1 for Cntrl 1 6th Read: OCW2 for Cntrl 1 7th Read: OCW3 for Cntrl 1 8th Read: ICW1 for Cntrl 2 9th Read: ICW2 for Cntrl 2 10th Read: ICW3 for Cntrl 2 11th Read: ICW4 for Cntrl 2 12th Read: OCW1 for Cntrl 2 13th Read: OCW2 for Cntrl 2 14th Read: OCW3 for Cntrl 2
0021h	R	In-Service Register for Cntrl 1	In-Service Register for Cntrl 1
00A0h	R	Interrupt Request Register for Cntrl 2	Interrupt Request Register for Cntrl 2
00A1h	R	In-Service Register for Cntrl 2	In-Service Register for Cntrl 2

11.5. System Management

PIIX4's system management capabilities include providing means to communicate system activities to system management software and to communicate with other devices on the system board. The first item is performed through the System Management Interrupt (SMI) function while the second is done with a System Management Bus host and slave controller.

11.5.1. SMI OPERATION

System Management Interrupts are generated to the processor through the assertion of the SMI# signal. Various system events, described below, will cause the SMI# signal to be asserted if enabled.

Figure 31 shows the operation of SMI generation logic. SMI generation is enabled by setting the [SMI_EN] bit and controlled by the End of SMI [EOS] bit. The [EOS] bit is first set to enable the generation of the first SMI. When an enabled SMI# generation event occurs, the EOS bit is reset to 0. When this bit is cleared the SMI# signal to the processor is asserted. The processor then enters System Management Mode (SMM) and the SMI handler services all requesting SMIs. If an SMI event occurs while PIIX4 has this bit cleared, no additional SMIs to the processor are generated; however, the appropriate status bits are set. At the end of the SMI handler, software sets this bit. When set, PIIX4 drives the SMI# signal inactive for a minimum of 1 PCI clock. The combination of this bit being set, and another SMI request being active (one of the SMI status bits is set) causes PIIX4 to reset [EOS] bit again and re-assert the SMI signal to the processor.

NOTE

EOS bit will not get set until all SMI status bits are cleared. Therefore, before exiting, the SMI handler needs to verify that the bit is actually set. Otherwise, there could be another pending SMI that will prevent the EOS bit from being set. In this case, the SMI handler will need to clear that SMI status bit and set the EOS bit again.

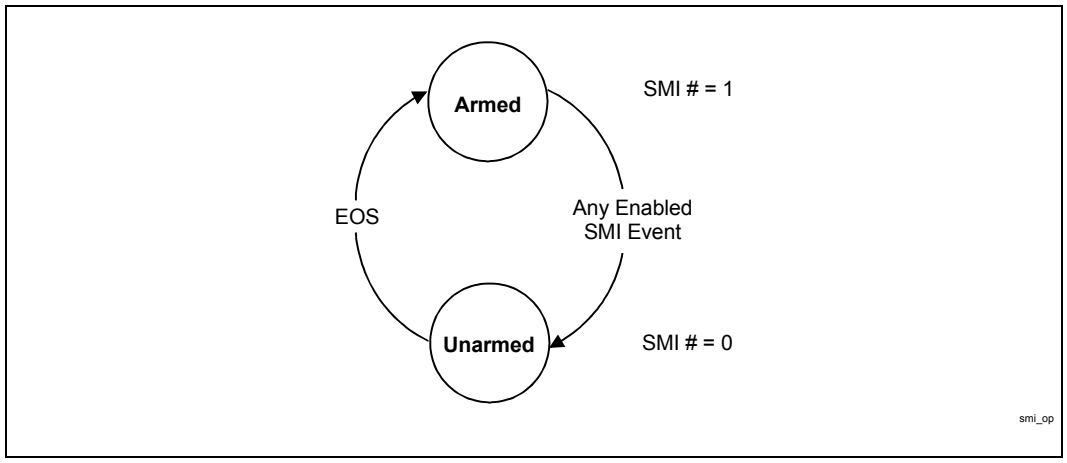


Figure 31. SMI Operation

11.5.2. SMI# GENERATION EVENTS

Some of the events can also generate the ACPI compatible System Control Interrupt (SCI) or suspend state resume events. The SMI# or SCI is selectable with the [SCI_EN] bit. When set to 1, these events generate an SCI, if enabled. When reset, these events generate an SMI#, if enabled. See the “APIC Support” section for additional details on ACPI and SCI. See the “Suspend/Resume and Power Plane Control” section for additional details on system suspend and resume.

When an event generates an SMI# event, it sets a status bit. Status bits from various sources are combined together (“ORed”) to create hierarchical status bits. The hierarchical status bits cannot be reset by software. Their respective “children” status bits must all be cleared in order for them to clear.

PWRBTN# Asserted: [PWRBTN_EN] [PWRBTN_STS]
[PWRBTNOR_EN] [PWRBTNOR_STS]

— The PWRBTN# input signal can be used to generate an SMI# upon its assertion. It contains a 16-ms debounce circuit to filter out mechanical switch bounce. When asserted, it will set the [PWRBTN_STS] bit after the 16-ms debounce. This will cause generation of an SMI# if enabled. If the PWRBTN# signal is held active for greater than 4 seconds and Power Button Override feature is enabled, the [PWRBTN_STS] bit is cleared, the [PWRBTNOR_STS] bit is set, and PIIX4 will automatically transition the system into the Soft Off Suspend State (see the “Suspend/Resume and Power Plane Control” section for more details on Soft Off Suspend). This signal can also be used to generate an SCI or a suspend state resume event.

LID Asserted: [LID_EN] [LID_STS]

— **Polarity Select:** [LID_POL]

— The LID signal, when asserted, sets the [LID_STS] bit after a 16-ms debounce, and when enabled, generates an SMI#. The assertion polarity can be controlled to allow system code to detect when LID signal transitions from low to high or high to low. This signal can also be used to generate an SCI or a suspend state resume event.

GPI1 Asserted: [GPI_EN] [GPI_STS]

— The GPI1 signal, when asserted LOW, will set the [GPI_STS] bit, and when enabled will generate an SMI#. This signal can also be used to generate an SCI or a suspend state resume event.

EXTSMI# Asserted: [EXTSMI_EN] [EXTSMI_STS]

— The EXTSMI# signal, when asserted, will set the [EXTSMI_STS] bit, and when enabled will generate an SMI#. This signal can also be used to generate a suspend state resume event.

SMBus Events: [ALERT_EN] [ALERT_STS]
[SLV_EN] [SLV_STS]
[SHDW1_EN] [SHDW1_STS]
[SHDW2_EN] [SHDW2_STS]

— The System Management Bus (SMBus) controller has various means to generate an SMI#. These can also be used to generate a suspend state resume events. See the “SMBus Functional Description” section on page 265 for additional information.

Global Standby Timer Expiration: [GSTBY_EN] [GSTBY_STS]

— The Global Standby Timer will set the [GSTBY_STS] bit upon expiration, and if enabled will generate an SMI#. It can also be used to generate a suspend state resume events. See below for more information on Global Standby Timer operation.

PCI Bus Master Requests: [BM_TRP_EN] [BM_STS]

— Assertion of PCIREQ[0:3] or PHOLD#, signifying PCI Master activity will generate an SMI# if enabled. This can also cause idle, burst, or global standby timer reloads as part of Device 8 Monitor logic.

APMC Control Register Write: [APMC_EN] [APM_STS]

— Writes to the APM Control Register (APMC, IO port B2h) generate an SMI#, if enabled.

USB Legacy Keyboard/Mouse: [LEGACY_USB_EN] [LEGACY_USB_STS]

— The USB Legacy Keyboard logic uses SMI# generation as part of its operation. See the “USB Host Controller Functional Description” section and the “Universal Host Controller Interface Design Guide” document for additional information concerning USB Legacy Keyboard. The [LEGACY_USB_EN] bit must be set active in order for USB Legacy Keyboard to function.

Software Timer SMI: [IDL_EN_DEV3] [IDL_STS_DEV3]
[IDL_RLD_EN_DEV3]

— The Idle Timer for Device 3 Monitoring can be used as a Software SMI Timer. If the Idle timer reload events are disabled (via [IDL_RLD_EN_DEV3] bit), the timer will count down without reload and its expiration will generate an SMI#. See the “Peripheral Device Management” section for more information on idle timer operation.

Device Monitor Trap: [TRP_EN_DEVx] [TRP_STS_DEVx]
x=0–13 [DEV_STS]

— The IO Traps for Device Monitoring subsystem generate an SMI# when the programmed trap event occurs. The [DEV_STS] bit is logical “OR” of [TRP_STS_DEVx] and [IDL_STS_DEVx] bits. See the “Peripheral Device Management” section for more information on device monitor idle timer operation.

Device Monitor Idle Timer Expiration: [IDL_EN_DEVx] [IDL_STS_DEVx]
x=0–11 [DEV_STS]

— The Idle Timers for Device Monitoring subsystem count down and generate an SMI# upon expiration, if enabled. The [DEV_STS] bit is logical “OR” of [TRP_STS_DEVx] and [IDL_STS_DEVx] bits. See the “Peripheral Device Management” section for more information on device monitor idle timer operation.

PIIX4 Master Abort on PCI [P4MA_EN] [PM4A_STS]

— If enabled, a Master Abort to a PIIX4 initiated PCI cycle will generate an SMI#.

Global Release: [BIOS_EN] [BIOS_STS]

— Writes to the Power Management 1 Control Register (PM1_CNTRL) with bit 2 set will generate an SMI# if enabled. See the “Peripheral Device Management” section for more information.

Thermal Alarm (THRM# Assertion): [THRM_EN] [THRM_STS]

— **Polarity Select:** [THRM_POL]

— The THRM# signal will set the [THRM_STS] bit when asserted and if enabled will generate an SMI#. The assertion polarity can be controlled to allow system code to detect when THRM# signal transitions from low to high or high to low. This signal can also be used to generate an SCI. When asserted, the THRM# will also cause automatic clock throttling (see the “Clock Control” section for additional details).

11.5.3. GLOBAL STANDBY TIMER OPERATION

The Global Standby Timer is used to monitor for global system activity during normal operation and can be reloaded by system activity events. When enabled, the timer loads and starts counting down. Enabled system events cause the timer to reload its initial value and begin counting down again. If no system events reload the timer, it will eventually count to zero. Upon this expiration, it generates an SMI#. When the system is placed in a Suspend Mode, the Global Standby Timer can also be used to generate a resume event.

The global standby timer stops counting when the SM_FREEZE bit is set. This can be used to keep it from counting down when the system is executing an SMI routine. The SM_FREEZE bit is disregarded while in a Suspend state, so that the Global Standby Timer counts down independent of SM_FREEZE value.

Global Standby Timer Programming Information:

Resolution: 32 second or 4 minute	[GSTBY_SEL]
Count: 7 bit	[GSTBY_CNT]
Count and SMI# Enable:	[GSTBY_EN]
Expiration Status:	[GSTBY_STS]

Global Standby Timer Reload Events

IRQ1, IRQ12/M:	[GRLD_EN_KBC_MS]
NMI, INIT, IRQ[1,3:7,9:15]:	[GRLD_EN_IRQ]
Device 0–13 Monitors:	[GRLD_EN_DEVx] x=0–13
Video Monitor (PCI Bus Utilization):	[VIDEO_EN]
PCI Bus Master Activity:	[BM_RLD_DEV8]
	[GRLD_EN_DEV8]

11.5.4. SMBUS FUNCTIONAL DESCRIPTION

The System Management Bus (SMBus) is a two-wire interface through which the system can communicate with simple power-related chips. With SMBus, a device can provide manufacturer information, indicate its model/part number, save its state for a suspend event, report different types of errors, accept control parameters, and return status.

PIIX4 provides a SMBus host controller, host controller slave port, and two SMBus slave shadow ports. The SMBus host controller provides a mechanism for the processor to initiate communications with SMBus peripherals. This can be used to configure system devices or to query devices for status. The SMBus slave interface provides a mechanism for other SMBus masters to communicate with PIIX4 and can be used to generate interrupts or resume events for a suspended system. PIIX4 also supports the SMBus ALERT# protocol. PIIX4's SMBus controller has 3.3V input buffers, which requires the system's SMBus to be designed with a 3.3V termination voltage. The programming model is split between function 3 PCI configuration registers and SMBus I/O space registers.

The System Management Bus is a subset of the Phillips* I^2C* protocol.

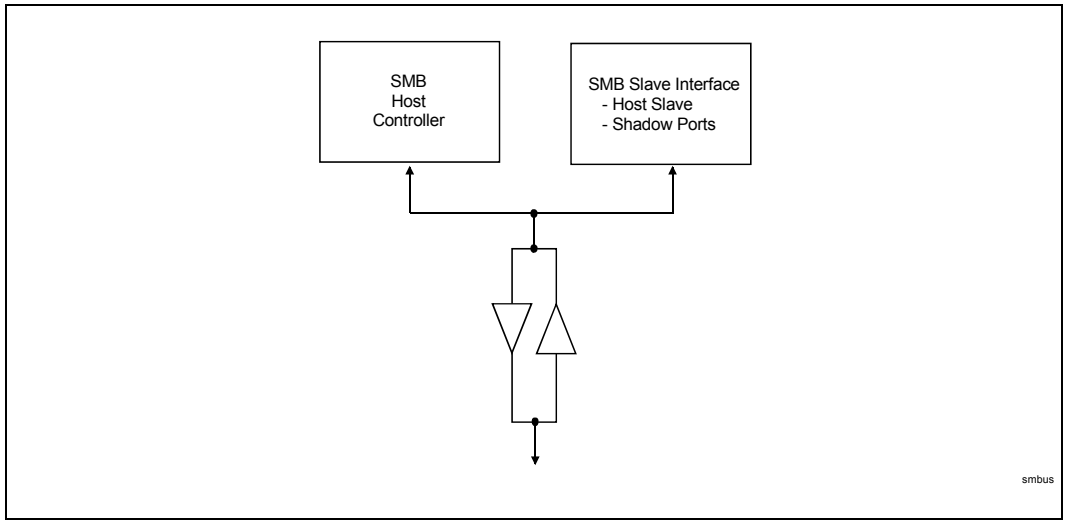


Figure 32. SMBus Interface

11.5.4.1. SMBus Host Interface

A SMBus Host Controller is used to send commands to various SMBus devices. The PIIX4 SMBus controller implements a full host controller implementation. The PIIX4 SMBus controller supports seven command protocols of the SMBus interface (see *System Management Bus Specification, Revision 1.0*): Quick Command, Send Byte, Receive Byte, Write Byte/Word, Read Byte/Word, Block Read, and Block Write.

To execute a SMBus host transaction, the type of transfer protocol, the address of SMBus device, the device specific command, the data, and any control bits are first setup. Then the START bit is set, which causes the host controller to execute the transaction. When the transaction is completed, PIIX4 generates an interrupt, if enabled. The interrupt can be selected between IRQ9 or SMI#. The system software can wait for interrupt to signal completion or it can monitor the HOST_BUSY status bit. An interrupt is also signaled if an error occurred during the transaction or if the transaction was terminated by software setting the KILL bit. The

SMBHSTCNT, SMBHSTCMD, SMBHSTADD, SMBHSTDAT0, SMBHSTDAT1, and SMBBLKDAT registers should not be accessed after setting the START bit while the HOST_BUSY bit is active (until completion of transaction).

The SMBus controller will not respond to the START bit being set unless all interrupt status bits in the SMBHSTSTS register have been cleared.

For Block Read or Block Write protocols, the data is stored in a 32-byte block data storage array. This array is addressed via an internal index pointer. The index pointer is initialized to zero on each read of the SMBHSTCNT register. After each access to the SMBBLKDAT register, the index pointer is incremented by one. For Block Write transactions, the data to be transferred is stored in this array and the byte count is stored in SMBHSTDAT0 register prior to initiating the transaction. For Block Read transactions, the SMBus peripheral determines the amount of data transferred. After the transaction completes, the byte count transferred is located in SMBHSTDAT0 register and data is stored in the block data storage array. Accesses to the array during execution of the SMBus transaction always start at address 0.

Any register values needed for computation purposes should be saved prior to the starting of a new transaction, as the SMBus host controller updates the registers while executing the new transaction.

11.5.4.2. SMBus Slave Interface

PIIX4 supports three separate mechanisms for SMBus peripherals to communicate to PIIX4. In addition to transferring data, these mechanisms can generate an interrupt or resume the system from a suspend state.

The first mechanism consists of accesses to the SMBus controller host slave port at address 10h. (Note this address is actually 0001 000x as this is a 7-bit address (bits[7:1]) with bit 0 being R/W bit.) The host slave port responds to Word Write transactions only with the incoming data being stored in the SMBSLVDAT register and incoming command in the SMBSHDWCMD register. An interrupt or resume event is generated (if enabled) if the incoming command matches the command stored in SMBSLVC register and at least one bit read into the SMBSLVDAT register matches with the corresponding bit in the SMBSLVEVT register.

The second mechanism monitors for accesses to the SMBus controller slave shadow ports at addresses stored in SMBSHDW1 and SMBSHDW2 registers. The shadow slave ports responds to Word Write transactions only with the incoming data being stored in the SMBSLVDAT register and incoming command being stored in the SMBSHDWCMD register. An interrupt or resume event is generated (if enabled) when the slave shadow ports are accessed.

The SLV_BSY bit indicates that the PIIX4 slave interface is receiving an incoming message. The SMBSLVCNT, SMBSHDWCMD, SMBSLVEVT, SMBSLVDAT, and SMBSLVC registers should not be accessed while the SLV_BSY bit is active (until completion of transaction).

The third method for SMBus devices to communicate with PIIX4 is with the SMBALERT# signal. When enabled and the SMBALERT# signal is asserted, PIIX4 generates an interrupt or resume the system from a suspend state. This simple mechanism allows a device without SMBus master capabilities to request service from the SMBus host (PIIX4). To determine which device asserted the SMBALERT# signal, the PIIX4 host controller should be programmed to execute a read command using the Alert Response Address.

Once the slave interface has received a transaction and generated an interrupt, it will stop responding to new requests until all the interrupt status bits in the SMBSLVSTS register are cleared.

11.6. ACPI Support

PIIX4 supports the ACPI I/O Register mapping, the SCI interrupt and the Power Management Timer. PIIX4 also supports a semaphore mechanism to coordinate access to the power management resources by either ACPI or the BIOS.

11.6.1. SCI GENERATION

The THRM#, GPI1#, LID, and PWRBTN# can be enabled to generate the ACPI interrupt, SCI (internal IRQ9) or an SMI#. The SMI# or SCI is selectable with the [SCI_EN] bit. When set to 1, these events generate an SCI, if enabled. When reset, these events generate an SMI#, if enabled. See the “System Management” section for additional details on SMI and the THRM#, GPI1#, LID and PWRBTN# events.

SCI Generation Events

PWRBTN# Asserted:	[PWRBTN_EN]
LID Asserted:	[LID_EN]
— Polarity Select:	[LID_POL]
GPI1 Asserted:	[GPI_EN]
Thermal Alarm (THRM# Assertion):	[THRM_EN]
— Polarity Select:	[THRM_POL]
Power Management Timer Expiration:	[TMROF_EN]
BIOS Release:	[GBL_EN]

11.6.2. POWER MANAGEMENT TIMER

A power management timer is used by the OS to evaluate when the system is idle. The timer consists of a constantly running time base (14.31818 MHz/4 or 3.579545 MHz), a running time base value, and a single interrupt source (Figure 33). The interrupt source indicates that the counter has changed bit 23 (high to low or low to high); this condition generates a System Control Interrupt (SCI). The overflow interrupt is used by software to understand when the timer is about to overflow, and allows software to emulate a much larger timer.

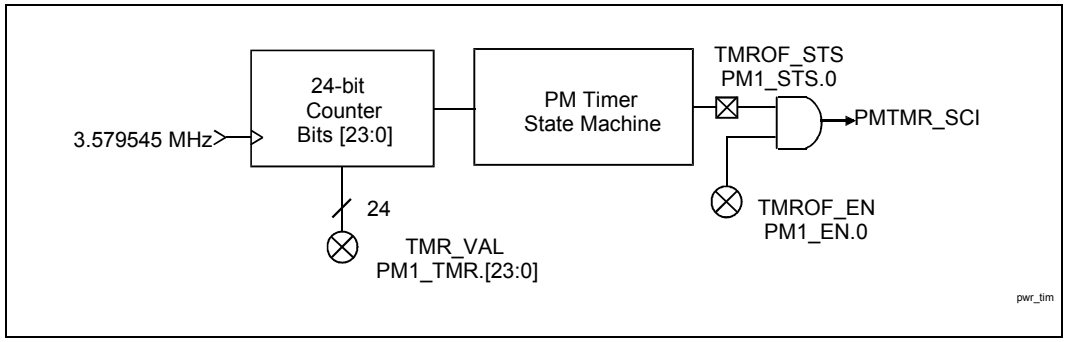


Figure 33. Power Management Timer

Power Management Timer Programming Information:

Clock Frequency: 3.579545 MHz (14.31818 MHz/4 clock)	
Start Count: 24 bit	[TMR_VAL]
Timer Overflow Status:	[TMROF_STS]
SCI Generation	[TMROF_EN]

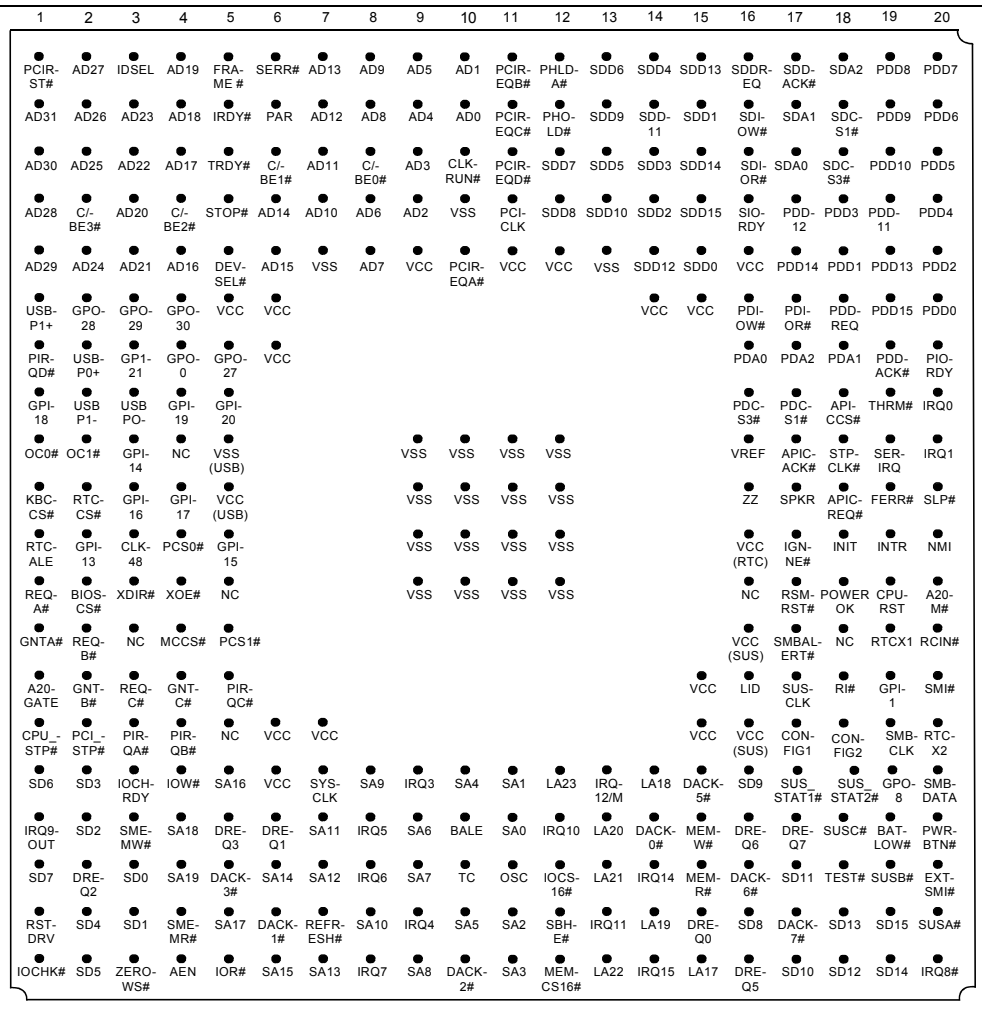
11.6.3. GLOBAL LOCK

If the BIOS and ACPI software share resources via a common I/O port, the Global Lock feature must be used as a semaphore to arbitrate access to these shared resources. For example, if both the BIOS and the ACPI driver use the system management microcontroller control/status ports to manage the system, this access must be controlled using the Global Lock feature.

In the event of a resource conflict, the Global Lock logic is used by the ACPI driver to inform the BIOS driver that it is finished using a shared resource. The BIOS software accesses the GBL_RLS bit to attempt to gain ownership of the lock. This access sets the BIOS_STS bit. ACPI software releases the lock by setting the BIOS_EN bit. PIIX4 then generates an SMI which informs BIOS software that the shared resource is now available.

Likewise, if the ACPI attempts to use the shared resources and there is a conflict, the Global Lock logic is used by the BIOS software to inform the ACPI driver that it is finished using the shared resource. The ACPI software accesses the BIOS_RLS bit to attempt to gain ownership of the lock. This access sets the GBL_STS bit. BIOS software releases the lock by setting the GBL_EN bit. PIIX4 then generates an SCI which informs ACPI software that the shared resource is now available.

12.0. PINOUT INFORMATION



NOTE: For multiplexed pins, only one of the two signal names is shown in this figure. For example, the name for “Y20” only lists IRQ8#(instead of IRQ8#/GPI6). The pin list in Table 55 includes both signal names for the multiplexed pins.

Figure 34. PIIX4 Pinout

Table 55. PIIX4 Alphabetical Pin List

Name	Pin
AD0	B10
AD1	A10
AD10	D7
AD11	C7
AD12	B7
AD13	A7
AD14	D6
AD15	E6
AD16	E4
AD17	C4
AD18	B4
AD19	A4
AD2	D9
AD20	D3
AD21	E3
AD22	C3
AD23	B3
AD24	E2
AD25	C2
AD26	B2
AD27	A2
AD28	D1
AD29	E1
AD3	C9
AD30	C1
AD31	B1
AD4	B9
AD5	A9
AD6	D8
AD7	E8
AD8	B8
AD9	A8
APICACK#	J17
APICCS#	H18

Table 55. PIIX4 Alphabetical Pin List

Name	Pin
APICREQ#/ GPI5	K18
AEN	Y4
A20GATE	P1
A20M#	M20
BALE	U10
BATLOW#/ GPI9	U19
BIOSCS#	M2
C/BE0#	C8
C/BE1#	C6
C/BE2#	D4
C/BE3#	D2
CLK48	L3
CLKRUN#	C10
CONFIG1	R17
CONFIG2	R18
CPURST	M19
CPU_STP#	R1
DACK0#	U14
DACK1#	W6
DACK2#	Y10
DACK3#	V5
DACK5#	T15
DACK6#	V16
DACK7#	W17
DEVSEL#	E5
DREQ0	W15
DREQ1	U6
DREQ2	V2
DREQ3	U5
DREQ5	Y16
DREQ6	U16
DREQ7	U17

Table 55. PIIX4 Alphabetical Pin List

Name	Pin
EXTSMI#	V20
FERR#	K19
FRAME#	A5
GNTA#	N1
GNTB#	P2
GNTC#	P4
GPI1	P19
GPI13	L2
GPI14	J3
GPI15	L5
GPI16	K3
GPI17	K4
GPI18	H1
GPI19	H4
GPI20	H5
GPI21	G3
GPO0	G4
GPO8	T19
GPO27	G5
GPO28	F2
GPO29	F3
GPO30	F4
IDSEL	A3
IGNNE#	L17
INIT	L18
INTR	L19
IOCHK#/GPI0	Y1
IOCHRDY	T3
IOCS16#	V12
IOR#	Y5
IOW#	T4
IRDY#	B5
IRQ0	H20
IRQ1	J20

Table 55. PIIX4 Alphabetical Pin List

Name	Pin
IRQ3	T9
IRQ4	W9
IRQ5	U8
IRQ6	V8
IRQ7	Y8
IRQ8#/GPI6	Y20
IRQ9OUT/ GP029	U1
IRQ10	U12
IRQ11	W13
IRQ12/M	T13
IRQ14	V14
IRQ15	Y14
KBCCS#/ GP026	K1
LA17	Y15
LA18	T14
LA19	W14
LA20	U13
LA21	V13
LA22	Y13
LA23	T12
LID/GPI10	P16
MCCS#	N4
MEMCS16#	Y12
MEMR#	V15
MEMW#	U15
NMI	L20
OC0#	J1
OC1#	J2
OSC	V11
PAR	B6
PCICLK	D11
PCIREQA#	E10

Table 55. PIIX4 Alphabetical Pin List

Name	Pin
PCIREQB#	A11
PCIREQC#	B11
PCIREQD#	C11
PCIRST#	A1
PCI_STP#	R2
PCS0#	L4
PCS1#	N5
PDA0	G16
PDA1	G18
PDA2	G17
PDCS1#	H17
PDCS3#	H16
PDD0	F20
PDD1	E18
PDD10	C19
PDD11	D19
PDD12	D17
PDD13	E19
PDD14	E17
PDD15	F19
PDD2	E20
PDD3	D18
PDD4	D20
PDD5	C20
PDD6	B20
PDD7	A20
PDD8	A19
PDD9	B19
PDDACK#	G19
PDDREQ	F18
PDIOR#	F17
PDIOW#	F16
PHOLD#	B12
PHLDA#	A12

Table 55. PIIX4 Alphabetical Pin List

Name	Pin
PIORDY	G20
PIRQA#	R3
PIRQB#	R4
PIRQC#	P5
PIRQD#	G1
PWRBTN#	U20
PWROK	M18
RCIN#	N20
RI#/GPI12	P18
RSMRST#	M17
RTCALE/ GP025	L1
RTCCS#/ GP024	K2
RTCX1	N19
RTCX2	R20
REFRESH#	W7
REQA#/GPI2	M1
REQB#/GPI3	N2
REQC#/GPI4	P3
RSTDRV	W1
SA0	U11
SA1	T11
SA10	W8
SA11	U7
SA12	V7
SA13	Y7
SA14	V6
SA15	Y6
SA16	T5
SA17	W5
SA18	U4
SA19	V4
SA2	W11
SA3	Y11

Table 55. PIIX4 Alphabetical Pin List

Name	Pin
SA4	T10
SA5	W10
SA6	U9
SA7	V9
SA8	Y9
SA9	T8
SBHE#	W12
SD0	V3
SD1	W3
SD10	Y17
SD11	V17
SD12	Y18
SD13	W18
SD14	Y19
SD15	W19
SD2	U2
SD3	T2
SD4	W2
SD5	Y2
SD6	T1
SD7	V1
SD8	W16
SD9	T16
SDA0	C17
SDA1	B17
SDA2	A18
SDCS1#	B18
SDCS3#	C18
SDD0	E15
SDD1	B15
SDD10	D13
SDD11	B14
SDD12	E14
SDD13	A15

Table 55. PIIX4 Alphabetical Pin List

Name	Pin
SDD14	C15
SDD15	D15
SDD2	D14
SDD3	C14
SDD4	A14
SDD5	C13
SDD6	A13
SDD7	C12
SDD8	D12
SDD9	B13
SDDACK#	A17
SDDREQ	A16
SDIOR#	C16
SDIOW#	B16
SERR#	A6
SIORDY	D16
SERIRQ/GPI7	J19
SLP#	K20
SMBALERT#/ GPI11	N17
SMBCLK	R19
SMBDATA	T20
SMEMR#	W4
SMEMW#	U3
SMI#	P20
SPKR	K17
STOP#	D5
STPCLK#	J18
SUSA#	W20
SUSB#	V19
SUSC#	U18
SUSCLK	P17
SUS_STAT1#	T17
SUS_STAT2#	T18
SYSCLK	T7

Table 55. PIIX4 Alphabetical Pin List

Name	Pin
TC	V10
TEST#	V18
THRM#/GPI8	H19

Table 55. PIIX4 Alphabetical Pin List

Name	Pin
TRDY#	C5
USBP0+	G2
USBPO-	H3

Table 55. PIIX4 Alphabetical Pin List

Name	Pin
USBP1+	F1
USBP1-	H2
XDIR#	M3

Table 55. PIIX4 Alphabetical Pin List

Name	Pin
XOE#	M4
ZEROWS#	Y3
ZZ	K16

Table 56. PIIX4 Pin List (Power, Ground, and No Connects)

Name	Ball
V_{CC}	E9, E11, E12, E16, F5, F6, F14, F15, G6, P15, R6, R7, R15, T6
V_{CC}(RTC)	L16
V_{CC}(SUS)	N16, R16
V_{CC}(USB)	K5
V_{REF}	J16
V_{SS}	D10, E7, E13, J[9:12], K[9:12], L[9:12], M[9:12]
V_{SS}(USB)	J5
No Connect Pins	J4, M5, M16, N3, N18, R5

13.0. PIIX4 PACKAGE INFORMATION

This specification outlines the mechanical dimensions for PIIX4. The package is a 324-pin ball grid array (BGA).

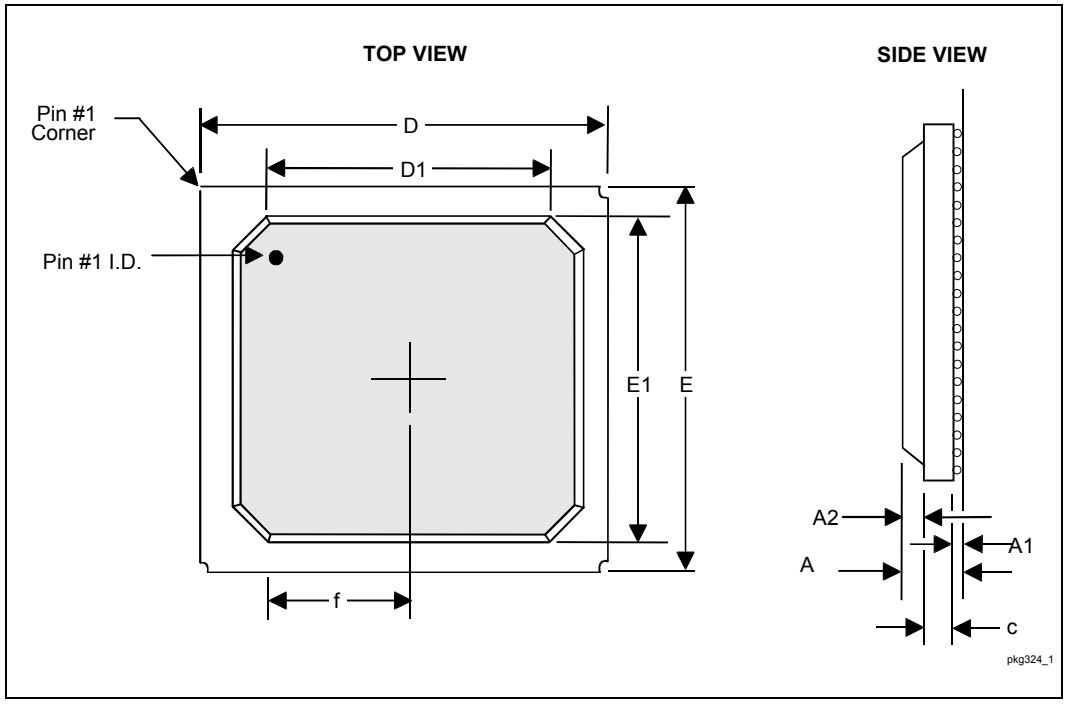


Figure 35. PIIX4 324-Pin Ball Grid Array (BGA)

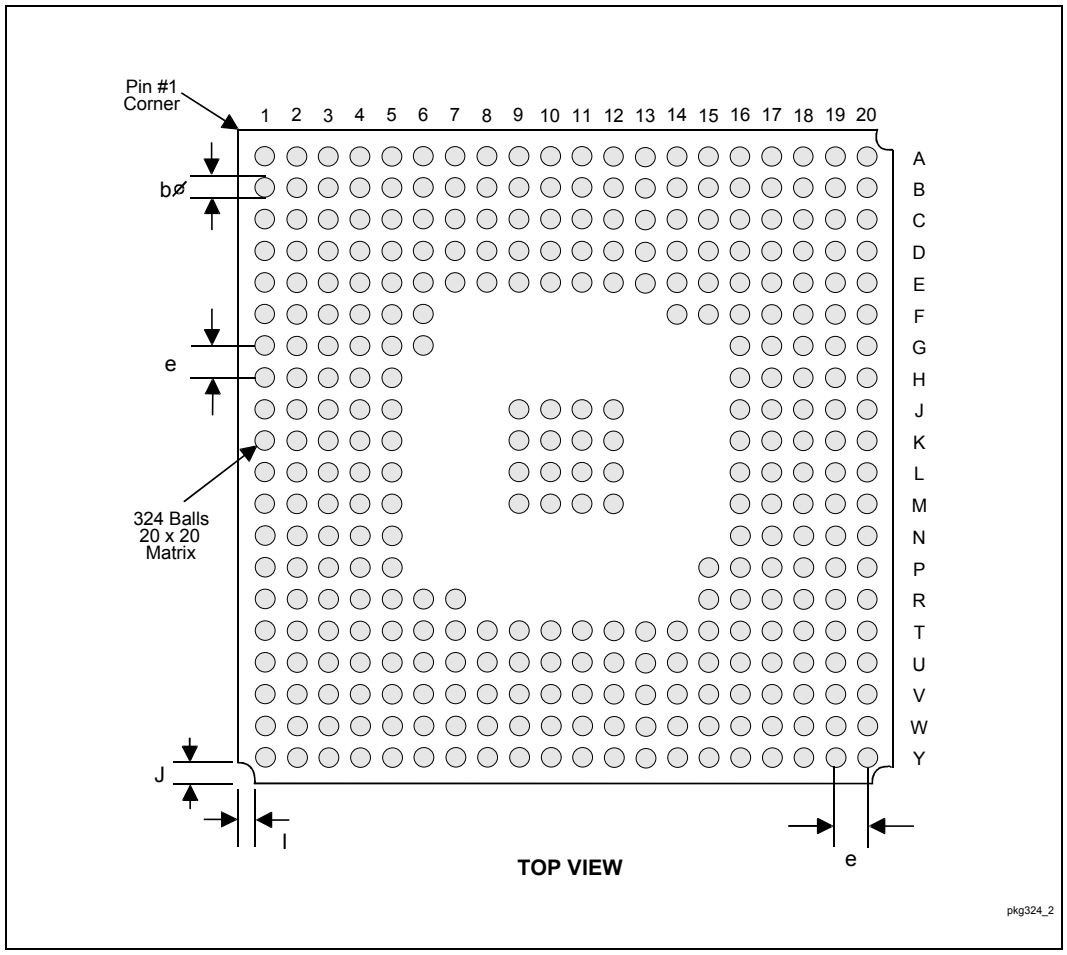


Figure 36. PIIX4 324-Pin Ball Grid Array (BGA) Ball Pattern

Table 57. PIIX4 324-Pin Ball Grid Array (BGA)

Symbol	e=1.27 (solder ball pitch)		
	Min[1]	**Nominal[1]**	**Max[1]**
A	1.90 (2.06)	2.09 (2.29)	2.30 (2.52)
A1	0.50	0.60	0.70
A2	1.12	1.17	1.22
D	26.80	27.00	27.20
D1		24.00	24.70
E	26.80	27.00	27.20
E1		24.00	24.70
I	1.44 REF.		
J	1.44 REF.		
M[2]	20 (Depopulated)		
N[3]	324		
b	0.60	0.76	0.90
c	0.28 (0.44)	0.32 (0.52)	0.38 (0.60)
f	8.05 REF.		
Remark	2 Layer (4 Layer)[4]		

NOTES:

1. All dimensions are in millimeters.
2. 'M' represents the maximum solder ball matrix size.
3. 'N' represents the maximum allowable number of solder balls.
4. PIIX4 is currently shipping with only 2 layer substrate. There are currently no plans to ship 4 layer parts.

14.0. TESTABILITY

14.1. Test Mode Description

PIIX4 supports two types of test modes, a tri-state test mode and a NAND tree test mode. Table 58 lists IRQ decodes necessary to enter each test mode. The test modes are decoded from the IRQ inputs (IRQ[7:3]) and qualified with the TEST# pin.

Table 58. Test Mode Selection

Test Mode	IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	TEST#
No Test Mode Selected	0	0	0	0	0	0
NAND Tree 1	0	0	0	0	1	0
NAND Tree 2	0	0	0	1	0	0
NAND Tree 3	0	0	0	1	1	0
NAND Tree 4	0	0	1	0	0	0
NAND Tree 5	0	0	1	0	1	0
All NAND Trees	1	1	1	1	1	0
Tri-state All Outputs	0	1	0	0	1	0

The entry into a test mode is shown in Figure 37. Each test mode entry sequence is characterized by two active low pulses on the test pin. During the first active low pulse the desired test mode should be driven onto the IRQ lines. The test mode is latched into PIIX4 on the rising edge of the first TEST# pulse. The test mode is actually entered upon the rising edge of the second active low TEST# pulse. To change test modes, the same sequence should be followed again. To restore the PIIX4 to normal operation, execute the sequence with IRQ[7:3]=00000 (No Test Mode Selected).

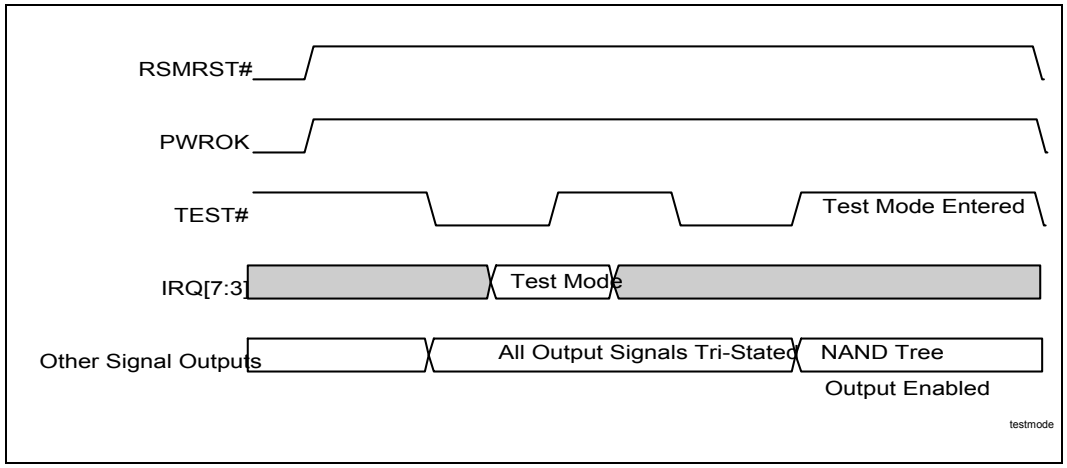


Figure 37. Test Mode Entry (NAND Tree Example)

14.2. Tri-state Mode

When in the tri-state test mode, all outputs and bi-directional pins are tri-stated, including the NAND tree outputs.

14.3. NAND Tree Mode

PIIX4 has five independent NAND trees. Each one can be enabled by itself or all five can be enabled at once.

When a NAND tree is enabled, all output and bi-directional buffers within that tree are tri-stated, except for the NAND tree output. All output and bi-directional buffers for pins not in the selected NAND tree are tri-stated. Every output signal except for each NAND tree's output buffer is configured as an input and is included in the NAND chain. Table 60–Table 64 on page 279 list each NAND tree pin ordering, with the first value being the first input and the last value being the NAND tree output.

Table 65 lists the signal pins not included in any NAND tree.

There are two methods for performing a NAND tree test.

1. The first is to drive all NAND tree input pins to 0. The output of the NAND tree will be a 1. Starting at the last signal in the NAND tree (signal at bottom of list next to output), drive a 1 onto each signal, one at a time. The NAND tree output will toggle on each input signal transitioned from 0 to 1. Allow 500 ns for the input signals to propagate to the NAND tree outputs (input-to-output propagation delay specification).
2. The second method works in reverse. Drive all NAND tree input pins to a 1. Then starting at the first signal in the NAND tree (signal at top of each list), drive a 0 onto each signal, one at a time. The NAND tree output will toggle on each input signal transitioned from 1 to 0. Allow 500 ns for the input signals to propagate to the NAND tree outputs (input-to-output propagation delay specification).

Figure 38 is a schematic of the NAND tree circuitry.

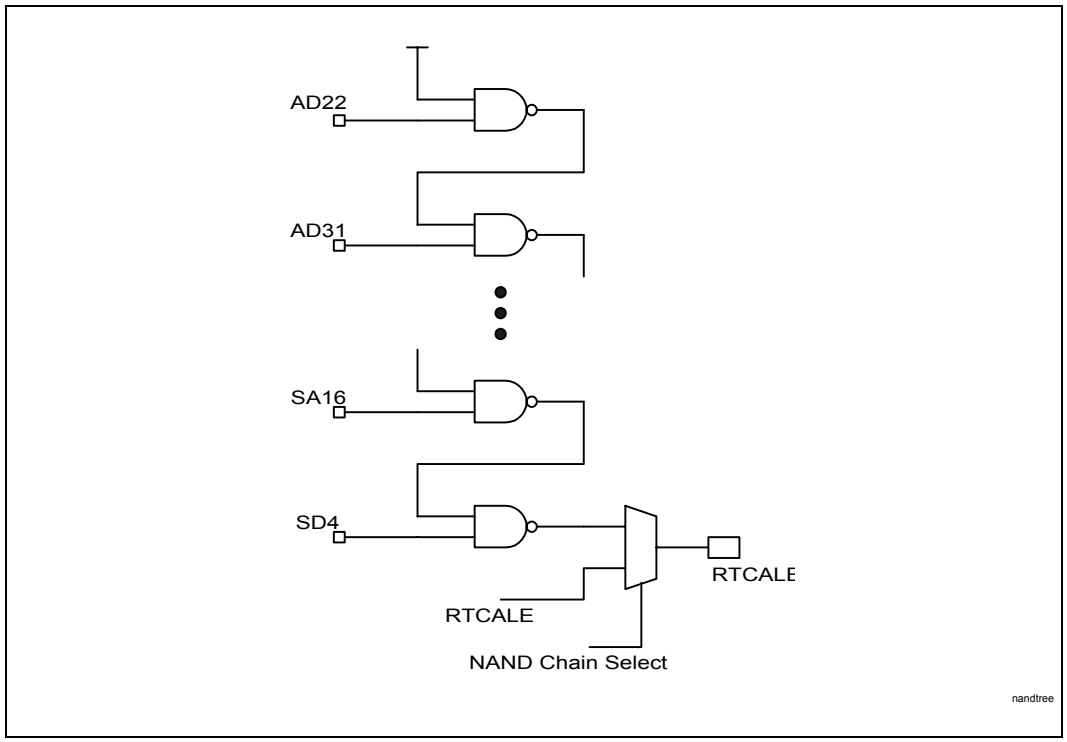


Figure 38. NAND Tree Circuitry (NAND Tree #1 example)

Table 59. NAND Tree Select Pins

Pin Name	Ball Position	Notes
TEST#	V18	Test mode latch signal.
IRQ7	Y8	Test mode select signal.
IRQ6	V8	Test mode select signal.
IRQ5	U8	Test mode select signal.
IRQ4	W9	Test mode select signal.
IRQ3	T9	Test mode select signal.

Table 60. NAND Tree #1 (IRQ[7:3]=00001)

Pin Name	Ball Position	Notes
AD22	C03	
AD31	B01	
AD16	E04	
AD25	C02	
GPO27	G05	
AD20	D03	
AD30	C01	
PCIRST#	A01	
GPI20	H05	
GPO30	F04	
C/BE3#	D02	
AD21	E03	
GPO29	F03	
AD28	D01	
GPO0	G04	
AD24	E02	
GPI21	G03	
GPI19	H04	
GPO28	F02	
AD29	E01	
USBP0P	G02	
USBP0N	H03	
USBP1P	F01	
USBP1N	H02	
GPI14	J03	
PIRQD#	G01	
GPI18	H01	
OC0#	J01	
GPI16	K03	
OC1#	J02	
RTCCS#	K02	
GPI17	K04	
KBCCS#	K01	
PCS0#	L04	

Table 60. NAND Tree #1 (IRQ[7:3]=00001) (Cont'd)

Pin Name	Ball Position	Notes
GPI13	L02	
GPI15	L05	
BIOSCS#	M02	
XDIR#	M03	
REQA#	M01	
REQB#	N02	
GNTA#	N01	
XOE#	M04	
A20GATE	P01	
GNTB#	P02	
MCCS#	N04	
REQC#	P03	
PCS1#	N05	
CPU_STP#	R01	
PCI_STP#	R02	
GNTC#	P04	
SD6	T01	
PIRQA#	R03	
IRQ9	U01	
SD3	T02	
PIRQB#	R04	
SD2	U02	
IOCHRDY	T03	
SMEMW#	U03	
PIRQC#	P05	
SD7	V01	
IOW#	T04	
RSTDRV	W01	
DREQ2	V02	
SA18	U04	
SA16	T05	
SD4	W02	
RTCALE	L01	NAND CHAIN 1 OUTPUT

Table 61. NAND Tree #2 (IRQ[7:3]=00010)

Pin Name	Ball Position	Notes
SD0	V03	
SD5	Y02	
IOCHK#	Y01	
SYSCLK	T07	
DREQ3	U05	
SD1	W03	
SA19	V04	
DREQ1	U06	
ZEROWS#	Y03	
SA9	T08	
DACK3#	V05	
SA14	V06	
SMEMR#	W04	
SA11	U07	
AEN	Y04	
SA17	W05	
SA12	V07	
IRQ5	U08	
DACK1#	W06	
IOR#	Y05	
REFRESH#	W07	
IRQ3	T09	
IRQ6	V08	
SA15	Y06	
SA10	W08	
SA4	T10	
SA6	U09	
SA7	V09	
SA13	Y07	
IRQ7	Y08	
IRQ4	W09	
TC	V10	
SA8	Y09	
SA5	W10	
BALE	U10	
DACK2#	Y10	
SA0	U11	

Table 61. NAND Tree #2 (IRQ[7:3]=00010)(Cont'd)

Pin Name	Ball Position	Notes
SA2	W11	
SA3	Y11	
SA1	T11	
SBHE#	W12	
IOCS16#	V12	
MEMCS16#	Y12	
IRQ11	W13	
LA22	Y13	
LA21	V13	
IRQ10	U12	
IRQ15	Y14	
LA23	T12	
LA19	W14	
LA20	U13	
IRQ14	V14	
IRQ12	T13	
LA17	Y15	
DREQ0	W15	
DACK0#	U14	
DREQ5	Y16	
MEMR#	V15	
SD10	Y17	
MEMW#	U15	
SD8	W16	
DACK7#	W17	
DACK6#	V16	
SD11	V17	
LA18	T14	
SD12	Y18	
DREQ6	U16	
SD13	W18	
DREQ7	U17	
SD14	Y19	
SD15	W19	
SD9	T16	
DACK5#	T15	NAND CHAIN 2 OUTPUT

Table 62. NAND Tree #3 (IRQ[7:3]=00011)

Pin Name	Ball Position	Notes
SUSA#	W20	
IRQ8#	Y20	
SUSC#	U18	
SUS_STAT1#	T17	
LID	P16	
SUSB#	V19	
EXTSMI#	V20	
CONFIG1	R17	
SUS_STAT2#	T18	
CONFIG2	R18	
BATLOW#	U19	
SUSCLK	P17	
PWRBTN#	U20	
RIA#	P18	
SMBALERT#	N17	
SMBCLK	R19	
SMBDATA	T20	
GPI1	P19	
GPO8	T19	NAND CHAIN 3 OUTPUT

Table 63. NAND Tree #4 (IRQ[7:3]=00100)

Pin Name	Ball Position	Notes
SMI#	P20	
RCIN#	N20	
CPURST	M19	
INIT	L18	
A20M#	M20	
INTR	L19	
IGNNE#	L17	
NMI	L20	
APICREQ#	K18	
FERR#	K19	
SLP#	K20	

Table 63. NAND Tree #4 (IRQ[7:3]=00100)(Cont'd)

Pin Name	Ball Position	Notes
ZZ	K16	
SERIRQ	J19	
STPCLK#	J18	
IRQ1	J20	
THRM#	H19	
IRQ0	H20	
APICCS#	H18	
APICACK#	J17	
PIORDY	G20	
PDDAK#	G19	
PDCS1#	H17	
PDA1	G18	
PDCS3#	H16	
PDD0	F20	
PDD15	F19	
PDA2	G17	
PDDRQ	F18	
PDD2	E20	
PDIOR#	F17	
PDD4	D20	
PDD13	E19	
PDD11	D19	
PDD1	E18	
PDD3	D18	
PDA0	G16	
PDD5	C20	
PDD14	E17	
PDD10	C19	
PDIOW#	F16	
PDD12	D17	
PDD6	B20	
PDD9	B19	
PDD8	A19	
PDD7	A20	
SPKR	K17	NAND CHAIN 4 OUTPUT

Table 64. NAND Tree #5 (IRQ[7:3]=00101)

Pin Name	Ball Position	Notes
SDCS3#	C18	
SDD0	E15	
SDA0	C17	
SIORDY	D16	
SDD12	E14	
SDCS1#	B18	
SDD15	D15	
SDA2	A18	
SDIOR#	C16	
SDA1	B17	
SDD14	C15	
SDD2	D14	
SDDAK#	A17	
SDIOW#	B16	
SDD3	C14	
SDD10	D13	
SDD1	B15	
SDDRQ	A16	
SDD11	B14	
SDD5	C13	
SDD13	A15	
SDD9	B13	
SDD8	D12	
SDD7	C12	
SDD4	A14	
SDD6	A13	
PCIREQD#	C11	
PHLDA#	A12	
PCIREQC#	B11	
PCIREQB#	A11	
CLOCKRUN#	C10	
AD0	B10	
AD1	A10	
PCIREQA#	E10	

Table 64. NAND Tree #5 (IRQ[7:3]=00101)(Cont'd)

Pin Name	Ball Position	Notes
AD4	B09	
AD3	C09	
AD5	A09	
AD8	B08	
AD9	A08	
C/BE0#	C08	
AD2	D09	
AD13	A07	
AD12	B07	
AD6	D08	
AD11	C07	
AD7	E08	
SERR#	A06	
PAR	B06	
FRAME#	A05	
AD10	D07	
C/BE1#	C06	
AD14	D06	
AD19	A04	
IRDY#	B05	
AD18	B04	
TRDY#	C05	
AD17	C04	
IDSEL	A03	
STOP#	D05	
AD23	B03	
AD15	E06	
DEVSEL#	E05	
AD27	A02	
C/BE2#	D04	
AD26	B02	
PHOLD#	B12	NAND CHAIN 5 OUTPUT

Table 65. Signals Not in NAND Tree

Pin Name	Ball Position	Notes
CLK48	L03	
OSC	V11	
TEST#	V18	
RTCX2	R20	
RTCX1	N19	
RSMRST#	M17	
PWROK	M18	
PCICLK	D11	