*Preliminary Information*

# AMD-756™

# Peripheral Bus Controller Data Sheet

Publication # **22548** Rev: **B**
Issue Date: **August 1999**

# Contents

# List of Figures

# List of Tables

# Revision History

| Date | Rev | Description |
|---|---|---|
| August 1999 | B | Initial public release |

# 1 Features

The AMD-750™ chipset is a highly integrated system solution designed to deliver outstanding performance for the AMD Athlon™ processor and other AMD Athlon system interface-compatible processors. The AMD-750 chipset consists of the AMD-751™ system controller and the AMD-756™ peripheral bus controller. There is a block diagram of the AMD-750 chipset in Figure 1, on the following page.

This document describes the features and operation of the AMD-756, which contains the following functional units:

- Integrated ISA bus controller
- Enhanced master-mode PCI IDE controller with ultra DMA-33/66 support
- USB controller
- Keyboard/mouse controller
- Real-time clock

There is a block diagram of the AMD-756 in Figure 2, on page 8. Key features of the AMD-756 controller are listed in this chapter.

For a description of the AMD-751, see the *AMD-751 System Controller Data Sheet*, order# 21910.

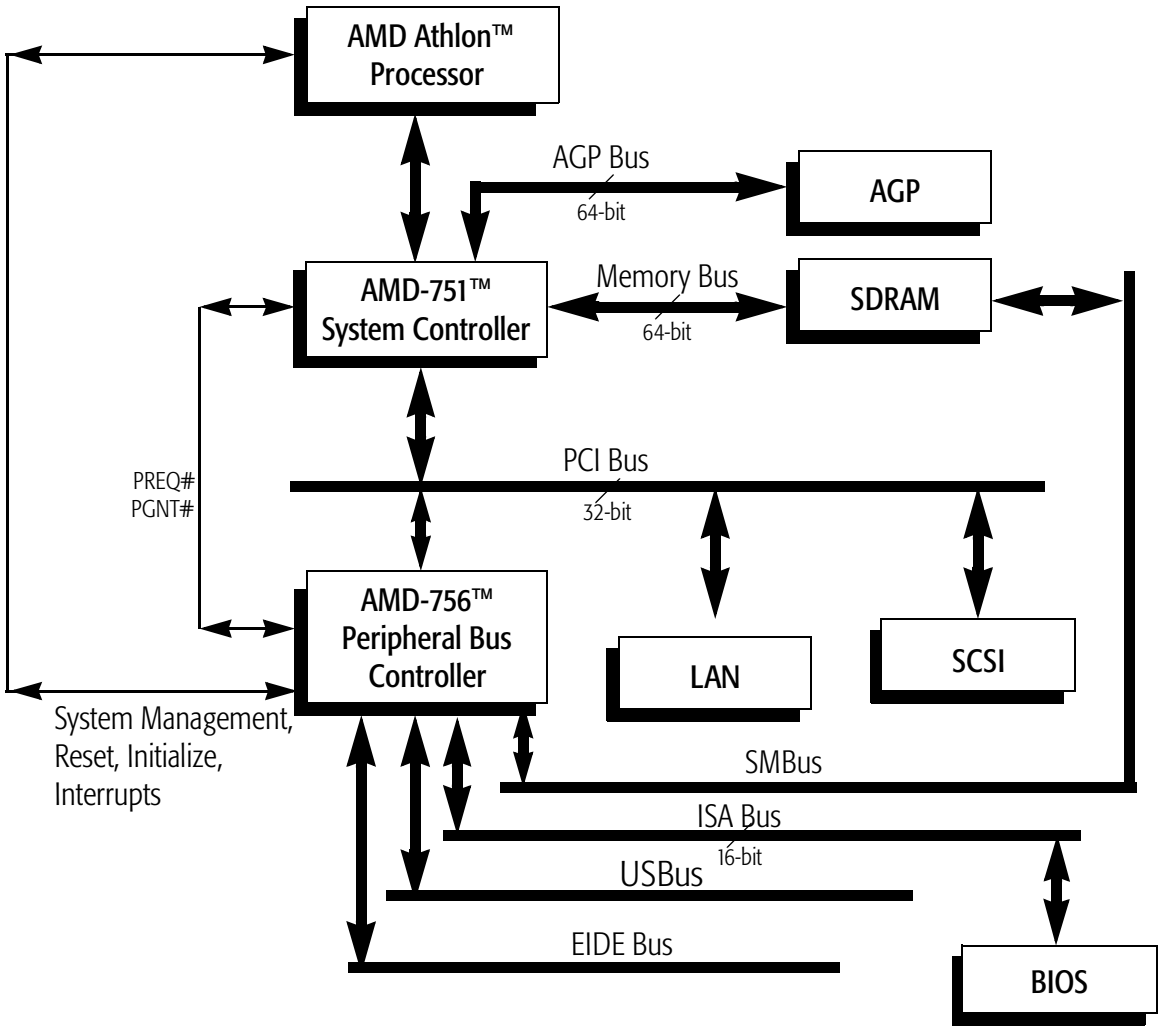


**Figure 1. AMD-750™ Chipset System Block Diagram**

## 1.1 PCI-to-ISA Bridge

The AMD-756 controller includes a PC97-compliant PCI-to-ISA bridge with the following features:

- PCI 2.2-compliant interface
- Eight-level doubleword buffer between PCI and ISA buses
- Dual cascaded AT-8259-compatible interrupt controllers
- Dual AT-8237-compatible DMA controllers
- Type F DMA transfer support
- Support for ISA legacy distributed DMA across the PCI bus
- AT-8254-compatible programmable interval timer
- Integrated real-time clock w/extended 256-byte CMOS RAM
- Programmable ISA bus clock
- Fast reset and gate A20 operation
- Edge-triggered or level-sensitive interrupts
- Flash, 2-Mbyte EPROM, BIOS support
- Integrated keyboard controller with PS/2 mouse support

## 1.2 Enhanced IDE Controllers

The AMD-756 controller includes enhanced master mode PCI and IDE controllers with the following features:

- Ultra DMA-33/66 support for a primary and secondary dual-drive port
- Transfer rates up to 33 Mbytes per second supporting PIO modes 1–4, multi-word DMA mode-2 drivers, and up to 66 Mbytes per second supporting the ultra DMA-66 interface
- Sixteen-level doubleword prefetch and write buffers
- Commands can be interleaved between the two channels
- Bus master programming interface for compliance with SFF-8038i 1.0 and Microsoft® Windows® 95
- Full-featured scatter-gather capability
- Support for ATAPI-compliant devices
- Support for PCI-native and ATA-compatibility modes
- Complete bus mastering software driver support

## 1.3 Universal Serial Bus Controller

The AMD-756 controller includes a universal serial bus (USB) controller with the following features:

- USB 1.0 and OHCI compliant
- Sixteen-level doubleword FIFO for burst PCI bus access
- Root hub and four ports
- Integrated physical-layer transceivers with over-current detection status on USB inputs
- Legacy keyboard and PS/2 mouse support

## 1.4 Plug-n-Play Support

The AMD-756 controller supports plug-n-play with the following features:

- PCI interrupts steerable to any of three interrupt channels
- Microsoft Windows 98 and plug-n-play BIOS compliant
- Serial IRQ compliant

## 1.5 Power Management

The AMD-756 controller includes the following sophisticated power management features:

- Supports advanced configuration and power interface power management (ACPI 1.0 compliant)
- Supports legacy power management (APM 1.2 compliant)
- Supports soft-off and power-on suspend with hardware automatic wakeup
- Two general-purpose timers, one system-inactivity timer, and a 24-bit or 32-bit APCI-compliant timer
- Dedicated external modem-ring input pin for system wakeup
- Normal, doze, sleep, suspend, and conserve modes
- Eighteen multiplexed general-purpose I/O pins

- SMBus implementation for JEDEC-compatible DIMM identification and on-board device power/thermal control
- Primary and secondary interrupt differentiation for individual channels
- Clock throttling control
- Multiple internal and external SMI# sources for flexible power management

# 2 Overview

The AMD-756 peripheral bus controller includes three primary blocks, each with independent access to the PCI bus, a complete set of PCI interface signals and state machines, and capable of working independently with separate devices. These blocks are as follows:

- A PCI-to-ISA bridge, which contains eight doubleword buffers and supports type F DMA transfers to streamline PCI bus operation and to comply with PCI 2.1. These buffers allow for pseudo-split transactions, freeing the PCI bus while ISA transactions complete.
- A USB controller interface with root hub and four ports with built-in physical layer transceivers
- An EIDE controller master mode EIDE controller with full scatter-gather capability

The AMD-756 also integrates many AT-compatible and system control functions, including a keyboard controller with PS/2 mouse support and a real-time clock with extended 256-byte CMOS RAM.

The PCI mux block (see Figure 2) determines which of the primary blocks accesses the PCI bus as a master. It selects the appropriate set of PCI output signals, and it enables the appropriate set of PCI input signals. The mux block includes an arbiter which sets the priority of the primary blocks. The default priority scheme is as follows:

1. PCI-to-ISA (ISA DMA and distributed DMA (DDMA), and system management cycles)
2. USB
3. EIDE

The priorities of the USB and EIDE controllers can be switched (through function 0, offset A4, bit 9[PRISCH]) to improve EIDE performance.

Distributed DMA cycles and cycles to the system management registers are controlled by the PCI state machines located in the PCI-to-ISA bridge. The USB and EIDE blocks each control

one group of configuration registers. The PCI-to-ISA bridge controls two sets of configuration registers—one associated with the PCI-to-ISA bridge (including the DDMA registers) and the other associated with the system management block.

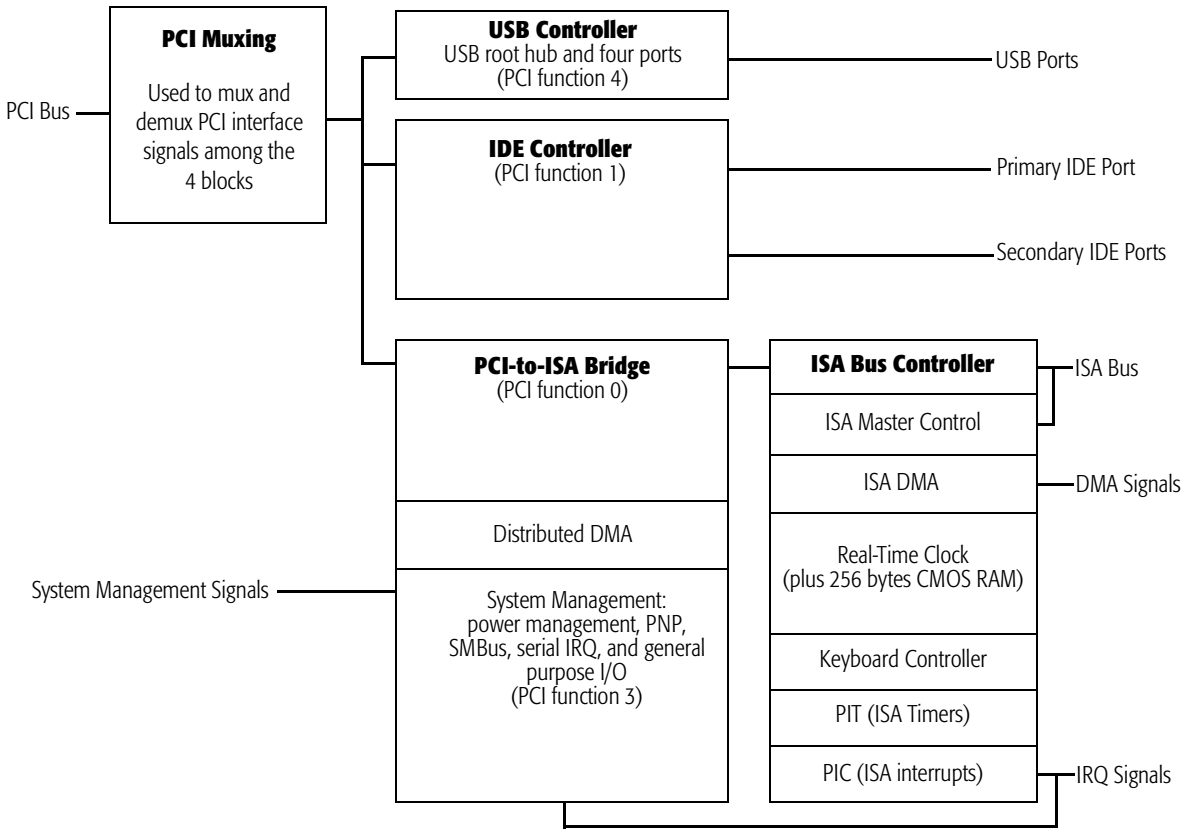


**Figure 2. AMD-756™ Peripheral Bus Controller Block Diagram**

# 2.1 PCI-to-ISA Bridge

The AMD-756 peripheral bus controller offers both PCI-compatible and ISA-compatible bus interfaces. These interfaces, which are PCI 2.1 compliant, control PCI and ISA bus communication. Two main blocks (PCI bus master and target blocks) make up the PCI interface control.

To become the PCI bus master, the AMD-756 peripheral bus controller arbitrates for control of the bus with the AMD-751 system controller. Once bus ownership has been granted, the AMD-756 assumes PCI bus master responsibility. The AMD-756 controller is in target mode when it does not own the PCI bus.

## 2.1.1 PCI Bus Master Mode

The AMD-756 controller arbitrates for bus ownership when an ISA bus resource requests a DMA-controlled transfer between memory and an I/O device or when an ISA bus master requests bus ownership for data transfers. In both DMA and ISA master modes, data transfer takes place either between two ISA bus resources or between an ISA and a PCI bus resource.

The AMD-756 can determine the destination of a bus master request as follows:

- By sampling an active DEVSEL# input. This indicates that a particular target on the PCI bus is responding to the current request.
- By a positive decode of the master-driven address, as a target of an ISA master transfer, to determine the internal destination.
- By subtractive decode, when the access occurs between two ISA bus resources.

The AMD-756 controller PCI interface translates all non-positive-decoded ISA master requests to the PCI bus. If the request is forwarded to the PCI bus, the AMD-756 ensures ISA and PCI bus synchronization by controlling the ISA-based IOCHRDY signal. If the DEVSEL# signal is not received (sampled as active) within the specified time, the AMD-756 controller master interface assumes the requested cycle was

between ISA resources and executes a PCI master abort cycle. If the DEVSEL# signal is received (sampled as active) within the specified time, the AMD-756 master interface executes a data transfer between the ISA and PCI buses.

### 2.1.2 PCI Bus Target Mode

The AMD-756 peripheral bus controller stays in PCI target mode when it does not own the bus. The target interface responds to any request from a PCI resource by asserting DEVSEL# if it positively decodes the current address as a destination either for the ISA bus or the on-chip I/O.

If the AMD-756 controller does not positively decode the current address, the AMD-756 target interface is deselected by an active DEVSEL# input driven by another PCI resource.

If no active DEVSEL# signal is received within a specified time, the AMD-756 acts as the subtractive decode resource by claiming all otherwise unclaimed PCI bus requests and directing the request to the ISA bus.

To ensure correct data synchronization between the two buses on PCI-to-ISA write cycles, the ISA command sequence begins only after the current PCI master has indicated valid data on the bus by asserting IRDY#.

The AMD-756 responds to read requests (and write requests if the posted write buffer is disabled) destined for the ISA bus or on-chip I/O by executing a single data transfer and signalling a target disconnect. If the AMD-756 controller samples an active DEVSEL# input within a specified time, it is deselected, allowing the transfer to take place between the two PCI resources.

The AMD-756 can post PCI-to-ISA memory write cycles. If posting is enabled, the PCI request is acknowledged immediately and the write data is latched to allow the ISA cycle to proceed independently of the PCI transaction.

## 2.2 ISA Bus Controller

The integrated ISA bus address latches and control logic allow the AMD-756 controller user to design a cost-effective system. In addition, the AMD-756 contains the decode logic to select an external keyboard controller.

The AMD-756 device controls accesses to the BIOS and ISA bus ROMs. The BIOS ROM must be 8 bits. All other ROM is accessed as either 8-bit or 16-bit ROM residing on the ISA bus, either on-board or off-board via the slots. Accesses in the C0000h–CFFFFh and E0000h–EFFFFh ranges can be defined as on-board system ROM or off-board memory via the ROM relocation register.

The 82C37A-compatible DMA controllers control data transfers between an I/O channel and on-board or off-board memory. The DMA controllers can transfer data using 24-bit addressing (giving a 16-Mbyte address space). Internal latches latch the middle address bits from the DMA controllers. A memory mapper generates the upper address bits.

The distributed DMA logic remaps I/O cycles to the legacy 82C37A-compatible DMA controller. These cycles are converted by the AMD-756 into PCI master accesses to another PCI device (called the DDMA target). Function 0, offsets 6Eh–60h specify individual enables for the DMA channels and the addresses to which the legacy accesses are redirected. A further description of distributed DMA (DDMA) can be found in Chapter 5.

The AMD-756 controller generates synchronous ISA bus timing and synchronous IDE interface timing from the 33-MHz PCI bus clock.

The AMD-756 performs data steering functions between the ISA and the PCI buses. PCI bus data accesses that are wider than those supported by the targeted ISA bus device are automatically split into two, three, or four ISA cycles. When PCI bus reads are split, the data returned by the ISA devices is assembled by latches before being returned to the PCI bus. The AMD-756 controller also performs low-to-high and high-to-low byte swaps on the 16-bit bus.

As a PCI target, the AMD-756 is capable of expanding PCI accesses with non-contiguous byte enables into the appropriate discrete ISA cycles.

The AMD-756 controller functions are programmable through internal device-specific configuration registers. The state of various interface pins during power-on reset determines the default configuration.

## 2.3 EIDE Controller

The AMD-756 peripheral bus controller's enhanced IDE controller provides a data path and control interface to standard IDE drives. The block is compatible with the ANSI ATA specifications for IDE hard disk operation. The bus mastering IDE interface supports transfer rates up to and beyond mode-4 programmed I/O and mode-2 DMA. Two independent channels are supported, with the ability to connect to both channels with no external logic. Data is transferred over independent 16-bit IDE data buses.

The AMD-756 peripheral bus controller's enhanced IDE interface provides a variety of features to optimize system performance. A 16-doubleword write FIFO and look-ahead read buffer support 32-bit PCI data transfers. The IDE-to-PCI interface operates at PCI speed and enhances system performance by allowing concurrent IDE and PCI operations.

The DMA bus-mastering state machine controls the IOW# and IOR# pulses for each IDE channel during DMA accesses.

The AMD-756 contains two IDE interfaces. Channel 0 is the primary interface, with target I/O addresses at 1F0h–1F7h and 3F6h. The channel 0 IRQ pin is mapped to IRQ14. Channel 1 is the secondary IDE interface, with target I/O addresses at 170h–177h and 376h. The channel 1 IRQ pin is mapped to IRQ15. Unless otherwise noted, when this document refers to one channel's resources the comments apply equally to the other channel.

The master mode registers for both channels are contained in a single I/O block located at the I/O address specified by the contents of the bus master control registers base address

register located at functions 1 and 2, offset 23h–20h. The first 8 bytes of the 16-byte block are associated with channel 0 and the second 8 bytes with channel 1 for each interface. Independent configuration registers exist in PCI configuration space for each channel.

## 2.4 Universal Serial Bus (USB)

The AMD-756 peripheral bus controller's USB host controller interface is compatible with both the USB 1.0 and the open host controller interface (OHIC) specification. The interface uses two sets of software-accessible registers, PCI configuration and USB I/O.

The USB interface supports eighteen levels (doublewords) of data FIFO and a root hub and four ports with built-in physical layer transceivers. The USB controller allows hot insertion of peripherals into the system with universal driver support.

In addition, the AMD-756 offers an emulation mode for legacy keyboard and PS/2 mouse support.

## 2.5 Power Management

The AMD-756 controller supports advanced configuration and power interface (ACPI 1.0) as well as the legacy advanced power management (APM 1.2). In addition, AMD-756 power management is compatible with PC98 and OnNow.

The real-time clock with 256-byte extended CMOS SRAM includes a data alarm and other enhancements for compatibility with the ACPI standard. Two sleep states are provided—soft-off and power-on suspend—as well as hardware automatic wake-up. Additional power management features include event monitoring, processor clock throttling, hardware-based and software-based event handling, general purpose I/O, and external SMI.

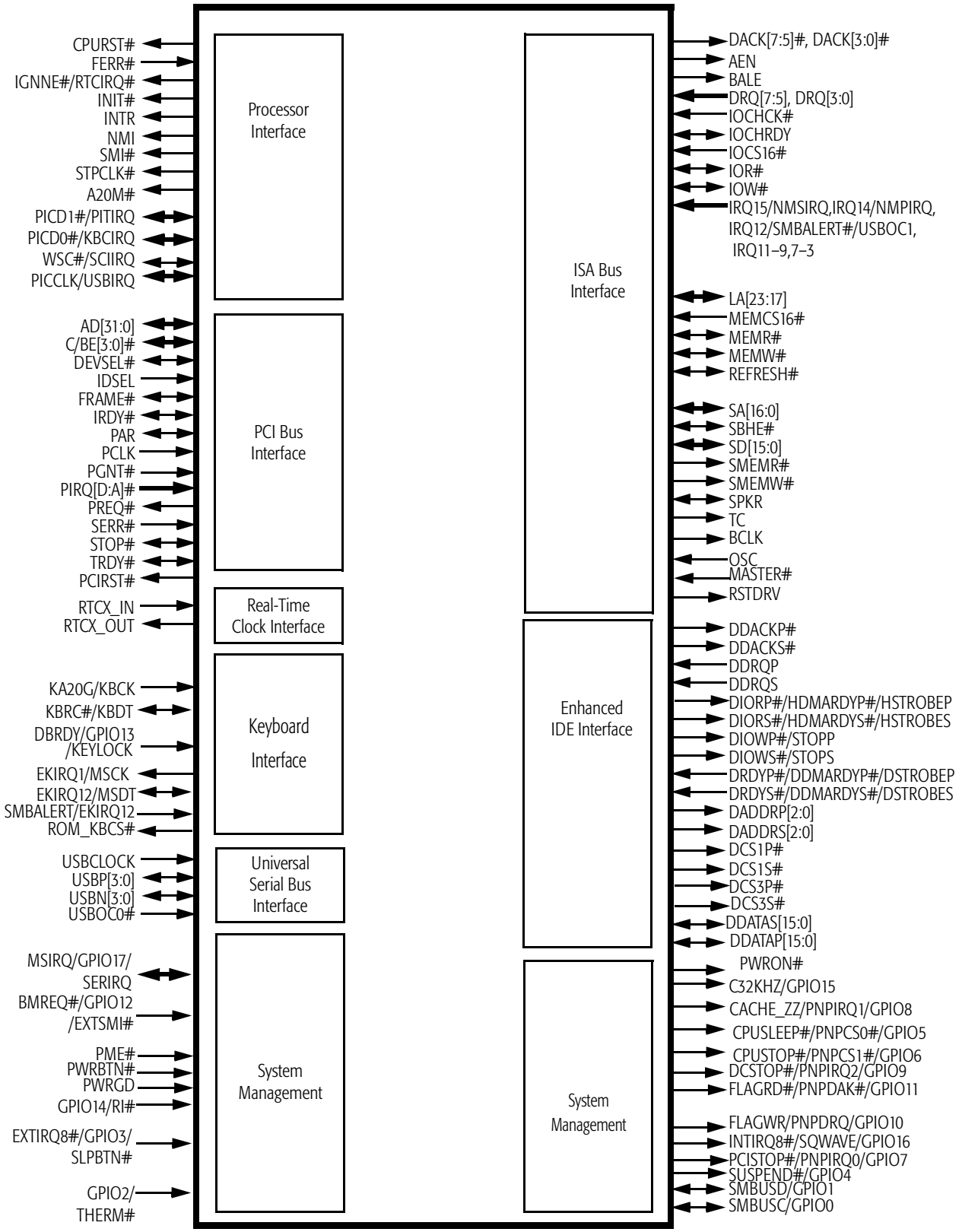


**Figure 3. AMD-756™ Peripheral Bus Controller Signal Groups**

# 3 Ordering Information

AMD standard products are available in several packages and operating ranges. The order number (valid combination) is formed by a combination of the elements below.


**Table 1. Valid Combinations**

| OPN | Package Type | Operating Voltage | Case Temperature |
|---|---|---|---|
| AMD-756 | 272-pin BGA | 3.0 V–3.6 V | 70°C |

***Notes:***

*Valid combinations are configurations that are or will be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations and to check on newly released combinations.*

# 4 Signal Descriptions

## 4.1 Signal Terminology

The following terminology is used in this chapter:

- *Driven*—The processor actively pulls the signal up to the high voltage state or pulls the signal down to the low voltage state.
- *Floated*—The signal is not being driven by the processor (high impedance state), which allows another device to drive this signal.
- *Asserted*—For all active-high signals, the term *asserted* means the signal is in the high voltage state. For all active-low signals, the term *asserted* means the signal is in the low voltage state.
- *Negated*—For all active-high signals, the term *negated* means the signal is in the low voltage state. For all active-low signals, the term *negated* means the signal is in the high voltage state.
- *Sampled*—The processor measures the state of a signal at predefined points in time and takes the appropriate action based on the state of the signal. If a signal is not sampled by the processor, its assertion or negation has no effect on the operation of the processor.

Signals with pound signs (#) are active low.

# 4.2 Processor Interface

## 4.2.1 A20M# (Processor A20 Mask)

### Output

***Summary*** A20M# mask is the mask of processor address bit[20] to the processor.

***Driven*** A20M# is the result of a logical OR of the signals from the keyboard controller output port bit[1] (after being multiplexed between the internal and external keyboard controllers) and PORT92[A20EN]. The keyboard controller output port bit defaults to high at reset and PORT92[A20EN] defaults to low causing this pin to remain high after reset.

## 4.2.2 CPURST# (Processor Reset)

### Output

***Summary*** The AMD-756 peripheral bus controller asserts CPURST# to reset the processor during power-up. When PWRGD is low, CPURST# is asserted. CPURST# is negated 1.8 msecs after PWRGD is asserted.

***Driven*** During power-up, RSTDRV and PCIRST# are driven inactive 1.8 msecs after PWRGD is driven active. CPURST# is driven inactive 1.5 μsecs later. CPURST# can be generated by writing a 1 to the configuration register function 0, offset 47, bit 0. This causes a pulse four PCLK cycles long to be generated on PCIRST# and RSTDRV. CPURST# remains active for three additional PCLK cycles.

## 4.2.3 FERR# (Floating Point Error)

### Input

***Summary*** FERR# is asserted by the processor to indicate that the execution of a floating-point instruction caused the occurrence of an unmasked floating-point exception. This signal is tied to the FERR# signal on the processor.

***Sampled*** If the processor's floating-point unit asserts FERR#, the AMD-756 generates an internal IRQ13, and asserts INTR to the processor. IRQ13 continues to be asserted until a write to port F0h occurs.

### 4.2.4 IGNNE# (Ignore Numeric Exception)

#### Output

***Summary*** IGNNE# is connected to the ignore numeric exception signal on the processor.

***Driven*** When FERR# is asserted, a write to port F0h asserts IGNNE#. IGNNE# is asserted until FERR# is negated.

### 4.2.5 INIT# (Initialization)

#### Output

***Summary*** The AMD-756 controller asserts INIT# if any of the following events occur:

- A shut-down special cycle on the PCI bus
- A soft reset is initiated by the register
- A write to port 92h bit 0

***Driven*** A pulse width of 1.0 to 1.5 µsecs is generated if function 0, offset 47, bit 7 is set.

### 4.2.6 INTR (Processor Interrupt)

#### Output

***Summary*** INTR is driven by the AMD-756 peripheral bus controller to signal the processor that an interrupt request is pending and needs service.

### 4.2.7 NMI (Non-Maskable Interrupt)

**Output**

***Summary*** NMI is used to force a non-maskable interrupt to the processor. The AMD-756 peripheral bus controller generates an NMI when either SERR# or IOCHK# is asserted.

### 4.2.8 SMI# (System Management Interrupt)

**Output**

***Summary*** SMI# is asserted by the AMD-756 peripheral bus controller to alert the processor in response to selected power management events.

### 4.2.9 PICCLK (Interrupt Message Bus Clock)

**Bidirectional**

***Summary*** By programming function 0, offset 4B [APPICCKS] bits, PICCLK can configured to be driven by the AMD-756 peripheral bus controller at the PCLK frequency or slower. It can also be configured to be driven by an external component. This pin can also be used to reveal internal interrupts via function 0, offset 49 [RVLINT] bit.

### 4.2.10 PICD0# and PICD1# (Interrupt Message Data Bits)

**Bidirectional**

***Summary*** These pins are the IOAPIC message data bits. This pin can also be used to reveal internal interrupts via function 0, offset 49 [RVLINT] bit.

### 4.2.11 WSC# (Write Snoop Complete)

#### Bidirectional

***Summary*** This pin connects to the AMD-751 system controller. It is used to signal that the most recent writes to system memory have made it to the coherent space. This signal requires an external 10-200 KOhms pull-up resistor. This pin can also be used to reveal internal interrupts via function 0, offset 49 [RVLINT] bit.

### 4.2.12 STPCLK# (Stop Clock)

#### Output

***Summary*** STPCLK# is asserted by the AMD-756 peripheral bus controller to the processor in response to selected power management events. The processor control register function 3, offset 10h enables throttling and the duty cycle of STPCLK#.

# 4.3 PCI Bus Interface

## 4.3.1 AD[31:0] (PCI Address/Data Bus) Summary

**Bidirectional**

***Summary***

AD[31:0] are the standard PCI address and data lines. They contain a physical address during the first clock of a PCI transaction, and data during subsequent clocks. The address is driven when FRAME# is asserted, and data is driven or received in subsequent cycles.

***Driven, Sampled, and Floated***

**As Outputs:** When the AMD-756 peripheral bus controller is a PCI master, it drives these signals with valid address or data off the rising edge of PCLK. During the first clock that FRAME# is asserted these signals contain address. During subsequent clocks these signals contain data. AD[31:0] become tri-stated at the end of the data phase.

**As Inputs:** When the AMD-756 peripheral bus controller is a PCI slave, these lines are inputs during the address and write data phases of a transaction. The AMD-756 peripheral bus controller samples these signals on the rising edge of PCLK. During the first clock after FRAME# is asserted the bus controller loads the bus contents into the internal address register. On each subsequent clock in which both TRDY# and IRDY# are asserted the bus controller loads data into the data FIFO.

AD[31:0] are floated for one clock in between the address phase and the data phase of a read transfer. AD[31:0] are also floated during RESET and when no initiator is driving the bus.

## 4.3.2 C/BE[3:0]# (PCI Command/Byte Enable)

**Bidirectional**

***Summary***

During the first clock of a PCI transaction, when FRAME# is asserted, these lines contain the PCI bus command (C[3:0]). On subsequent clocks, these lines contain PCI byte enables (BE[3:0]#) corresponding to supplied or requested data.

***Driven, Sampled, and Floated***

**As Outputs:** BE[3:0]# are outputs when the AMD-756 peripheral bus controller is a master. During the address phase of the transaction, C/BE[3:0] define the bus command and are asserted when FRAME# is asserted. C/BE[3:0] are used as byte enables and are asserted during the data phase. C/BE[3:0] become tri-stated at the end of data phase.

**As Inputs:** When the AMD-756 peripheral bus controller is a slave, it samples these lines on the rising edge of PCLK. If FRAME# is asserted, then the bus carries command information.

C/BE[3:0]# are floated during RESET and when no initiator is driving the bus.

### 4.3.3 DEVSEL# (PCI Bus Device Select)

**Bidirectional**

***Summary***

DEVSEL# indicates either that the driving device is the target of the current access or whether any device on the bus has responded to the current address.

***Driven, Sampled, and Floated***

**As Output:** When the AMD-756 is not a PCI bus master, it defaults to target mode, and DEVSEL# is an output indicating that the AMD-756 claims a PCI transaction through either positive or subtractive decoding. In a positive decode, the AMD-756 controller asserts DEVSEL# one PCLK cycle after FRAME# is sampled active and holds DEVSEL# low through the end of the transaction. In a subtractive decode, DEVSEL# is asserted three PCLK cycles after FRAME# is asserted. Positive and negative decoding are explained in Section 5.1.

**As Input:** When the AMD-756 peripheral bus controller is a PCI bus master, DEVSEL# is an input that indicates whether a slave has responded to the current address. If DEVSEL# is sampled inactive in the fourth PCLK cycle after FRAME# is asserted, the AMD-756 aborts the PCI bus cycle.

DEVSEL# is floated during RESET and when no initiator is driving the bus.

### 4.3.4 FRAME# (PCI Bus Cycle Frame)

**Bidirectional**

***Summary*** Asserting FRAME# indicates the address phase of a PCI transfer, and its negation indicates that the cycle initiator wants one more data transfer. While FRAME# is asserted, data transactions can continue. When FRAME# is negated, data transactions are in the final phase.

***Driven, Sampled, and Floated*** **As Output:** When the AMD-756 peripheral bus controller is a PCI bus master, FRAME# is asserted at the beginning of a PCI cycle and is held asserted until the beginning of the last data transfer in the cycle.

**As Input:** When the AMD-756 peripheral bus controller is a slave, it samples and latches the C/BE[3:0]# and AD[31:0] signals and asserts DEVSEL# at the first PCLK on which FRAME# is asserted.

FRAME# is floated during RESET and when no initiator is driving the bus.

### 4.3.5 IDSEL (PCI Initialization Device Select)

**Input**

***Summary*** IDSEL is used as a chip select during configuration read and write cycles.

***Sampled*** IDSEL is sampled at the rising edge of PCLK when FRAME# is asserted.

### 4.3.6 IRDY# (PCI Bus Initiator Ready)

**Bidirectional**

***Summary*** IRDY# is asserted by a PCI initiator from the first clock cycle after FRAME# is asserted to the last clock of the transaction to indicate that the initiator is ready for data transfer. IRDY# asserted during a read cycle indicates the master is ready to accept data. IRDY# asserted during a write cycle indicates that

write data on AD[31:0] is valid. Data is transferred on the PCI bus on each PCLK in which both IRDY# and TRDY# are asserted. Wait states are inserted on the bus until both IRDY# and TRDY# are asserted together.

***Driven, Sampled, and Floated***

**As Output:** When acting as a PCI bus master, the AMD-756 peripheral bus controller asserts IRDY# one PCLK after asserting FRAME# and holds IRDY# asserted until one cycle before the last data transactions. The AMD-756 does not complete a read or write cycle until both IRDY# and TRDY# are sampled asserted.

**As Input:** When acting as a PCI slave, the AMD-756 peripheral bus controller samples IRDY# on every rising edge of PCLK. When IRDY# and TRDY# are both asserted, the bus controller accepts the data. If either signal is negated, the current data is held on the bus.

IRDY# is floated when no bus master is currently driving the bus.

### 4.3.7 PAR (PCI Bus Parity)

#### Bidirectional

***Summary***

PAR, when asserted, indicates even parity. When acting as a PCI master, The AMD-756 peripheral bus controller drives PAR one clock after the address phase and one clock after each data write phase to indicate even parity across AD[31:0] and C/BE[3:0]#. When acting as a PCI target, the AMD-756 drives PAR one clock after each data read phase.

***Driven, Sampled, and Floated***

**As Output:** When the AMD-756 peripheral bus controller is driving data on AD[31:0], PAR is driven one PCLK after an address, read, or write data phase.

**As Input:** When the AMD-756 peripheral bus controller is accepting data on AD[31:0], PAR is sampled one clock cycle after a read is completed.

PAR is only floated when changing bus ownership from one initiator to another.

### 4.3.8 PCIRST# (PCI Reset)

**Output**

***Summary*** PCIRST# is a reset signal for the PCI bus. The AMD-756 peripheral bus controller can assert reset during power-up. A PCI reset can be forced during normal operation by setting configuration register function 0, offset 47h, bit 0.

***Driven*** During power-up, PCIRST# is negated 1.8 msecs after PWRGD is asserted. PCIRST# can also be generated by setting the configuration register function 0, offset 47, bit 0. A pulse four PCLK cycles long is generated on PCIRST#.

### 4.3.9 PCLK (PCI Bus Clock)

**Input**

***Summary*** PCLK provides timing for all transactions on the PCI bus. All signals except PCIRST# and PIRQ[D:A]# are sampled on the rising edge of PCLK, and all timing parameters are defined with respect to this edge. PCLK runs at a frequency up to 33 MHz. PCLK can be divided down to generate the ISA bus clock.

### 4.3.10 PGNT# (PCI Grant)

**Input**

***Summary*** The AMD-751 system controller asserts PGNT# to grant PCI bus access to the AMD-756 peripheral bus controller.

***Sampled*** PGNT# is sampled at the rising edge of PCLK. If PCLK is asserted, the AMD-756 takes control of the bus.

### 4.3.11 PIRQ[D:A]# (PCI Interrupt Requests)

**Input**

***Summary*** Interrupts on the PCI bus are asserted low and are asynchronous. After an interrupt is asserted, it should remain asserted until the device driver clears the pending request. These pins are typically connected to the PCI bus INT lines as shown in Table 2.

**Table 2. Connecting PIRQ Lines to PCI INT Lines**

| | PIRQA# | PIRQB# | PIRQC# | PIRQD# |
|---|---|---|---|---|
| PCI Slot 1 | INTA# | INTB# | INTC# | INTD# |
| PCI Slot 2 | INTB# | INTC# | INTD# | INTA# |
| PCI Slot 3 | INTC# | INTD# | INTA# | INTB# |
| PCI Slot 4 | INTD# | INTA# | INTB# | INTC# |

***Sampled*** PIRQ[D:A]# are sampled at the rising edge of PCLK.

### 4.3.12 PREQ# (PCI Request)

**Output**

***Summary*** The AMD-756 peripheral bus controller asserts PREQ# to request control of the PCI bus.

***Driven*** PREQ# is driven off of the rising edge of PCLK. RESET forces PREQ# inactive. PREQ# is always driven.

### 4.3.13 SERR# (System Error)

**Input**

***Summary*** SERR# reports address parity errors and data parity errors on the special cycle command. SERR# is a synchronous signal. Any PCI device that detects a system error condition can alert the system by asserting SERR# for one PCI clock. SERR# has no timing relationship to any PCI transaction.

***Sampled*** SERR# is sampled at the rising edge of PCLK.

### 4.3.14 STOP# (Stop)

**Bidirectional**

***Summary***

A PCI target asserts STOP# to request that the master stop the current transaction.

***Driven, Sampled, and Floated***

**As Output:** When acting as a PCI slave, the AMD-756 peripheral bus controller asserts STOP# and TRDY# simultaneously to indicate a target disconnect following the data transfer or burst. STOP# is not asserted if the transfer is a single, non-bursted transfer.

**As Input:** When acting as a PCI bus master, STOP# causes the AMD-756 peripheral bus controller to terminate the data transfer and either to abort or to retry the transfer depending on the state of DEVSEL# and TRDY#. STOP# is sampled on every rising edge of PCLK in the data phase of a transaction.

STOP# is floated during RESET and when not being asserted by a target.

### 4.3.15 TRDY# (PCI Target Ready)

**Bidirectional**

***Summary***

TRDY# indicates that the target agent can complete the current data phase of the transaction.

***Driven, Sampled, and Floated***

**As Output:** As a PCI slave, the AMD-756 peripheral bus controller asserts TRDY# to indicate it has sampled the data from the PCI address/data bus during a write phase, or presented valid data on the bus during a read phase.

**As Input:** A PCI target asserts TRDY# when ready for data transfer. When the AMD-756 peripheral bus controller is the PCI bus master, TRDY# is an input that indicates that the target device can complete the data phase of the transaction. After a PCI bus transaction is initiated, the AMD-756 peripheral bus controller inserts wait cycles until TRDY# is sampled active.

TRDY# is sampled on every rising edge of PCLK in the data phase, and is floated when no bus master is driving the bus.

# 4.4 ISA Bus Interface

## 4.4.1 AEN (Address Enable)

**Output**

***Summary*** AEN is asserted during DMA transfer cycles to the I/O resources on the bus to prevent I/O slaves from misinterpreting DMA cycles as valid I/O cycles.

***Driven*** AEN is asserted only when the DMA controller is the bus owner.

## 4.4.2 BALE (Bus Address Latch Enable)

**Output**

***Summary*** BALE is asserted for one bus clock at the beginning of any bus cycle initiated by a PCI master.

***Driven*** BALE is asserted by the AMD-756 peripheral bus controller to indicate that the address signal lines (SA[19:0], LA[23:17], and SBHE#) are valid.

## 4.4.3 BCLK (Bus Clock)

**Output**

***Summary*** BCLK is the reference clock for the ISA bus, and is derived either by dividing PCLK by 2, 3, 4, 5, 6, 10, or 12, or by dividing OSC by 2. BCLK timing is controlled by programming the ISA clock control register, function 0, offset 42h. Bit 3 of that register, the ISA clock select enable bit, is cleared at reset, forcing BCLK to default to the value equal of PCLK/4.

***Driven*** BCLK is always active.

### 4.4.4 DACK[7:5]#, DACK[3:0]# (DMA Acknowledge)

**Output**

***Summary*** These signals, when asserted, indicate that the corresponding request for DMA service has been granted.

***Driven*** The signal is asserted before I/O and memory command lines are asserted and stays asserted until the end of the DMA cycle.

### 4.4.5 DRQ[7:5], DRQ[3:0] (DMA Request)

**Input**

***Summary*** These asynchronous DMA request lines are used by external devices to request services from the AMD-756 peripheral bus controller DMA controller. DRQ[3:0] is used for transfers between 8-bit I/O adapters and system memory. DRQ[7:5] is used for transfers between 16-bit I/O adapters and system memory. DRQ4 is not available externally.

***Sampled*** These signals are sampled at the rising edge of BCLK.

### 4.4.6 IOCHCK# (I/O Channel Check)

**Input**

***Summary*** IOCHCK# is asserted by a device or memory on the ISA bus to indicate that a parity error or other uncorrectable error has occurred.

If I/O checking is enabled and IOCHCK# is sampled asserted, the AMD-756 peripheral bus controller generates an NMI to the processor.

***Sampled*** IOCHCK# is sampled at the rising edge of BCLK.

## 4.4.7 IOCHRDY (I/O Channel Ready)

### Bidirectional

***Summary*** Devices on the ISA bus negate IOCHRDY to indicate that additional time is required to complete the cycle. The cycle can be generated by the processor, DMA controllers, or the refresh controller. The AMD-756 peripheral bus controller responds by inserting wait states to add more time to the cycle.

The default number of wait states for cycles initiated by the processor is as follows:

- 8-bit peripherals 4 wait states
- 16-bit peripherals 1 wait state
- ROM cycles 3 wait states
- DMA cycles 1 DMA wait state

Any peripheral that cannot present read data or strobe in write data in this amount of time must assert IOCHRDY to extend these cycles.

The AMD-756 peripheral bus controller always drives IOCHRDY low in either DMA or master mode to allow for PCI bus latency. This signal is an input while PWRGD is low and used to select the state of function 3, offset 48 [NMLRST].

***Sampled*** IOCHRDY is sampled at the rising edge of BCLK when I/O or memory command lines are active.

## 4.4.8 IOCS16# (16-Bit I/O Chip Select)

### Input

***Summary*** IOCS16# is driven by I/O devices on the ISA bus to indicate that they support 16-bit I/O bus cycles. The AMD-756 peripheral bus controller samples IOCS16# to determine when a processor access requires a 16-bit to 8-bit conversion. The AMD-756 peripheral bus controller also performs a conversion if IOCS16# is sampled high and a 16-bit I/O cycle is requested. In a conversion, the AMD-756 inserts a command delay of one bus cycle and the cycle becomes four wait states long.

***Sampled*** IOCS16# is sampled at the rising edge of BCLK when I/O or memory command lines are active. If IOCS16# is sampled low, the AMD-756 peripheral bus controller performs an I/O access in one wait state, inserting one command delay.

### 4.4.9 IOR# (I/O Read)

**Bidirectional**

***Summary*** IOR# is the command to an ISA I/O slave device indicating the slave can drive data onto the ISA data bus.

***Driven, Sampled, and Floated*** **As Output:** During DMA transfers, IOR# is driven by the DMA controller. IOR# is inactive during a refresh cycle.

**As Input:** IOR# is an input when the AMD-756 peripheral bus controller is bus master and an output at all other times. When the AMD-756 is a PCI slave, IOR# is driven by the internal ISA bus controller.

### 4.4.10 IOW# (I/O Write)

**Bidirectional**

***Summary*** IOW# is the command to an ISA I/O slave device indicating the slave can latch data from the ISA data bus.

***Driven, Sampled, and Floated*** **As Output:** During DMA transfers, IOW# is driven by the DMA controller. IOW# is inactive during a refresh cycle.

**As Input:** IOW# is an input when the AMD-756 peripheral bus controller is bus master and an output at all other times. When the AMD-756 is a PCI slave, IOW# is driven by the internal ISA bus controller.

### 4.4.11 IRQ15, IRQ14, IRQ[12:9], IRQ[7:3] (Interrupt Requests)

**Input**

***Summary*** The IRQ signals provide both system board components and ISA bus I/O devices with a mechanism for asynchronously interrupting the processor.

***Sampled*** IRQs are sampled at the rising edge of BCLK.

### 4.4.12 NMPIRQ (Native Mode Primary IDE Port IRQ)

**Input**

***Summary*** This pin is multiplexed with IRQ14 and is selected by setting function 1, offset 8, bit 8. NMPIRQ is an active high, shared interrupt that is logically combined with PIRQA# such that it can be shared with other PCI devices.

### 4.4.13 NMSIRQ (Native Mode Secondary IDE Port IRQ)

**Input**

***Summary*** This pin is multiplexed with IRQ15 and is selected by setting function 1, offset 8, bit 10. NMSIRQ is an active high, shared interrupt that is logically combined with PIRQA# such that it can be shared with other PCI devices.

### 4.4.14 LA[23:17] (Unlatched Address)

**Bidirectional**

***Summary*** The LA[23:20] address lines are bidirectional and allow accesses to physical memory on the ISA bus up to 16 Mbytes. The LA[19:17] are connected to SA[19:17] pins on the ISA connector.

***Driven, Sampled, and Floated*** **As Outputs:** The LA pins are output when MASTER# is high. The value driven on the LA bus is the address stored in the AD address register during PCI-initiated cycles and the refresh counter during non-ISA bus master refresh cycles.

**As Inputs:** The LA pins are input when MASTER# is low.

### 4.4.15 MASTER# (ISA Master Cycle Indicator)

**Input**

***Summary*** An external bus master device asserts MASTER# to indicate that it has control of the bus.

***Sampled*** MASTER# is sampled at the beginning of an ISA bus cycle.

### 4.4.16 MEMCS16# (16-Bit Memory Chip Select)

**Input**

***Summary*** ISA 16-bit slave memory devices drive this line low to indicate support for 16-bit memory bus cycles.

***Sampled*** This line is sampled to determine when a 16-bit to 8-bit conversion is needed for processor accesses. Conversion is performed when the AMD-756 peripheral bus controller requests a 16-bit memory cycle and MEMCS16# is sampled high. A command delay of one clock cycle is inserted, and the cycle becomes four wait states long. If MEMCS16# is sampled low, a memory access is performed in one wait state with no command delays inserted.

MEMCS16# is ignored for DMA and refresh cycles.

### 4.4.17 MEMR# (Memory Read)

**Bidirectional**

***Summary*** MEMR# is the command to an ISA memory slave device indicating that it can drive data onto the ISA data bus.

***Driven, Sampled, and Floated*** **As Output:** The memory read command signal is asserted after BALE is asserted until the end of the command. MEMR# is driven during memory, DMA, or ISA master cycles.

**As Input:** This signal is an input when an external bus master is in control.

### 4.4.18 MEMW# (Memory Write)

**Bidirectional**

***Summary*** MEMW# is the command to an ISA memory slave device indicating that it can latch data from the ISA data bus.

***Driven, Sampled, and Floated*** **As Output:** The memory write command signal is asserted after BALE is asserted until the end of the command. MEMW# is driven during memory, DMA, or ISA master cycles.

**As Input:** This signal is an input when an external bus master is in control.

### 4.4.19 OSC (Oscillator)

**Input**

***Summary*** OSC is a 14.31818-MHz clock used by the internal timers and the ACPI timer. It can also be used as the source for BCLK.

### 4.4.20 REFRESH# (Refresh)

**Bidirectional**

***Summary*** REFRESH#, when asserted, indicates a refresh cycle is in progress. It enables the SA[7:0] address to drive the row address of the DRAM on the ISA bus. DRAM is refreshed when MEMR# is asserted.

***Driven, Sampled, and Floated*** **As Output:** REFRESH# indicates a refresh cycle is in progress. It is asserted by the AMD-756 peripheral bus controller whenever a refresh cycle is initiated. This signal is driven directly to the ISA bus.

**As Input:** REFRESH# is driven by 16-bit ISA bus masters to indicate a refresh cycle.

### 4.4.21 ROM_KBCS# (ROM and Keyboard Chip Select)

**Output**

***Summary*** During ISA memory cycles, ROM_KBCS# is the chip select to the ROM BIOS. During ISA I/O cycles ROM_KBCS# is the chip select to the external keyboard controller.

### 4.4.22 RSTDRV (Reset Drive)

**Output**

***Summary*** RSTDRV is the reset signal to the ISA bus. It is generated from the received RST signal and is synchronized to PCLK, though it is used for the ISA bus.

***Driven*** During power-up, RSTDRV is inactive 1.8 msecs after PWRGD is driven active. RSTDRV can also be generated by writing a 1 to the configuration register function 0, offset 47, bit 0. A pulse of four PCLK cycles long is generated on RSTDRV.

### 4.4.23 SA[16:0] (System Address Bus)

**Bidirectional**

***Summary*** SA[16:0] address lines, together with LA[23:17], are used to access physical memory on the ISA bus. In I/O accesses, only SA[15:0] are used.

***Driven, Sampled, and Floated*** **As Output:** SA[16:0] are output when the AMD-756 peripheral bus controller owns the bus and are driven valid when BALE is asserted.

**As Input:** SA[16:0] are inputs when an ISA master owns the ISA bus.

### 4.4.24 SBHE# (System Byte High Enable)

**Bidirectional**

***Summary*** When asserted, SBHE# indicates that a byte is being transferred on the upper byte of the ISA data bus (SD[15:8]). SBHE# is negated during refresh cycles.

### 4.4.25 SD[15:0] (ISA System Data)

**Bidirectional**

***Summary*** SD[15:0] provide the data path for devices residing on the ISA bus. The low order ISA lines, SD[7:0], are connected to XD[7:0].

### 4.4.26 SMEMR# (Standard Memory Read)

**Output**

***Summary*** SMEMR# is the command that permits a slave to drive data residing below the 1-MByte region onto the ISA data bus.

***Driven*** SMEMR# is driven during memory, DMA, or ISA master cycles. SMEMR# is a delayed version of MEMR#.

### 4.4.27 SMEMW# (Standard Memory Write)

**Output**

***Summary*** SMEMW# is the command that permits a slave to latch data residing below the 1-MByte region from the ISA data bus.

***Driven*** SMEMW# is driven during memory, DMA, or ISA master cycles. SMEMW# is a delayed version of MEMW#.

### 4.4.28 SPKR (Speaker)

**Bidirectional**

***Summary*** SPKR is the output of counter 2 for the speaker. This signal is an input while PWRGD is low and is used to select the state of function 3, offset 48 [ENIDE, ENPCI, ENISA].

### 4.4.29 TC (Terminal Count)

**Output**

***Summary*** The AMD-756 peripheral bus controller asserts TC to DMA slaves to indicate that one of the DMA channels has transferred all data.

# 4.5 Ultra DMA Enhanced IDE Interface

## 4.5.1 DADDRP[2:0] (Primary IDE Address)

### Output

***Summary*** DADDRP[2:0] is the IDE controller primary port address. It addresses the ATA command or control block.

***Driven*** DADDRP[2:0] is driven before the read or write command signals are asserted to satisfy the address setup time of the IDE drives.

## 4.5.2 DADDRS[2:0] (Secondary IDE Address)

### Output

***Summary*** DADDRS[2:0] is the IDE controller secondary port address. It addresses the ATA command or control block.

***Driven*** DADDRS[2:0] is driven before the read or write command signals are asserted to satisfy the address setup time of the IDE drives.

## 4.5.3 DCS1P# (Primary Port Chip Select)

### Output

***Summary*** DCS1P# is the primary port chip select 1xx. This is active during accesses to the address space specified by function 1, offset 10h, which defaults to I/O addresses 1F7h - 1F0h.

***Driven*** DCS1P# is driven before the read or write command signals are asserted to satisfy the disk chip select setup time of the IDE drives.

### 4.5.4 DCS1S# (Secondary Port Chip Select)

**Output**

***Summary*** DCS1S# is the secondary port chip select 1xx. This is active during accesses to the address space specified by function 1, offset 18h, which defaults to I/O addresses 177h - 170h.

***Driven*** DCS1S# is driven before the read or write command signals are asserted to satisfy the disk chip select setup time of the IDE drives.

### 4.5.5 DCS3P# (Primary Port Chip Select)

**Output**

***Summary*** DCS3P# is the primary port chip select 3xx. This is active during accesses to the address space specified by function 2, offset 14h, which defaults to I/O addresses 3F7h - 3F4h.

***Driven*** DCS3P# is driven before the read or write command signals are asserted to satisfy the disk chip select setup time of the IDE drives.

### 4.5.6 DCS3S# (Secondary Port Chip Select)

**Output**

***Summary*** DCS3S# is the secondary port chip select 3xx. This is active during accesses to the address space specified by function 2, offset 1Ch, which defaults to I/O addresses 377h - 374h.

***Driven*** DCS3S# is driven before the read or write command signals are asserted to satisfy the disk chip select setup time of the IDE drives.

### 4.5.7 DDATAP[15:0] (Primary IDE Data Bus)

**Bidirectional**

***Summary*** DDATAP[15:0] transfers data to or from the primary IDE device.

***Driven, Sampled, and Floated***

**As Outputs:** When the AMD-756 peripheral bus controller is writing to an IDE device, DDATAP[15:0] is driven valid before the negation of the DIOWP# command.

**As Inputs:** When the AMD-756 peripheral bus controller is reading from an IDE device, DDATAP[15:0] is sampled at the rising edge of the DIORP# command. DDATAP[15:0] is tri-stated when no read or write command is in process.

### 4.5.8 DDATAS[15:0] (Secondary IDE Data Bus)

**Bidirectional**

***Summary***

DDATAS[15:0] transfers data to or from the secondary IDE device.

***Driven, Sampled, and Floated***

**As Outputs:** When the AMD-756 peripheral bus controller is writing to an IDE device, DDATAS[15:0] is driven valid before the negation of the DIOWS# command.

**As Inputs:** When the AMD-756 peripheral bus controller is reading from an IDE device, DDATAS[15:0] is sampled at the rising edge of the DIORS# command. DDATAS[15:0] is tri-stated when no read or write command is in process.

### 4.5.9 DDACKP# (Primary IDE DMA Acknowledge)

**Output**

***Summary***

DDACKP# is the primary IDE channel DMA acknowledge. The AMD-756 peripheral bus controller responds to DDRQP either to acknowledge that data has been accepted or to inform that data is available.

***Driven***

DDACKP# is driven at the rising edge of PCLK after sampling DDRQP asserted.

### 4.5.10 DDACKS# (Secondary IDE DMA Acknowledge)

**Output**

***Summary*** DDACKS# is the secondary IDE channel DMA acknowledge. The AMD-756 peripheral bus controller responds to DDRQS either to acknowledge that data has been accepted or to inform that data is available.

***Driven*** DDACKS# is driven at the rising edge of PCLK after sampling DDRQS asserted.

### 4.5.11 DDMARDYP# (Primary Device DMA Ready, Ultra DMA Mode)

**Input**

***Summary*** DDMARDYP# is the primary channel flow control signal for output data bursts, and shares the same pin with DRDYP#. When ready to receive DMA data, a device asserts DDMARDYP#. The device negates DDMARDYP# to pause an Ultra DMA output data transfer.

***Sampled*** DDMARDYP# is sampled at the rising edge of internal PCLK.

### 4.5.12 DDMARDYS# (Secondary Device DMA Ready, Ultra DMA Mode)

**Input**

***Summary*** DDMARDYS# is the secondary channel flow control signal for output data bursts. It shares the same pin with DRDYS#. When ready to receive DMA data, a device asserts DDMARDYS#. The device negates DDMARDYS# to pause an Ultra DMA output data transfer.

***Sampled*** DDMARDYS# is sampled at the rising edge of internal PCLK.

### 4.5.13 DDRQP (Primary IDE DMA Request)

**Input**

***Summary*** DDRQP is the primary IDE channel DMA request. When ready to read or write DMA data, a device asserts DDRQP.

***Sampled*** DDRQP is sampled at the rising edge of PCLK.

### 4.5.14 DDRQS (Secondary IDE DMA Request)

**Input**

***Summary*** DDRQS is the secondary IDE channel DMA request. When ready to read or write DMA data, a device asserts DDRQS.

***Sampled*** DDRQS is sampled at the rising edge of PCLK.

### 4.5.15 DIORP# (Primary I/O Read)

**Output**

***Summary*** DIORP# is the primary IDE channel drive read strobe. The falling edge of DIORP# enables the transfer of data from a register or data port of the drive onto the IDE data bus, DDATAP[15:0].

***Driven*** The rising edge of DIORP# latches the data.

### 4.5.16 DIORS# (Secondary I/O Read)

**Output**

***Summary*** DIORS# is the secondary IDE channel drive read strobe. The falling edge of DIORS# enables the transfer of data from a register or data port of the drive onto the IDE data bus, DDATAS[15:0].

***Driven*** The rising edge of DIORS# latches the data.

### 4.5.17 DIOWP# (Primary I/O Write)

**Output**

***Summary*** DIOWP# is the primary IDE channel drive write strobe.

***Driven*** The rising edge of DIOWP# clocks data from the IDE data bus (DDATAP[15:0]) into either a register or the data port of the drive.

### 4.5.18 DIOWS# (Secondary I/O Write)

#### Output

***Summary*** DIOWS# is the secondary IDE channel drive write strobe.

***Driven*** The rising edge of DIOWS# clocks data from the IDE data bus (DDATAS[15:0]) into either a register or the data port of the drive.

### 4.5.19 DRDYP# (Primary Device Ready)

#### Input

***Summary*** DRDYP# is the primary channel device ready indicator. If a device is not ready to respond to a data transfer request, the device negates DRDYP# to extend the AMD-756 peripheral bus controller read or write cycle. When negated, DRDYP# is in a high impedance state.

***Sampled*** DRDYP# is sampled at the rising edge of PCLK.

### 4.5.20 DRDYS# (Secondary Device Ready)

#### Input

***Summary*** DRDYS# is the secondary channel device ready indicator. If a device is not ready to respond to a data transfer request, the device negates DRDYS# to extend the AMD-756 peripheral bus controller read or write cycle. When negated, DRDYS# is in a high impedance state.

***Sampled*** DRDYS# is sampled at the rising edge of PCLK.

## 4.5.21 DSTROBEP (Primary Device Strobe, Ultra DMA Mode)

### Input

***Summary*** DSTROBEP and DRDYP# share the same pin. DSTROBEP is the primary channel input data strobe signal from the device for an Ultra DMA input data transfer. Both edges of DSTROBEP latch data from DDATAP[15:0] into the host. The device may stop toggling DSTROBEP to pause an Ultra DMA data in transfer.

***Sampled*** DSTROBEP is sampled at the rising edge of PCLK.

## 4.5.22 DSTROBES (Secondary Device Strobe, Ultra DMA Mode)

### Input

***Summary*** DSTROBES and DRDYS# share the same pin. DSTROBES is the secondary channel input data strobe signal from the device for an Ultra DMA input data transfer. Both edges of DSTROBES latch data from DDATAS[15:0] into the host. The device may stop toggling DSTROBES to pause an Ultra DMA data in transfer.

***Sampled*** DSTROBES is sampled at the rising edge of PCLK.

## 4.5.23 HDMARDYP# (Primary Host DMA Ready, Ultra DMA Mode)

### Output

***Summary*** HDMARDYP# and DIORP# share the same pin. HDMARDYP# is the primary channel flow control signal for Ultra DMA input data bursts and is asserted when the host is ready to receive DMA data. The host negates HDMARDYP# to pause an Ultra DMA data in transfer.

***Driven*** HDMARDYP# is asserted after the IDE device asserts DDACKP# to signal that the host is ready to receive data.

### 4.5.24 HDMARDYS# (Secondary Host DMA Ready, Ultra DMA Mode)

**Output**

***Summary*** HDMARDYS# and DIORS# share the same pin. HDMARDYS# is the secondary channel flow control signal for Ultra DMA input data bursts, and is asserted when the host is ready to receive DMA data. The host negates HDMARDYS# to pause an Ultra DMA data in transfer.

***Driven*** HDMARDYS# is asserted after the IDE device asserts DDACKS# to signal that the host is ready to receive data.

### 4.5.25 HSTROBEP (Primary Host Strobe, Ultra DMA Mode)

**Output**

***Summary*** HSTROBEP and DIORP# share the same pin. HSTROBEP is the primary channel strobe signal from the host for an Ultra DMA output data transfer. Both edges of HSTROBEP latch data from DDATAP[15:0] into the device. The host may stop toggling HSTROBEP to pause an Ultra DMA output data transfer.

***Driven*** HSTROBEP is driven after the IDE device asserts DDMARDYP# to signal that it is ready to receive data.

### 4.5.26 HSTROBES (Secondary Host Strobe, Ultra DMA Mode)

**Output**

***Summary*** HSTROBES and DIORS# share the same pin. HSTROBES is the secondary channel strobe signal from the host for an Ultra DMA output data transfer. Both edges of HSTROBES latch data from DDATAP[15:0] into the device. The host may stop toggling HSTROBES to pause an Ultra DMA output data transfer.

***Driven*** HSTROBES is driven after the IDE device asserts DDMARDYS# to signal that it is ready to receive data.

### 4.5.27 STOPP

**Output**

***Summary*** STOPP and DIOWP# share the same pin. STOPP halts data transfer in the primary channel.

***Driven*** The host asserts STOPP before an Ultra DMA burst is initiated and negates STOPP before an Ultra DMA burst is transferred. The host asserts STOPP during or after data transfers in Ultra DMA mode to signal the termination of the burst.

### 4.5.28 STOPS

**Output**

***Summary*** STOPS and DIOWS# share the same pin. STOPS halts data transfer in the secondary channel.

***Driven*** The host asserts STOPS before an Ultra DMA burst is initiated and negates STOPS before an Ultra DMA burst is transferred. The host asserts STOPS during or after data transfer in Ultra DMA mode to signal the termination of the burst.

# 4.6 System Management Pins

## 4.6.1 C32KHZ

**Output**

***Summary*** The C32KHZ clock is available in the FON, C2, C3, and POS power states and is not available in the SOFF or MOFF states. This pin can also be configured as GPIO15 by the I/O mapped register (base pointer: function 3, offset 58h), offset CFh.

## 4.6.2 CACHE_ZZ

**Output**

***Summary*** When CACHE_ZZ is connected to the ZZ input of the second-level cache, the second-level cache enters sleep mode. CACHE_ZZ can be asserted in the C2, C3, or POS states if enabled by function 3, offset 50h. This pin can also be configured as GPIO8 by the I/O mapped register (base pointer: function 3, offset 58h), offset C8h.

## 4.6.3 PNPIRQ1 (Plug and Play Interrupt Request 1)

**Input**

***Summary*** PNPIRQ1 and CACHE_ZZ share the same pin, and are selected by the I/O mapped register (base pointer: function 3, offset 58h), offset C8h. PNPIRQ1 can be assigned to control any of the 12 internal ISA IRQ signals and is configured by function 3, offset 44h.

## 4.6.4 CPUSLEEP#

**Output**

***Summary*** When connected to the sleep pin, asserting CPUSLEEP# places the processor into a non-snoop-capable low power state. This signal can be asserted in the C2, C3, or POS states if enabled by function 3, offset 50h. CPUSLEEP# can also be configured as GPIO5 by the I/O mapped register (base pointer: function 3, offset 58h), offset C5h.

### 4.6.5 PNPCS0# (Plug and Play Chip Select 0)

**Output**

***Summary*** PNPCS0# and CPUSLEEP# share the same pin, and are selected by the I/O mapped register (base pointer: function 3, offset 58h), offset C5h. This signal is used as a programmable chip select to external ISA bus devices. PNPCS0# is active during ISA bus cycles to memory space or I/O space as specified by function 3, offset 46h[CS0M and CS0IS].

### 4.6.6 CPUSTOP# (Processor Clock Stop)

**Output**

***Summary*** CPUSTOP# is connected to the system PLL clock chip to control the host clock signals. CPUSTOP# can be asserted in the C2, C3, or POS states if enabled by function 3, offset 50h. This pin can also be configured as GPIO6 by the I/O mapped register (base pointer: function 3, offset 58h), offset C6h.

### 4.6.7 PNPCS1# (Plug and Play Chip Select 1)

**Output**

***Summary*** PNPCS1# and CPUSTOP# share the same pin, and are selected by the I/O mapped register (base pointer: function 3, offset 58h), offset C6h. This signal is used as a programmable chip select to external ISA bus devices. PNPCS1# is active during ISA bus cycles to memory space or I/O space as specified by function 3, offset 46h [CS1M and CS1IS].

### 4.6.8 DCSTOP# (DRAM Controller Stop)

**Output**

***Summary*** DCSTOP# is connected to the sleep pin of the AMD-751 system controller and is asserted when the host clock is stopping. This enables an alternative DRAM refresh scheme to start. DCSTOP# can be asserted in the C2, C3, or POS states if enabled in function 3, offset 50h. This pin can also be configured as GPIO9 by the I/O mapped register (base pointer: function 3, offset 58h), offset C9h.

### 4.6.9 PNPIRQ2 (Plug and Play Interrupt Request 2)

**Input**

***Summary*** PNPIRQ2 and DCSTOP# share the same pin, and are selected by the I/O mapped register (base pointer: function 3, offset 58h), offset C9h. This signal can be assigned to control any of the 12 internal ISA IRQ signals by function 3, offset 44h.

### 4.6.10 EXTSMI# (External SMI Input)

**Input**

***Summary*** EXTSMI# can be used to generate SMI or SCI interrupts and resume events. This pin can also be configured as GPIO12 by the I/O mapped register (base pointer: function 3, offset 58h), offset CCh.

### 4.6.11 BMREQ# (PCI Bus Master Request)

**Input**

***Summary*** BMREQ# and EXTSMI# share the same pin, and are selected by the I/O mapped register (base pointer: function 3, offset 58h), offset CCh. This signal is the OR of the external PCI bus request signals. This function drives the I/O mapped register (base pointer: function 3, offset 58h), offset 00h[BM_STS] status bit if selected by the I/O mapped register (base pointer: function 3, offset 58h), offset CCh. If this function is not selected by the I/O mapped register (base pointer: function 3, offset 58h), offset CCh, FRAME# is used to drive the status bit. BMREQ# is not guaranteed to meet minimum setup time to the PCI clock and is treated as an asynchronous input.

### 4.6.12 FLAGRD# (Flag Read)

**Output**

***Summary*** FLAGRD# is connected to the output-enable input of one or two external 244-like buffers. The inputs to the external buffers are flags designating which processor read access is required. The outputs drive onto the ISA data bus. FLAGRD# is asserted during reads of the I/O mapped register (base pointer:

function 3, offset 58h), offset 1Ah register so that the external latch inputs are read back to the PCI bus. This pin can also be configured as GPIO11 by the I/O mapped register (base pointer: function 3, offset 58h), offset CBh.

### 4.6.13 PNPDAK# (Plug and Play DMA Acknowledge)

#### Output

***Summary*** PNPDAK# and FLAGRD# share the same pin, and are selected by the I/O mapped register (base pointer: function 3, offset 58h), offset CBh. This pin can be controlled by any of the DACK[7:5, 3:0]# pins that are outputs of the internal DMA controller and is configured by function 3, offset 46h[DMASEL].

### 4.6.14 FLAGWR (Flag Write)

#### Output

***Summary*** FLAGWR is connected to the latch-enabled input of one or two external 373-like latches. The inputs to the external latches are driven by the ISA data bus; the outputs are flags that can be latched off the ISA bus during writes to the I/O mapped register (base pointer: function 3, offset 58h), offset 18h register. This pin can also be configured as GPIO10 by the I/O mapped register (base pointer: function 3, offset 58h), offset CAh.

### 4.6.15 PNPDRQ (Plug and Play DMA Request)

#### Input

***Summary*** PNPDRQ and FLAGWR share the same pin, and are selected by the I/O mapped register (base pointer: function 3, offset 58h), offset CAh. PNPDRQ can be assigned to control any of the DRQ[7:5, 3:0] pins that are inputs to the internal DMA controller and is configured by function 3, offset 46h[DMASEL].

### 4.6.16 INTIRQ8# (Internal Real Time Clock Interrupt)

**Output**

***Summary*** INTIRQ8# is the interrupt from the internal real time clock of the AMD-756 peripheral bus controller. INTIRQ8# is available in the FON, C2, C3, and POS power states and not available in the SOFF or MOFF states. This pin can also be configured as GPIO16 by the I/O mapped register (base pointer: function 3, offset 58h), offset D0h.

### 4.6.17 SQWAVE (Square Wave Clock)

**Output**

***Summary*** SQWAVE and INTIRQ8# share the same pin, and are selected by the I/O mapped register (base pointer: function 3, offset 58h), offset D0h. The frequency of SQWAVE is specified by function 3, offset 4Eh.

### 4.6.18 PCISTOP# (PCI Bus Clock Stop)

**Output**

***Summary*** PCISTOP# is connected to the system PLL clock chip to control the PCI bus clock signals. PCISTOP# can be asserted in the C2, C3, or POS states if enabled by function 3, offset 50h. This pin can also be configured as GPIO7 by the I/O mapped register (base pointer: function 3, offset 58h), offset C7h.

### 4.6.19 PNPIRQ0 (Plug and Play Interrupt Request 0)

**Input**

***Summary*** PNPIRQ0 and PCISTOP# share the same pin, and are selected by the I/O mapped register (base pointer: function 3, offset 58h), offset C7h. PNPIRQ0 can be assigned to control any of the 12 internal ISA IRQ signals and is configured by function 3, offset 44h.

## 4.6.20 PME# (Power Management Interrupt)

### Input

***Summary*** PME# is used to generate SMI or SCI interrupts and resume events.

## 4.6.21 PWRBTN# (Power Button)

### Input

***Summary*** When the system is in the SOFF state, this signal controls the automatic transition to FON. PWRBTN# can be programmed to generate SCI or SMI interrupts from any state other than SOFF. If asserted for four seconds from any state other than SOFF, a power button override event is generated. A power button override event causes the PWRON# pin to be driven high and the I/O mapped register (base pointer: function 3, offset 58h), offset 00h[PBOR_STS] to be set high. This pin has a 16-msec debounce when enabled by function 3, offset 41h[PBDEBNC].

## 4.6.22 PWRGD (Power Good)

### Input

***Summary*** PWRGD is connected to the POWERGOOD signal on the power supply. This is the reset source for the AMD-756 peripheral bus controller when PWRON# is asserted and the system transitions from SOFF to FON. When inactive, the CPURST#, PCIRST#, and RSTDRV pins are driven active. After this signal becomes active, at least 1 msec after power and the PCICLK input are stable, the logic sequences the release of the reset outputs.

### 4.6.23 PWRON# (Main Power On)

**Output**

***Summary*** PWRON# is the output to the power supply that is used to control the main power sources to the system board, including the VDD3 plane of the AMD-756 peripheral bus controller. This pin is low during the FON, C2, C3, and POS states and high during the SOFF state. When power is applied to the VDD_SOFT plane, this signal is reset by the real time clock logic.

### 4.6.24 RI# (Ring Indicator)

**Input**

***Summary*** RI# can be connected to external modem circuitry to allow the system to be reactivated by a received phone call.

RI# causes the system to resume to the FON state and generates SCI or SMI interrupts. RI# operation is reflected in register bits the I/O mapped register (base pointer: function 3, offset 58h), offset 16h[RI_RSM], 22h[RI_EN], 26h[RI_CTL], 28h[RI_STS], and 2Ah[RISMI_EN]. This pin can also be configured as GPIO14 by the I/O mapped register (base pointer: function 3, offset 58h), offset CEh.

### 4.6.25 SERIRQ (Serial IRQ)

**Bidirectional**

***Summary*** This pin supports the VESA serial-IRQ protocol. SERIRQ is controlled by function 3, offset 4Ah. This pin can also be configured as GPIO17 by the I/O mapped register (base pointer: function 3, offset 58h), offset D1h.

### 4.6.26 MSIRQ (Mouse Interrupt Request)

**Output**

***Summary*** MSIRQ and SERIRQ share the same pin and are selected by the I/O mapped register (base pointer: function 3, offset 58h), offset D1h. This signal is the mouse IRQ from the keyboard controller that normally connects to the PIC IRQ12.

### 4.6.27 SMBALERT# (SMBus Alert)

**Input**

***Summary*** SMBALERT# and IRQ12 share the same pin, and are selected by function 3, offset 46h[10:9]. When enabled, SMBALERT# can be used to generate an SMI or SCI interrupt associated with the SMBus logic.

***Sampled*** SMBALERT# is sampled at the rising edge of BCLK.

### 4.6.28 SMBUSC (System Management Bus Clock)

**Bidirectional**

***Summary*** SMBUSC is the system management bus (SMBus) clock. This pin can also be configured as GPIO0 by the I/O mapped register (base pointer: function 3, offset 58h), offset C0h.

### 4.6.29 SMBUSD (System Management Bus Data)

**Bidirectional**

***Summary*** SMBUSD is the system management bus (SMBus) data. This pin can also be configured as GPIO1 by the I/O mapped register (base pointer: function 3, offset 58h), offset C1h.

### 4.6.30 SLPBTN# (Sleep Button)

**Input**

***Summary*** SLPBTN# causes the system to transition between the POS and FON states. If SLPBTN# is asserted, the I/O mapped register (base pointer: function 3, offset 58h), offset 00h[SLPBTN_STS] is set. Enabling SLPBTN# by the I/O mapped register (base pointer: function 3, offset 58h), offset 02h[PWRBTN_EN] causes an SMI or SCI interrupt. If the I/O mapped register (base pointer: function 3, offset 58h), offset 26h[SLPBTN_CTL] bit is set, this pin transitions the system from SOFF to FON. This pin can also be configured as GPIO3 by the I/O mapped register (base pointer: function 3, offset 58h) offset C3h.

## 4.6.31 EXTIRQ8# (Real Time Clock Interrupt)

### Input

***Summary*** EXTIRQ8# and SLPBTN# share the same pin, and are selected by the I/O mapped register (base pointer: function 3, offset 58h), offset C3h. If an external real time clock is used, this signal can be used as the IRQ8# input from that source. EXTIRQ8# can also wake the system from SOFF, C2, C3, and POS. If selected, EXTIRQ8# originates directly from the pad, not from the GPIO input path.

## 4.6.32 SUSPEND# (Processor Suspend)

### Output

***Summary*** This signal is activated during the POS state to control the external power planes. This pin can also be configured as GPIO4 by the I/O mapped register (base pointer: function 3, offset 58h), offset C4h.

## 4.6.33 THERM# (Thermal Warning Detect)

### Input

***Summary*** If enabled by function 3, offset 50h[TTH_EN], this signal automatically enables throttling as specified by function 3, offset 50h[TTH_RATIO]. This pin can also be configured as GPIO2 by the I/O mapped register (base pointer: function 3, offset 58h), offset C2h.

# 4.7 Universal Serial Bus Interface

## 4.7.1 USBCLK (Universal Serial Bus Clock)

### Input

***Summary*** A 48 MHz clock provides timing for both the full speed (12 Mbps) and low speed (1.5 Mbps) operation.

## 4.7.2 USBP[3:0] (USB Port [3:0] Data Positive)

### Bidirectional

***Summary*** USBP[3:0] are the positive signals to differential output drivers that drive the USB port [3:0] data signals on to the USB cable. They support both the full speed (12 Mbps) and low speed (1.5 Mbps) data rates.

## 4.7.3 USBN[3:0] (USB Port [3:0] Data Negative)

### Bidirectional

***Summary*** USBN[3:0] are the negative signals to differential output drivers that drive the USB port[3:0] data signals on to the USB cable. They support both the full speed (12 Mbps) and low speed (1.5 Mbps_ data rates.

## 4.7.4 USBOC0# (USB Over-Current Detect 0)

### Input

***Summary*** Input. USBOC0# is used to monitor the USB power over-current.

***Sampled*** USBOC0# is sampled at the rising edge of BCLK.

### 4.7.5 USBOC1# (USB Over-Current Detect 1)

**Input**

***Summary*** USBOC1# and IRQ12 share the same pin, and are selected by function 3, offset 46h[10:9]. When enabled, it can be routed to be a second source of USB port over-current detection.

***Sampled*** USBOC1# is sampled at the rising edge of BCLK.

# 4.8 Keyboard Interface

## 4.8.1 KBCK (Keyboard Clock)

**Input**

***Summary*** When the internal keyboard controller is enabled, this pin functions as the clock to the keyboard interface.

## 4.8.2 KA20G (Keyboard Gate A20)

**Input**

***Summary*** KA20G and KBCK share the same pin, and are selected by function 3, offset 48h[INTKBC]. The gate A20 function from an external keyboard is summed with the other sources of gate A20 before being passed directly to the processor via A20M#. This pin is a direct connection to A20M# on the processor.

## 4.8.3 KBDT (Keyboard Data)

**Output**

***Summary*** KBDT functions as the data line to the keyboard interface.

## 4.8.4 KBRC# (Keyboard Reset)

**Input**

***Summary*** KBRC# and KBDT share the same pin, and are selected by function 3, offset 48h [INTKBC]. KBRC# functions as a reset input from the external keyboard controller that is used to generate a pulse on CPURST# or INIT[1:0]#.

### 4.8.5 KEYLOCK (Keyboard Lock)

**Input**

***Summary*** KEYLOCK is the keyboard lock signal for the internal keyboard controller. This pin can be configured as GPIO13 by the I/O mapped register (base pointer: function 3, offset 58h), offset CDh.

### 4.8.6 DBRDY (Debug Ready)

**Input**

***Summary*** DBRDY and KEYLOCK share the same pin, and are selected by the I/O mapped register (base pointer: function 3, offset 58h), offset CDh. When this signal is active, the AMD-756 peripheral bus controller halts the timers specified by function 3, offset 4Ch. This allows the processor to execute debugging and performance-monitoring code while the rest of the system is halted.

### 4.8.7 MSCK (Mouse Clock)

**Output**

***Summary*** MSCK is the mouse clock to the legacy mouse connector.

### 4.8.8 MSDT (Mouse Data)

**Bidirectional**

***Summary*** MSDT is the mouse data to the legacy mouse connector.

### 4.8.9 EKIRQ1 (External Keyboard Controller IRQ1)

**Output**

***Summary*** EKIRQ1 and MSCK share the same pin, and are selected by function 3, offset 48h[INTKBC]. This signal sends the external keyboard controller IRQ1 state to the internal interrupt controller logic.

## 4.8.10 EKIRQ12 (External Keyboard Controller IRQ12)

### Output

***Summary*** EKIRQ12 and MSDT share the same pin, and are selected by function 3, offset 48h[INTKBC]. This signal is the external keyboard controller IRQ12 sent to the internal interrupt controller logic. Refer to function 3, offset 46h[10:9] for information on how the IRQ12 pin and the mouse interrupt are combined.

# 4.9 Internal Real-Time Clock

## 4.9.1 RTCX_IN (Crystal/Oscillator Input)

**Input**

***Summary*** RTCX_IN is the 32.768-KHz crystal input. This pin can be driven by a CMOS driver or connected through a crystal oscillator to RTCX_OUT.

*Note: Whether an internal or external real-time clock is selected, this pin is required to provide the 32-KHz clock to the AMD-756 peripheral bus controller.*

## 4.9.2 RTCX_OUT (Crystal/Oscillator Output)

**Output**

***Summary*** RTCX_OUT is the 32.768-KHz crystal output.

# 4.10 Power and Ground

## 4.10.1 GND (Power Ground)

## 4.10.2 GND_USB (USB Differential Output Ground)

## 4.10.3 $V_{DD3}$ (Power Supply for the Processor I/O Voltage)

This pin should be connected to the same voltage as the processor I/O circuitry.

## 4.10.4 $V_{DD\_REF}$ (Power Reference)

## 4.10.5 $V_{DD\_RTC}$ (Power Supply to RTC)

This supply is connected to the real time clock.

## 4.10.6 $V_{DD}$-SOFT (Power Supply)

$V_{DD}$-SOFT is always available unless the mechanical switch of the power supply is turned off.

## 4.10.7 $V_{DD}$-USB (USB Differential Output Power)

# 5 Functional Operations

## 5.1 PCI Bus-Initiated Accesses

The AMD-756 peripheral bus controller is responsible for decoding PCI bus requests from PCI bus masters, initiating the requested actions, and responding as required by the PCI bus protocol.

### 5.1.1 Overview

The AMD-756 responds to PCI bus cycles by positive decoding or subtractive decoding as defined in the PCI specification. The length of the time for both positive and subtractive decoding is fixed at medium decoding.

The AMD-756 controller also generates an ISA bus cycle for any memory or I/O cycle claimed by the ISA function.

## 5.2 PCI Bus Commands

The AMD-756 responds to the PCI bus commands as described in the following sections.

### 5.2.1 Interrupt Acknowledge

The AMD-756 controller releases an 8-bit interrupt vector on AD[7:0] in response to an interrupt acknowledge cycle.

### 5.2.2 Special Bus Cycles

The AMD-756 controller ignores all but two PCI special bus cycles. When the AMD-756 decodes a Shutdown (0000_0000h during the address phase) the controller will assert INIT to the processors. In response to a Stop-Grant (0012_0002h during the

address phase) the controller receives and enters the appropriate power state. The AMD-756 can then assert DCSTOP to the AMD-751 system controller to signal that it should deassert CKE to the SDRAMs and stop its internal clocks.

### 5.2.3 I/O Read and Write

All I/O accesses not claimed by other PCI targets through asserting DEVSEL# are passed to the ISA bus controller and executed as standard ISA bus cycles. The AMD-756 steers the data among the PCI AD bus and the ISA SD bus or the IDE data bus, as required by the cycle type. If the access is to an on-chip I/O location, then the data is steered among the AD bus, the SD bus, and the selected internal location, as required by the cycle type.

The AMD-756 controller asserts TRDY# when all ISA bus accesses are completed. In the case of I/O reads, valid data is placed on the PCI AD bus before TRDY# is asserted. The timing of a PCI cycle forwarded to the ISA bus is shown in Figure 6 on page 67.

An 8-bit cycle is four wait states long while a 16-bit cycle has no wait states if the default configuration is used. Additional wait states can be inserted by setting bit 5 or bit 4 of the ISA Bus Control register, or by negating IOCHRDY. Figure 4 on page 65 illustrates I/O accesses for both read and write, including the insertion of wait states.

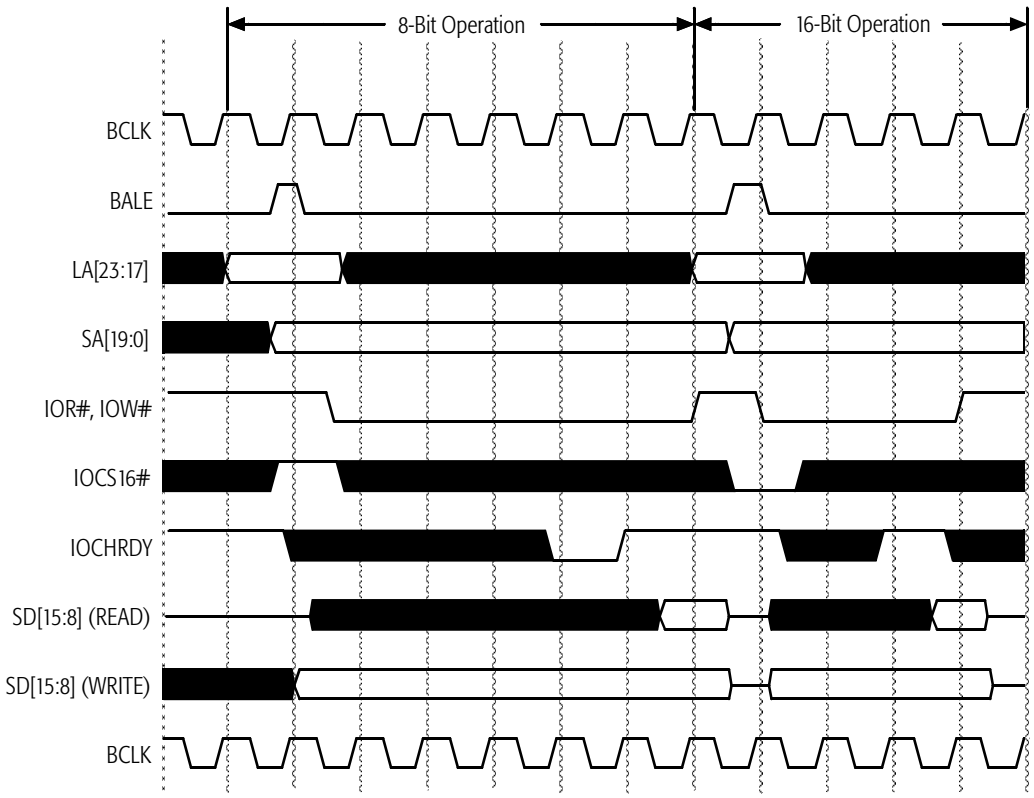


**Figure 4. I/O Access**

For 32-bit or 24-bit accesses to 16-bit ISA bus targets, or for 32-bit, 24-bit, or 16-bit accesses to 8-bit ISA bus targets, the AMD-756 controller generates multiple ISA bus cycles for each PCI bus cycle in order to match the size of the access requested by the PCI initiator. Requests for non-contiguous bytes are handled by converting the access to the appropriate ISA bus cycles. Converting a single PCI cycle to multiple ISA cycles is invisible to the PCI interface, except for the increased latency required to complete the operation. The AMD-756 converts a CPU request for 16-bit data from an 8-bit peripheral into two 8-bit cycles as depicted in Figure 5 on page 66.

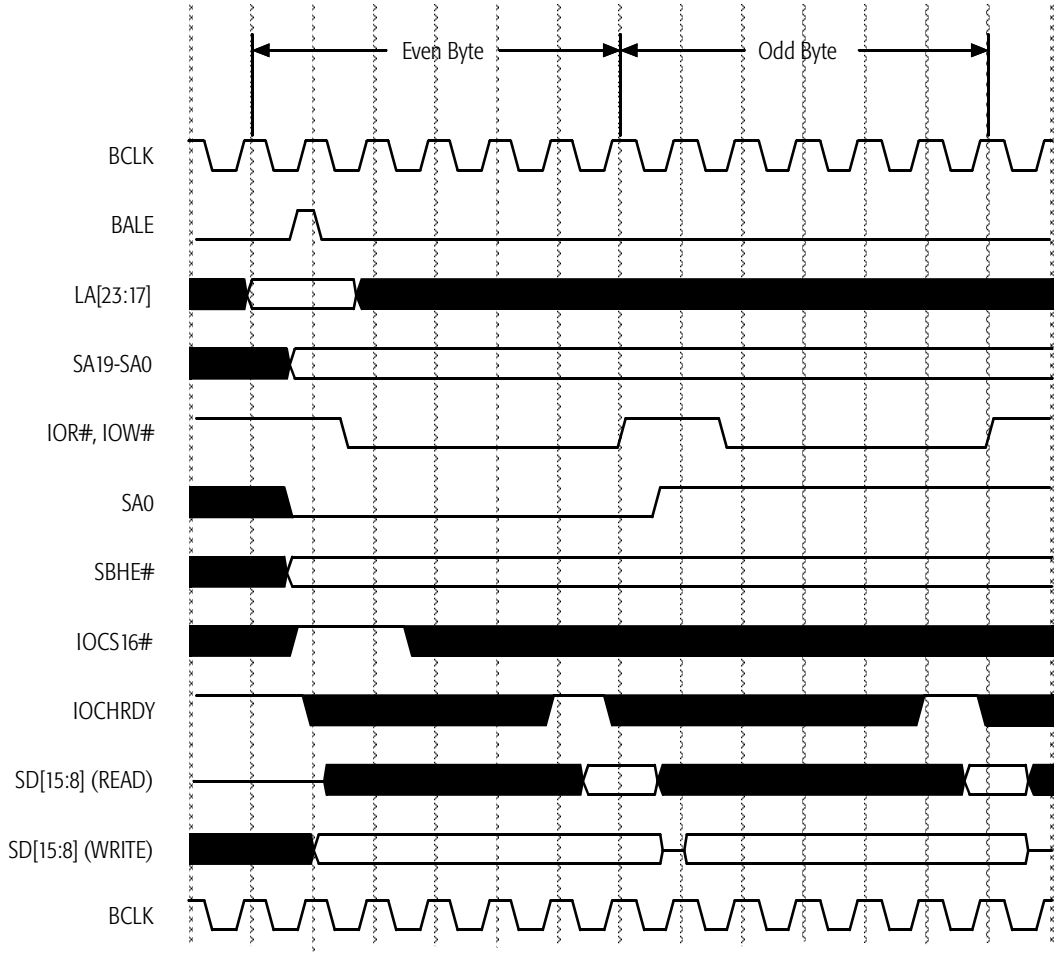


**Figure 5. I/O Cycle 16-Bit to 8-Bit Conversion**

The slot address lines SA1, SA0, and SBHE# function the same for I/O reads and writes as they do for memory reads and writes.

## 5.2.4 Memory Read and Write

The AMD-756 controller directs all memory accesses not claimed by other targets to the ISA bus. The AMD-756 steers data between the PCI AD bus and the ISA data bus as required by the requested cycle.

The AMD-756 supports bursting (multiple read or write transactions). If FRAME# and IRDY# are asserted at the same time, the AMD-756 will not disconnect if it is able to complete

the data phase within specified latency requirements. Target latency is limited to 16 PCI clocks from the assertion of FRAME# for initial accesses, and limited to eight PCI clocks from the end of the previous data phase for subsequent accesses of a burst cycle. All non-posted ISA writes and all ISA reads use delayed transactions to meet these latency requirements. Figure 6 shows the timing of a non-posted PCI cycle forwarded to the ISA bus.

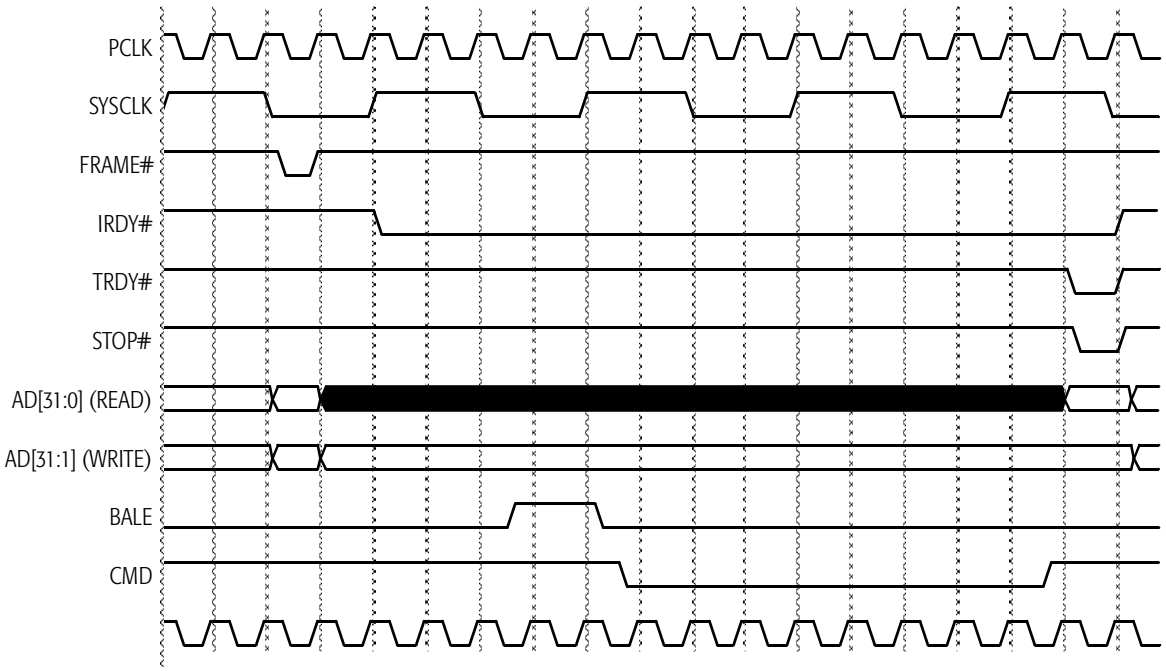


**Figure 6. Non-Posted PCI-to-ISA Access**

If the AMD-756 controller is unable to complete the initial data phase within the required initial latency, it begins a delayed transaction and terminates with retry by asserting STOP# but not asserting TRDY# at the end of the initial data phase. If the next data phase in a burst cannot be completed within the required incremental latency, the AMD-756 disconnects by asserting TRDY# and STOP# at the end of the current data phase.

Memory write posting in the AMD-756 controller is enabled by setting the Post Memory Write Enable bit, configuration register Function 0, offset 46h, bit 0 (see page 200). When write posting is enabled, TRDY# is asserted one clock cycle after both FRAME# and IRDY# are sampled asserted. The AMD-756

completes the access on the ISA bus. Attempts to access the ISA bus before the posted write is complete must wait for the ISA bus cycle to complete. The timing for a posted write cycle is shown in Figure 7.

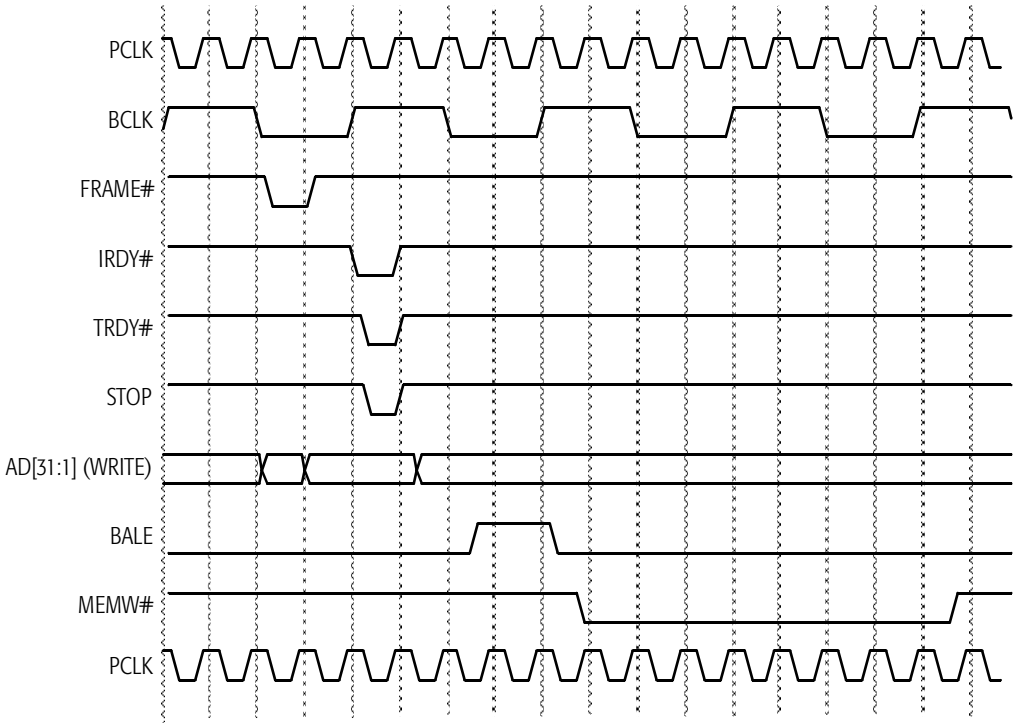


**Figure 7. Posted PCI-to-Memory Write**

The memory-related ISA bus control signals are MEMR#, SMEMR#, MEMW#, SMEMW#, and MEMCS16#. SMEMR# and SMEMW# are asserted only if the access is within the first Mbyte of memory. The state of MEMCS16# at the beginning of bus cycle state TC determines whether the present cycle is 8-bit or 16-bit, as shown in Figure 8 on page 69.

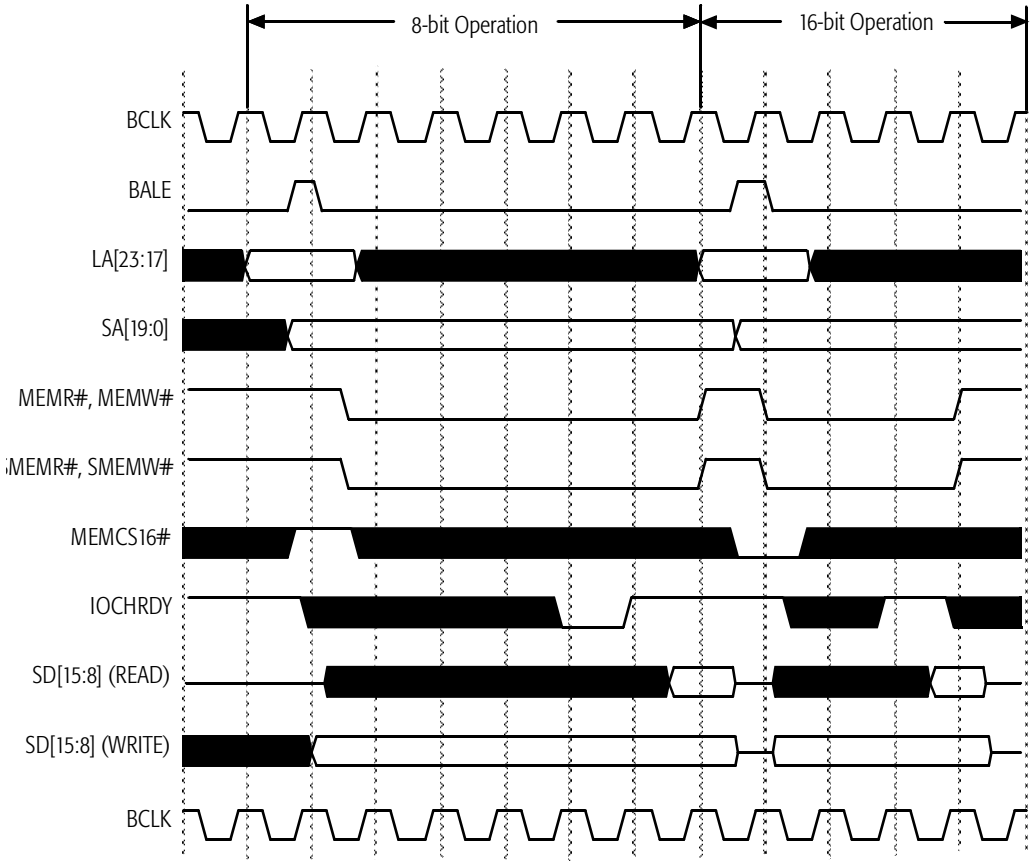


**Figure 8. ISA Bus Memory Access Cycle**

The command signals become asserted at the start of TC for 16-bit cycles, or in the middle of TC for 8-bit cycles. The falling edge of a command signal can be delayed by one or two BCLKs by setting bit 7 of the ISA Bus Control register, Function 0, offset 40h (see page page 198). To delay the rising edge of command signals by one BCLK set bit 5 in the ISA Bus Control register, Function 0, offset 40h. For slow peripherals, wait states may be inserted by negating IOCHRDY.

The AMD-756 converts a PCI bus master request for 16-bit, 24-bit, or 32-bit data from an 8-bit ISA memory into two, three, or four 8-bit cycles, respectively. A request for 32 bits from a 16-bit ISA target results in two 16-bit accesses. The AMD-756 also converts requests for non-contiguous bytes by converting the access to the appropriate ISA bus cycles. These conversion cycles are shown in Figures 9, 10, and 11.

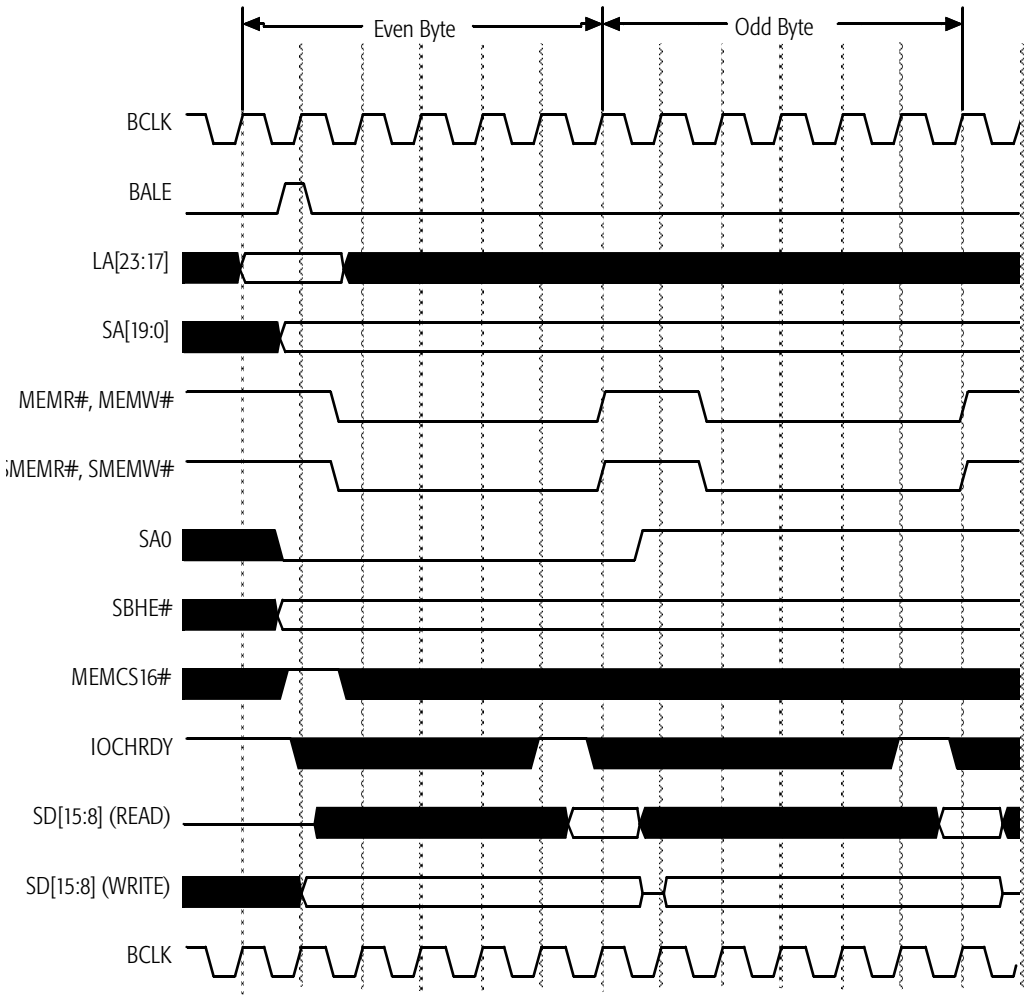


**Figure 9. ISA Bus Memory Cycle: 16-Bit to 8-Bit Conversion**

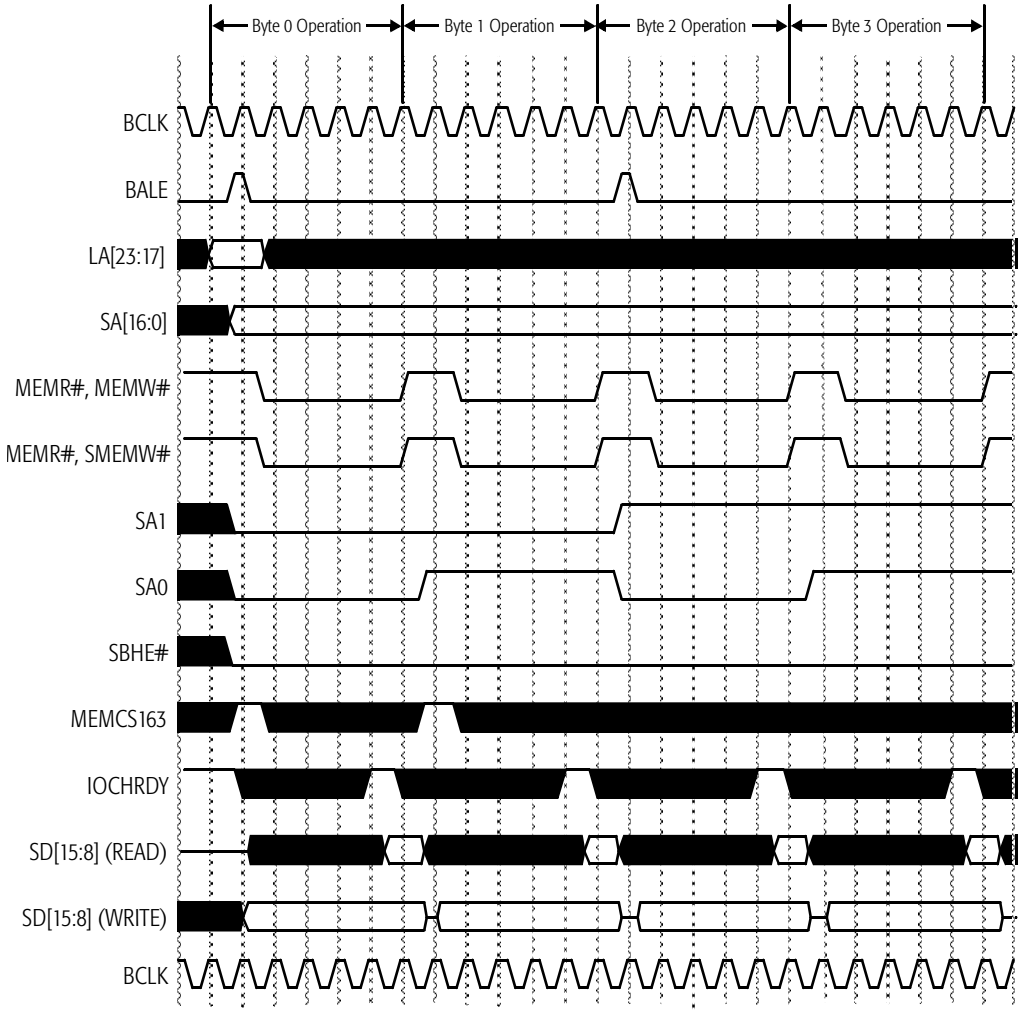


**Figure 10. Memory Cycle 32-Bit to 8-Bit Conversion**

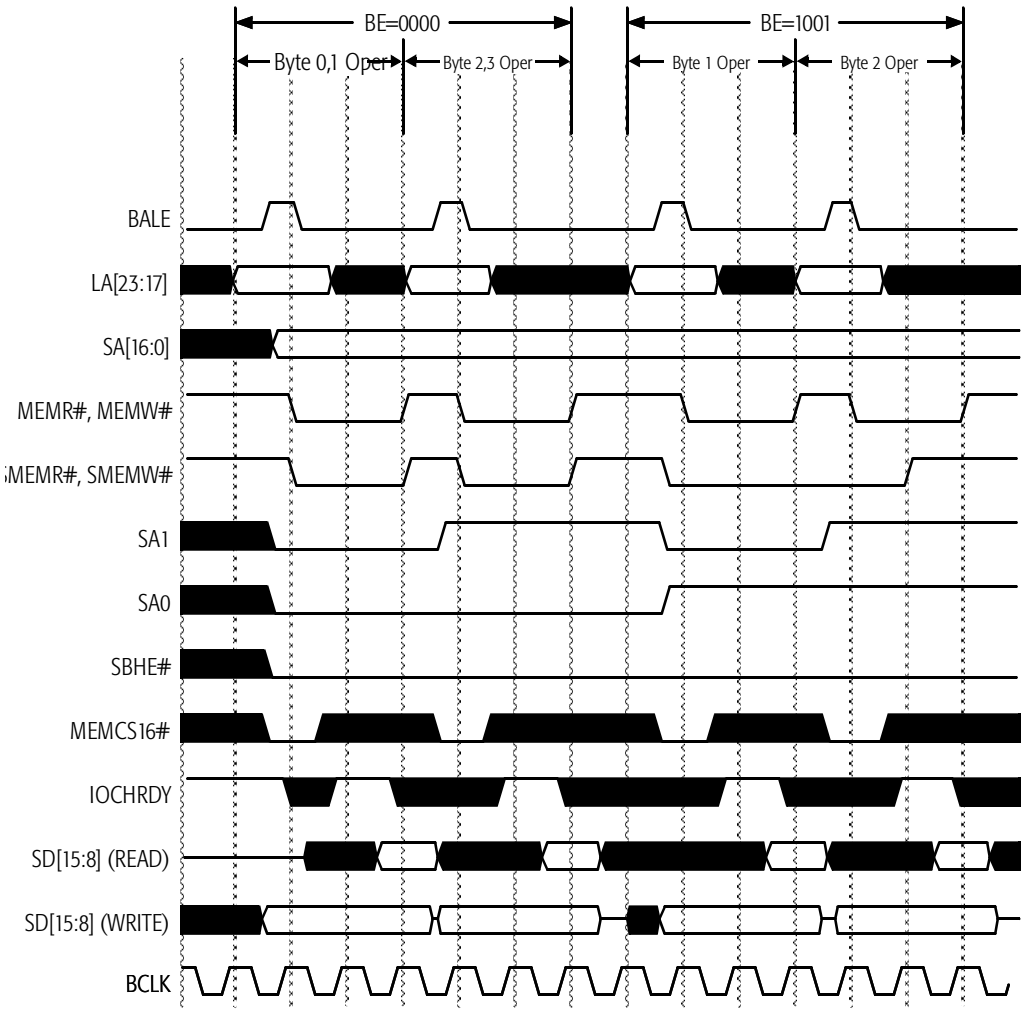


**Figure 11. Memory Cycle 32-Bit to 16-Bit Conversion**

If the memory accessed is ROM, the timing is different for command signals MEMR# and SMEMR#, which are asserted at the falling edge of BALE. Both 8-bit and 16-bit ROM access cycles are three wait states long. They can be programmed to be zero or one wait states using bit 1 of the ISA bus controller configuration register (see page 198). Figure 12 on page 73 shows a ROM access. Figure 13 on page 74 shows requests for 32 bits of data from 8-bit ROMs.

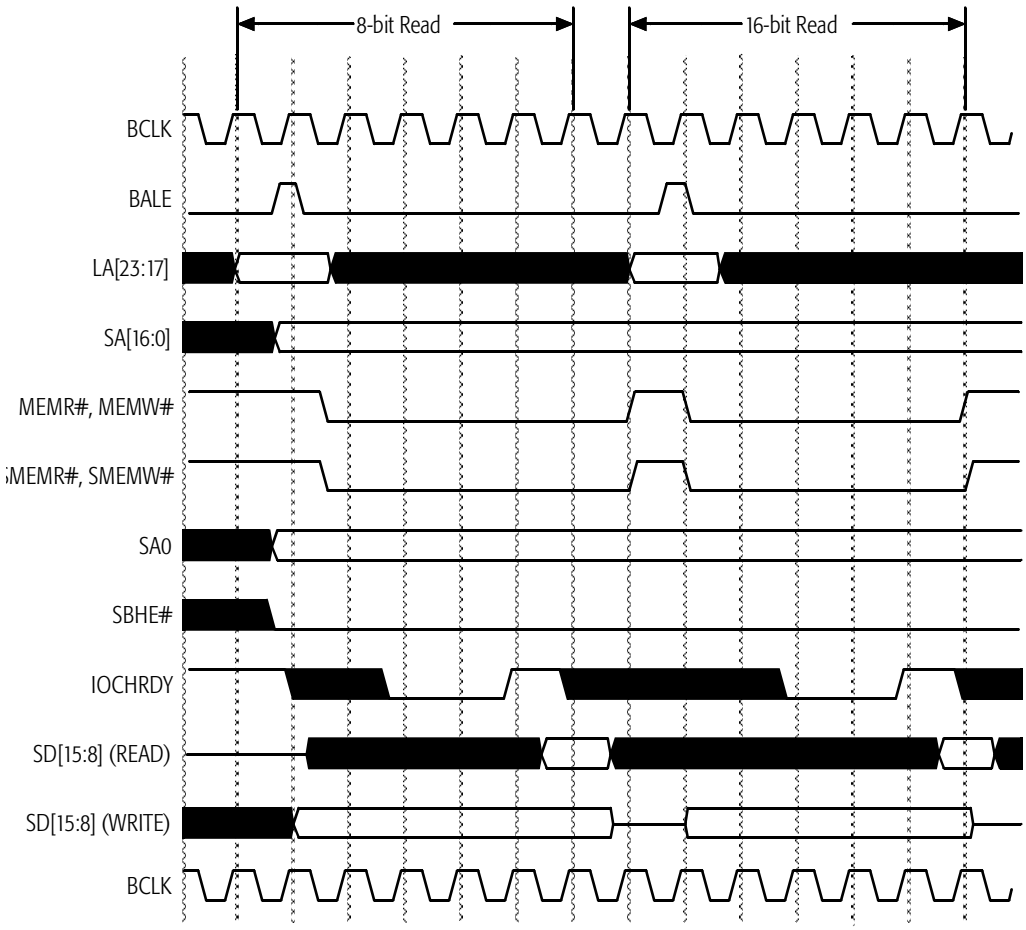


**Figure 12. ROM Access**

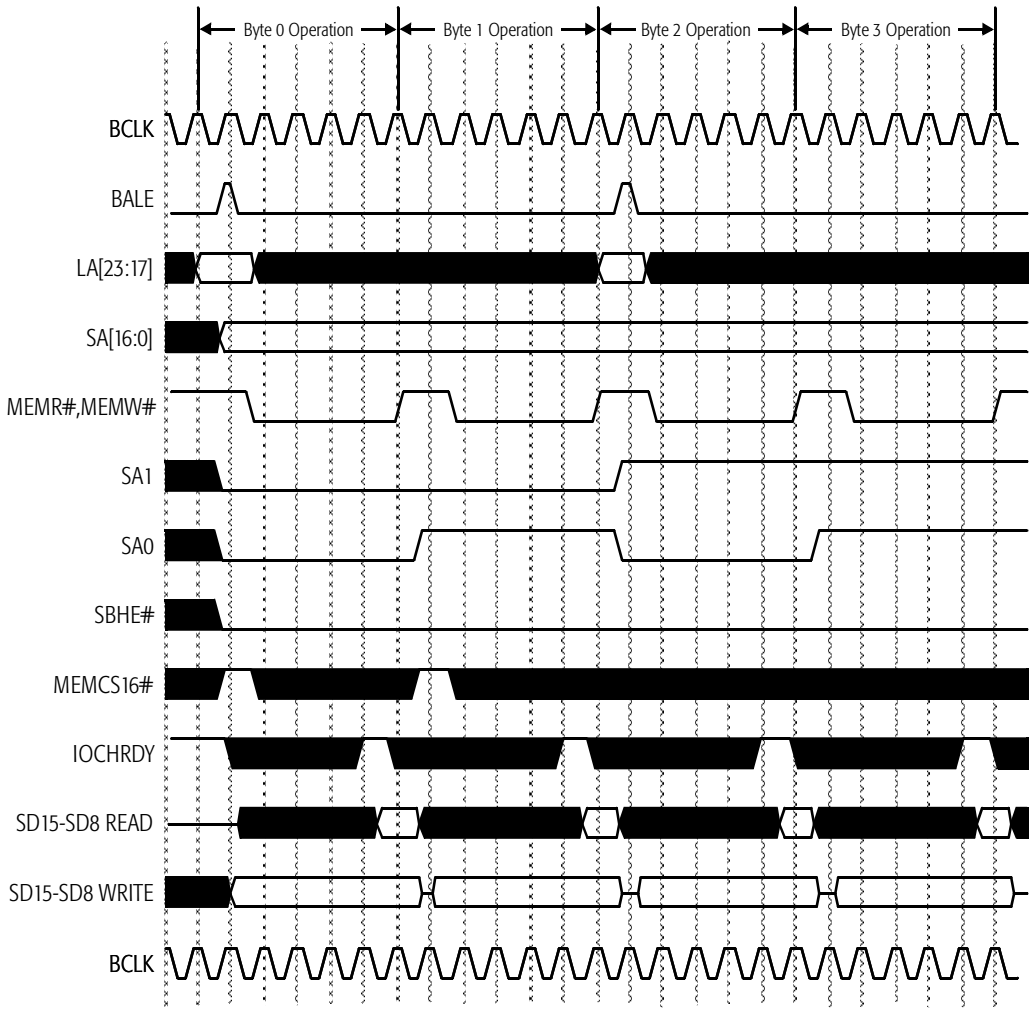


**Figure 13. ROM Cycle 32-Bit to 8-Bit Conversion**

SA1, SBHE#, and SA0 contain a direct decode of the C/BE[3:0]# inputs from the PCI bus. During a conversion cycle, SBHE# and SA0 are toggled so that the appropriate bytes are accessed, as shown in Table 3.

**Table 3. ISA Byte and Word Accesses**

| SBHE# | SA0 | Description |
|---|---|---|
| 0 | 0 | 16-Bit |
| 1 | 0 | 8-Bit, LSB |
| 0 | 1 | 8-Bit, MSB |
| 1 | 1 | undefined |

### 5.2.5 Configuration Read and Write

As a target, the AMD-756 controller responds to both read and write configuration cycles when device selection decoding is done externally via the IDSEL pin. The AMD-756 IDSEL connection is system-specific, but the recommended connection is to AD18.

If the AMD-756 is selected during a PCI master-initiated configuration cycle, DEVSEL# is asserted two clocks after FRAME# assertion. On PCI-to-configuration register reads, the AMD-756 controller drives the requested configuration register data onto AD[31:0], asserts TRDY# four clocks after FRAME# is asserted, and negates TRDY# and DEVSEL# one clock after IRDY# is asserted. On PCI-to-configuration register writes, the AMD-756 asserts TRDY# four clocks after FRAME# is asserted or two clocks after IRDY# is asserted, whichever is later. Data is strobed into the configuration registers the cycle before TRDY# is asserted.

The timing of these cycles is shown in Figure 14 and Figure 15.

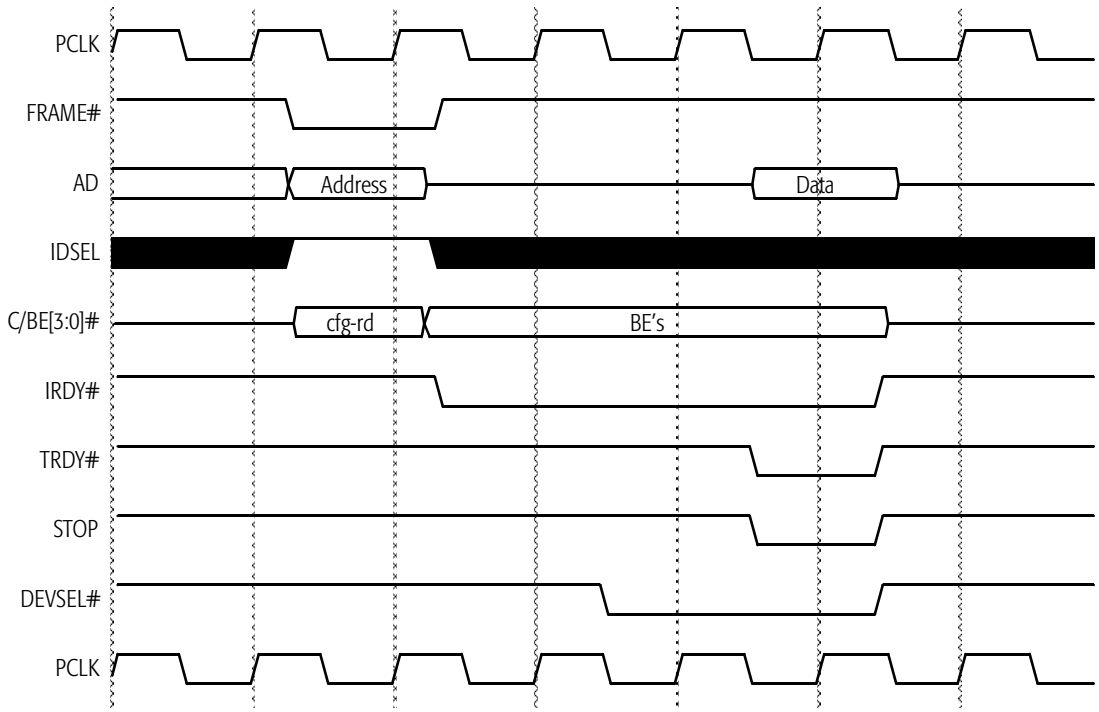


**Figure 14. Configuration Read Cycle**

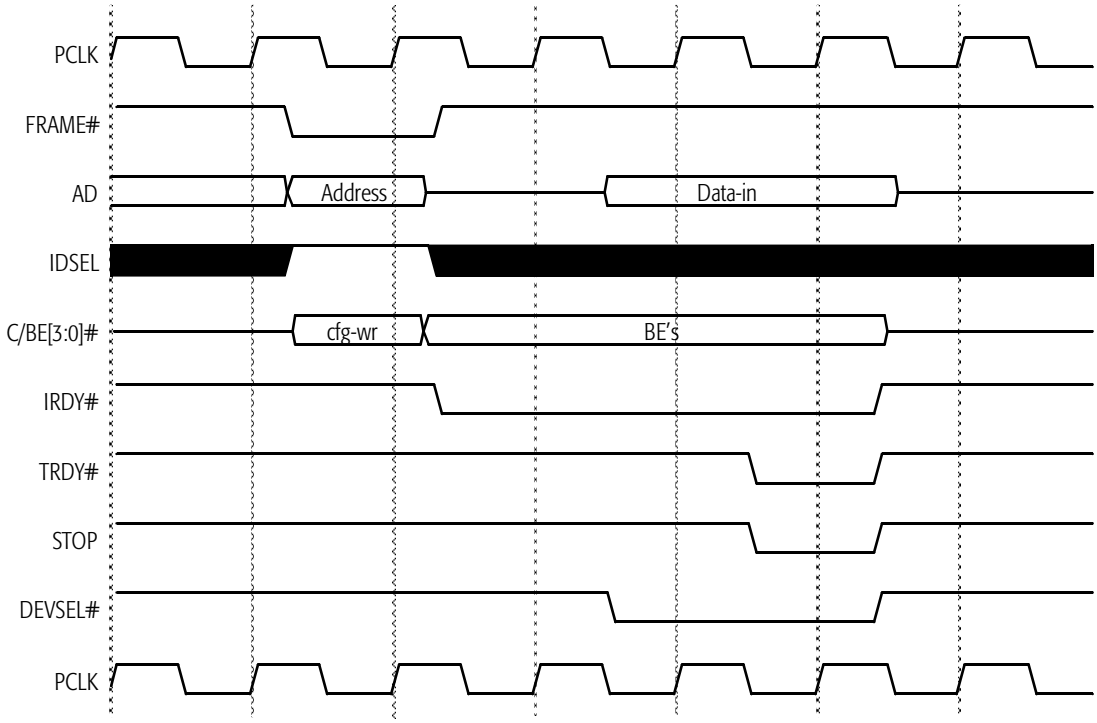


**Figure 15. Configuration Write Cycle**

## 5.2.6 Memory Read Multiple

The memory read multiple command is treated the same as a memory read command by the AMD-756 controller.

## 5.2.7 Dual Address Line

The AMD-756 supports 32-bit addressing only, so dual address line commands are ignored. There is no response.

## 5.2.8 Memory Read Line

The AMD-756 controller treats the memory read line command just as it does the memory read command.

## 5.2.9 Memory Write Invalidate

The AMD-756 treats the memory write invalidate command just as it does the memory write command.

# 5.3 PCI Bus Features

## 5.3.1 Back-to-Back Cycles

As a target, the AMD-756 controller can respond to fast back-to-back cycles as described in the PCI specification. All back-to-back cycles by the same initiator require at least one turn-around cycle, except when both transactions are writes to the same target.

## 5.3.2 Subtractive Decoding

Subtractive decoding ensures that every PCI bus access gets a response. Any PCI cycle not claimed by other targets and whose address is not defined in the AMD-756 controller address block is forwarded to the ISA bus. The timing for subtractive decoding is shown in Figure 16.

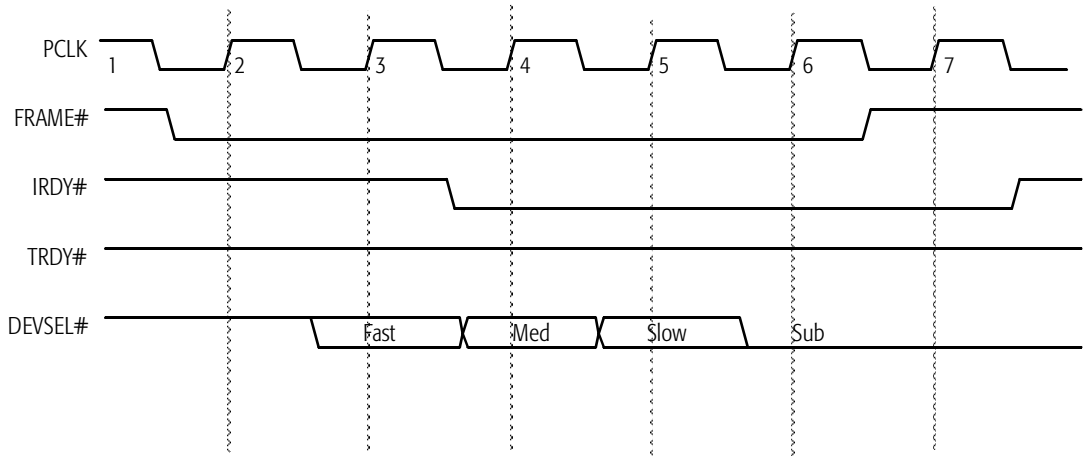


**Figure 16. Subtractive Decode Timing**

## 5.3.3 ISA Bus Control Register

Bus control options can be programmed via the ISA Bus Control register, Function 0, offset 40h (see page 198). This register controls the number of wait states to be inserted in the 8-bit and 16-bit slot cycles and determines the output drive of the slot bus buffers. More than five wait states are possible if IOCHRDY is pulled low before the last normal wait state.

# 5.4 ISA Bus-Initiated Cycles

The AMD-756 controller is responsible for forwarding ISA bus cycles to the PCI bus. The only two initiators on the ISA bus are the DMA controller and the ISA bus master. The DMA controller can only generate memory read and write cycles, while an ISA master can generate I/O as well as memory cycles.

Masters must repeat a read or write transaction that is terminated with retry. Masters must assert IRDY# within eight clocks during all data phases. Ideally, IRDY# is asserted with no delay on all data phases.

## 5.4.1 DMA-Initiated Cycles

In the PC/AT, DMA transfers occur between peripherals and memory with a data width of either 8 or 16 bits. Of the seven external DMA channels available, four are used for 8-bit transfers and three for 16-bit transfers. One byte or word is transferred in each DMA cycle.

Normally, an add-on card issues a DMA request by asserting one of the DRQ[7:5] or DRQ[3:0] signals. When the AMD-756 detects this request and the request is a read from memory, it generates a request to the PCI arbiter. When it receives a PCI grant, the AMD-756 controller initiates a PCI memory read transaction using the current DMA address, prefetching all data within the addressed doubleword. When the transaction is complete, the AMD-756 asserts the corresponding DACK# line to indicate a DMA acknowledge. Prefetch data is transferred in response to subsequent DMA requests without further PCI bus accesses.

When the AMD-756 controller detects a memory write request, it asserts the corresponding DACK# line to indicate the DMA acknowledge, reads the data from the DMA device, and merges the data into a single doubleword. When the last byte of the doubleword has been read, the AMD-756 generates a request to the PCI arbiter. When it receives a PCI grant, it starts a PCI memory write transaction for the entire doubleword with appropriate byte enables.

AEN and BALE are asserted after the DMA is acknowledged and any pending ISA bus cycle has completed. The DMA address is placed on LA[23:20] and SA[16:0]. Two DMACLK cycles later, either MEMR# and IOW# or MEMW# and IOR# are asserted, depending on the direction of the transfer. If the ISA Command Delay bit of the ISA Bus Control register is set, MEMR# is asserted one DMACLK cycle earlier. The command remains asserted for three DMACLK cycles. The data transfer takes place on the rising edges of command signals. TC is activated before the end of the command if the transfer is from one 8-bit device to another or one 16-bit device to another. If the transfer is from a 16-bit device to an 8-bit device, the command signals are again asserted after a delay of two DMACLK cycles and the transfer is complete. Figure 17 on page 80 shows the timing for a typical DMA transfer.

Due to concurrent PCI and ISA bus operation during DMA, the timing on each bus is independent of the state of the other bus. The state of the data buffers determines when PCI bus requests are generated and when DMA wait states are generated by negating IOCHRDY. PCI bus requests to the arbiter during memory reads are issued only when the memory read buffer is empty. During memory writes, PCI bus requests are issued when the MSB of the memory write buffer is full. IOCHRDY is negated when the memory read buffer is empty during memory reads, or when the memory write buffer is full during memory writes.

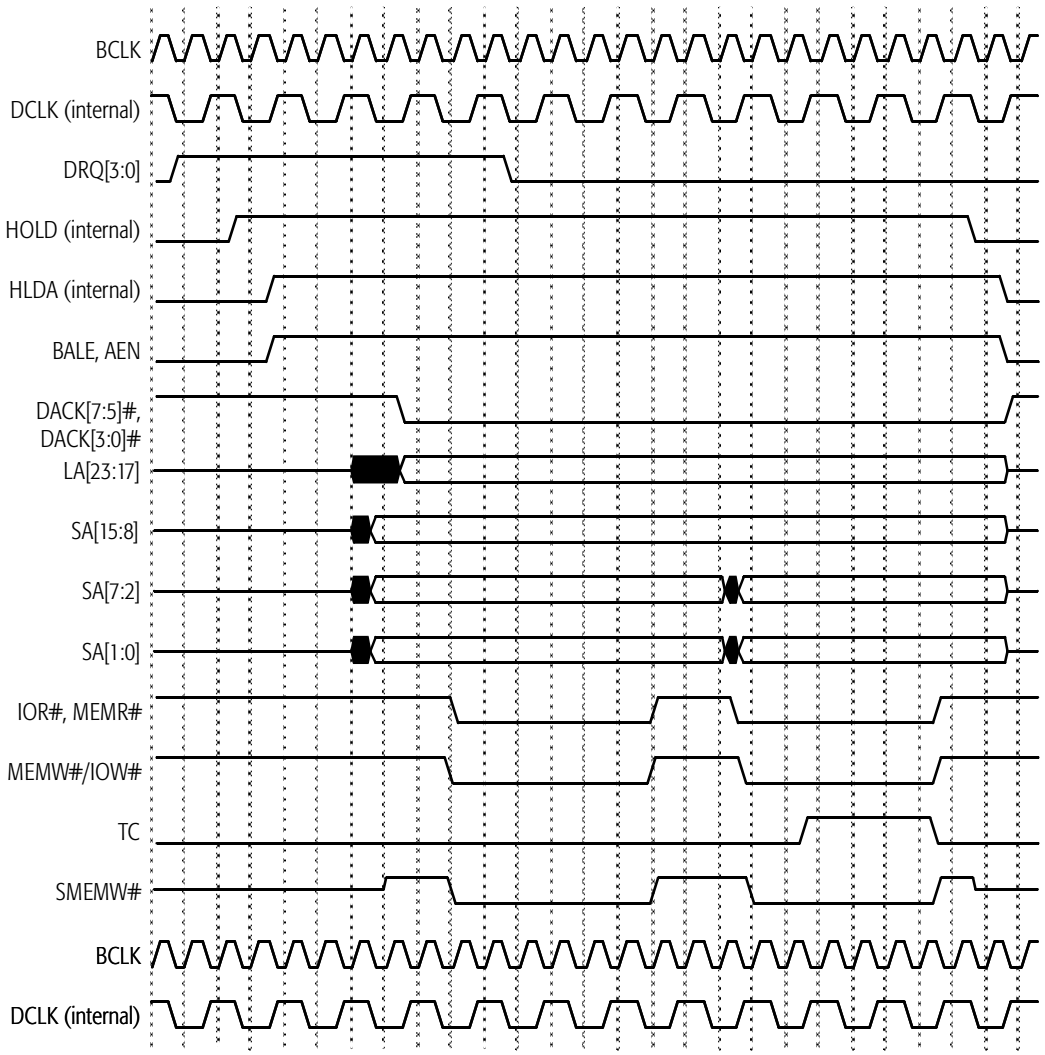


**Figure 17. DMA Transfer Cycle**

## 5.4.2 ISA Bus Master Initiated Cycles

An ISA bus master card issues a DMA request on the ISA bus, as shown in Figure 18 on page 81, using a DMA channel which has been placed in the cascade mode. The AMD-756 controller responds with an acknowledge signal in the same manner as for a DMA cycle. The add-on card then gains control of the ISA bus by asserting the MASTER# signal. Unlike DMA cycles, there can be multiple data transfers in master mode. An ISA bus master can generate both memory and I/O accesses.

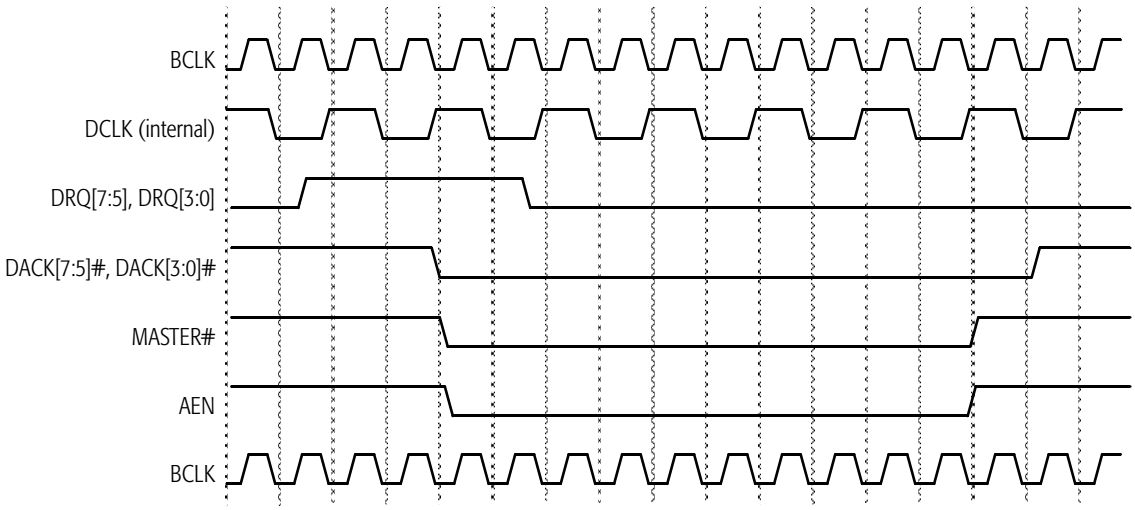


**Figure 18. ISA Bus Master Arbitration Timing**

When the AMD-756 controller detects MEMR# or MEMW# asserted, it starts the PCI cycle, asserts FRAME#, and negates IOCHRDY. This procedure guarantees that the ISA cycle will not complete before the PCI cycle has provided or accepted the data. IOCHRDY is asserted when IRDY# and TRDY# are sampled asserted. Figure 19 on page 82 shows an ISA bus master memory read, and Figure 20 on page 82 shows a ISA bus master memory write.

The ISA bus and PCI bus operate concurrently. A separate PCI bus request is issued for each ISA master command and the PCI bus ownership is relinquished after the transaction is completed. The AMD-756 converts ISA bus master I/O cycles into PCI I/O cycles. The timing of these cycles is similar to that of the memory cycles shown in Figure 17 on page 80 and Figure 18 on page 81, with the substitution of IOR# and IOW# for MEMR# and MEMW#.

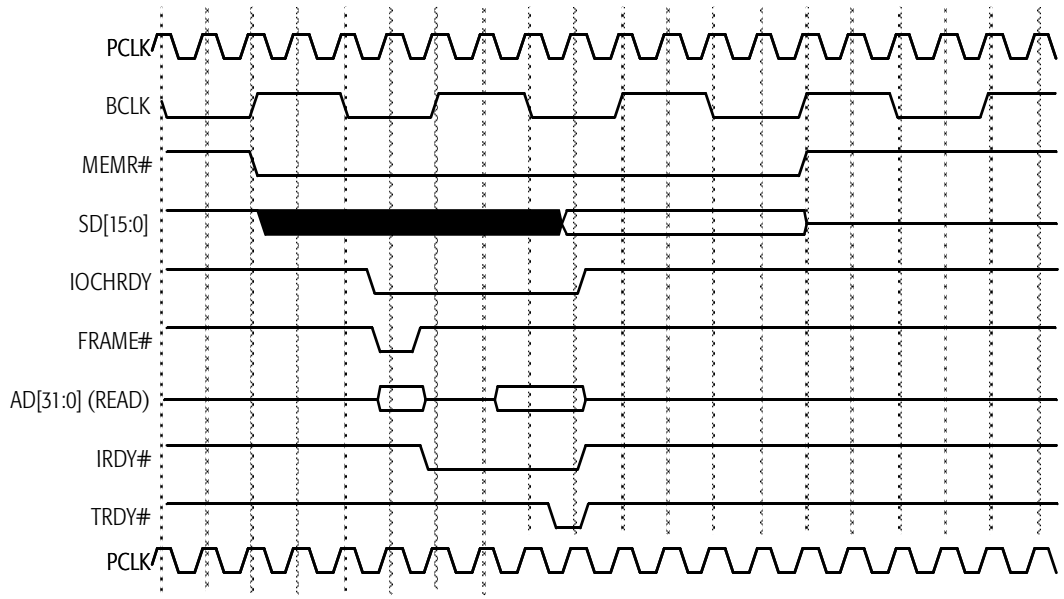


**Figure 19. ISA Bus Master-to-PCI Memory (Memory Read)**

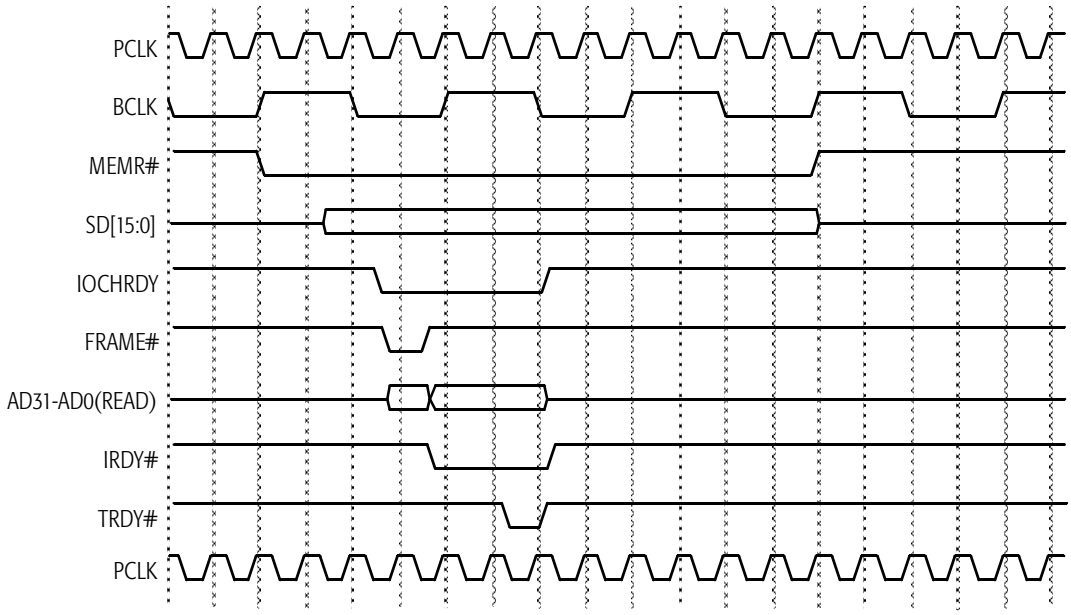


**Figure 20. ISA Bus Master-to-PCI Memory (Memory Write)**

## 5.5 PCI Bus Arbitration

The signals PREQ# and PGNT# are used to control requests for and grants of the PCI bus between the AMD-756 peripheral bus controller and the AMD-751 system controller.

## 5.6 I/O and Memory Mapping

The AMD-756 decodes PCI bus addresses to determine the destination of a PCI memory or I/O request. The AMD-756 address decoder distinguishes five regions for memory or I/O accesses. The regions are:

- IDE bus I/O
- Bus master IDE register I/O
- ISA bus I/O (on-chip)
- ISA bus I/O (off-chip)
- ISA bus off-board memory

The region selected is a function of the PCI address, the PCI cycle type, and the values placed in the configuration registers that control memory mapping. The regions are described below.

**IDE Bus I/O Location**

The AMD-756 controller generates an IDE bus access cycle via positive decoding and responds to the cycle when it recognizes an IDE target address.

**Bus Master IDE Register I/O Location**

An internal I/O access cycle is generated via positive decoding to the appropriate bus master IDE register I/O block, and is responded to by the AMD-756 when it recognizes a bus master IDE register target address. The base address of the bus master IDE registers is set by the configuration base registers and the size is fixed at 16 bytes (8 bytes for each channel).

**ISA Bus I/O Location (On-Chip)**

The AMD-756 controller generates an ISA bus I/O access cycle by subtractive decoding and is responded to by the AMD-756 when it recognizes an on-chip address during the ISA bus cycle.

**ISA Bus I/O Location (Off-Chip)**

The AMD-756 generates a standard ISA bus I/O access cycle by subtractive decoding when no other PCI target responds to a

PCI I/O cycle. Data is passed between the PCI data bus (AD[31:0]) and the ISA data bus (SD[15:0]). ROM_KBCS# is asserted to select the keyboard controller if the I/O address is port 60h or port 64h.

**ISA Bus Off-Board Memory Location**

Standard 8-bit or 16-bit ISA bus cycles are generated when the AMD-756 controller detects a memory access in the ISA slot bus address range. Data is passed between the PCI data bus (AD[31:0]) and the ISA data bus (SD[15:0]). The AMD-756 determines off-board memory locations through subtractive decoding of a PCI-to-ISA access (when none of the other targets asserts DEVSEL#). If the ISA address is defined as a ROM region, ROM_KBCS# is asserted.

## 5.6.1 I/O Mapping

I/O addresses not inhibited by DEVSEL# are run as ISA bus cycles. The data steering is based on the actual I/O addresses, depending on whether the I/O location is on-chip or off-chip.

**On-Chip I/O**

For on-chip centralized and distributed DMA devices, the ISA bus cycle is run normally. Only the steering on read cycles is affected. ISA bus masters have access to all on-chip registers. The centralized DMA I/O locations are at a fixed address, as shown in Table 4, while the distributed DMA I/O locations are at programmable base addresses.

**Table 4. I/O Fixed Address Mapping**

| Address | Device | Location |
|---|---|---|
| 0000h–000Fh | DMA#1 | On-chip or PCI bus |
| 0080h–008Fh | DMA page registers | On-chip |
| 00C0h–00DFh | DMA#2 | On-chip or PCI bus |
| 0170h–0177h | IDE channel 2 | IDE bus |
| 01F0h–01F7h | IDE channel 1 | IDE bus |
| 0376h | IDE channel 2 | IDE bus |
| 03F6h | IDE channel 1 | IDE bus |
| 0010h–007Fh, 0090h–00BFh, 00E0h–016Fh, 0178h–01EFh, 01F8h–0375h, 0378h–03F5h, 03F8h–FFFFh | General I/O Locations | PCI/ISA bus |

**SA Bus I/O**

All I/O write cycles drive the data from the AD bus onto the SD bus and generate an IOW# strobe. All I/O read cycles drive data from the SD bus onto the AD bus and generate an IOR# strobe. The AMD-756 controller drives data onto the SD bus during all on-chip reads, while the SD bus is the data source for all other I/O reads.

## 5.6.2 Memory Mapping

Memory accesses are divided into PCI memory, ROM, and ISA bus memory accesses. Table 5 shows the various memory regions and the destinations (PCI, ROM, or ISA) supported by the AMD-756.

**Table 5. Memory Address Mapping**

| Range | Address | Destination | Comments |
|---|---|---|---|
| 0 to 786 Kbytes | 0_0000h–B_FFFFh | PCI bus space<br>ISA bus space | Selected by asserted DEVSEL#<br>(By subtractive decode) |
| 786Kbytes to 960Kbytes | C_0000h–E_FFFFh | PCI bus space<br>ISA bus space<br>ISA ROM space | Selected by asserted DEVSEL#<br>(By subtractive decode) or<br>selected by ROM decode control |
| 960Kbytes to 1 Mbyte | F_0000h–F_FFFFh | ISA bus space<br>ISA ROM space | (By subtractive decode) |
| 1 Mbyte to<br>15.875 Mbytes | 10_0000h–FD_FFFFh | PCI bus space<br>ISA bus space | Selected by asserted DEVSEL#<br>(By subtractive decode) |
| 15.875 Mbytes to<br>16 Mbytes | FE_0000h–FF_FFFFh | PCI bus space<br>ISA bus space | Selected by asserted DEVSEL#<br>(By subtractive decode) |
| 16 Mbytes t0 128 Mbytes | 100_0000h–7FF_FFFFh | PCI bus space<br>Aliased ISA<br>bus space | Selected by asserted DEVSEL#<br>(By subtractive decode) |
| 128 Mbytes to<br>(4 Gbytes – 512 Kbytes) | 8000_0000h–FFF7_FFFFh | PCI bus space<br>Aliased ISA<br>bus space | Selected by asserted DEVSEL#<br>By subtractive decode only |
| (4Gbytes – 512Kbytes)<br>to 4 Gbytes | FFF8_0000h–FFFF_FFFFh | ISA ROM space | (By subtractive decode) or<br>selected by ROM decode control |

When a PCI memory access is generated, one of the following events will occur.

- If the DEVSEL# input is sampled asserted within the fast, medium, or slow sample periods, the AMD-756 controller is deselected and a PCI target device completes the cycle.

- If the DEVSEL# input is not sampled asserted within the fast, medium, or slow sample periods, the AMD-756 executes a subtractive decode which directs the access to the ISA bus.

When a master mode or DMA ISA memory access is generated, the AMD-756 controller initiates a PCI cycle. If DEVSEL# is not asserted within the fast, medium, or slow sample periods, the AMD-756 executes a subtractive decode which directs the access to the ISA bus, and IOCHRDY is re-asserted to allow the ISA cycle to complete.

**ISA Memory**

All memory accesses at or below FF_FFFFh (16 Mbytes) that are not accepted by PCI bus devices through the assertion of DEVSEL# are directed to the ISA bus. The AMD-756 controller asserts DEVSEL# for the cycles and generates standard ISA cycles. It also provides the data latching and steering logic to allow the PCI initiator to perform 8-bit, 16-bit, 24-bit, or 32-bit accesses to either 8-bit or 16-bit ISA memory devices.

Accesses to the PCI bus performed subtractively above FF_FFFFh (16 Mbytes) are aliased to the 24-bit ISA bus addresses. PCI accesses to these regions are performed only if no DMA or master mode cycles ever access the referenced locations, because a slot bus memory device might occupy the same aliased address as a PCI bus memory device, causing bus contention.

Access to system ROM is provided in the top 512 Kbytes of the aliased ISA bus address space for correct reset vectoring.

### 5.6.3 System ROM Memory Mapping

Setting bits in the ROM decode control enable different address ranges to be included in the ROMCS# decode. PCI accesses to the highest 512 Kbytes of each 16 Mbyte memory space (XXF8_0000h to XXFF_FFFFh) are system ROM accesses. System ROM accesses are a subset of ISA bus accesses, and are generated as standard ISA bus accesses, except that:

- ROMCS# is always asserted
- Additional ISA bus wait states can be programmed via the ROM wait states bit of the ISA bus control register.

The AMD-756 controller provides data latching and steering logic to allow the initiators to perform 8-bit, 16-bit, 24-bit, or 32-bit accesses to 8-bit system ROMs. It also performs the required ISA bus cycles to assemble and latch the appropriate data and to present it to the PCI initiator as requested. System ROM is also accessible by ISA bus masters and DMA cycles.

Video ROM and fixed disk ROM, memory range C0000h to CFFFFh, can be defined to be in the system ROM range using bits 7–0 of the ROM Decode Control register (Function 0, offset 43h). The programmable values of these bits are shown in Table 6. Setting the indicated bit enables the address range shown to be included in the ROMCS# decode.

**Table 6. ROM Decode Control Register**

| Bit Value | Address Range Enabled |
|---|---|
| Bit 7 = 1 | FFFE0000h–FFFEFFFFh Enabled |
| Bit 6 = 1 | FFF80000h–FFFDFFFFh Enabled |
| Bit 5 = 1 | 000E8000h–000EFFFFh Enabled |
| Bit 4 = 1 | 000E0000h–000E7FFFh Enabled |
| Bit 3 = 1 | 000D8000h–000DFFFFh Enabled |
| Bit 2 = 1 | 000D0000h–000D7FFFh Enabled |
| Bit 1 = 1 | 000C8000h–000CFFFFh Enabled |
| Bit 0 = 1 | 000C0000h–000C7FFFh Enabled |

Subtractive decodes are always performed, and the ROM access may be inhibited by a PCI target that is asserting DEVSEL# and claiming the cycle.

### Flash Memory Support

Support for programmable flash memory is provided by enabling write cycles to the BIOS ROM regions. Bit 0 of the ISA Bus Control register (Function 0 offset 40h) is provided to enable write cycle generation.

# 5.7 Power Planes and Reset

The AMD-756 controller has six power planes listed in Table 7.

**Table 7. AMD-756™ Peripheral Bus Controller Power Planes**

| Name | # of Pins | Description |
|---|---|---|
| VDD3 | 11 | Main source of 3.3 volts in the IC core and I/O cells. This plane is powered down when the PWRON# output (to the power supply) is in the negated state. This plane powers all pins not powered by VDD_SOFT, VDD_USB, and VDD_RTC. |
| VDD_SOFT | 1 | Sleep-mode 3.3-volt power. When most of the system is powered down (PWRON# is in the negated state), this plain remains powered. This plane is used to detect wake events. The pins powered by this plane are PWRBTN#, PWRON#, PME#, SMBUS[C,D], EXTSMI#, SLPBTN#, RI#, and PWRGD. |
| VDD_REF | 1 | Reference voltage for 5-volt tolerant I/O cells. In 3.3-volt system (where 5-volt tolerance is not necessary), this plane should be tied to the 3.3-volt supply. This pin should be connected to the 5-volt output of the power supply that is asserted in the SOFF power state (soft off)—the 5-volt version of VDD_SOFT from the power supply enters this pin and is regulated down to 3.3 volts to drive the VDD_SOFT pin. |
| VDD_USB and GND_USB | 1 each | USB differential output power source. These are used to power the USB transceivers. The pins powered by this plane are USBP[3:0] and USBN[3:0]. |
| VDD_RTC | 1 | Battery power for the real-time clock. This plane is normally always powered. The pins powered by this plane are RTCX_IN and RTCX_OUT. |

In addition to these six power planes, there is a single ground plane called GND.

*Note: The operational tolerance on VDD3, VDD_SOFT, VDD_USB, and VDD_RTC is from 3.0 to 3.6 volts.*

# 5.8 Clock Generation

The clocks described in the following paragraphs are used or generated by the AMD-756.

**PCLK** This input signal is the PCI clock used to synchronize the interface to all PCI bus devices. PCLK can operates frequencies up to 33.333Mhz.

**OSC** This input signal is a 14.318-MHz clock common to the ISA bus signal OSC and is used by the programmable interval timer (PIT), by some of the power management logic, and to optionally create BCLK.

**USBCLK** This input 48-Mhz clock is used by the USB controller.

**KBCK** This clock is the output clock to the keyboard when the internal keyboard controller is selected

**MSK** This clock is the output clock to the mouse when the internal keyboard controller is selected.

**RTC_XIN, RTC_XOUT** The clock generated by this oscillator is a 32.768-KHz oscillator pair and is used by the real-time clock. This clock is powered by the VDD_RTC power plane.

**PICCLK** This clock is the APIC interrupt message bus clock. This is used as the clock for the programmable frequency clock for interrupt message bus. The frequency is specified by Function 0, offset 4B bits [APICCKS].

**BCLK** This output signal is the ISA bus system clock. It is derived either by a division of PCLK by 2, 3, 4, 5, 6, 10, or 12, or by a division of OSC by 2. BCLK timing is controlled by programming the ISA Clock Control register, Function 0, offset 42h. Bit 3 of this register, the ISA Clock Select Enable bit, is cleared at reset, forcing BCLK to default to a value of = PCLK/4.

To program a different time value for BCLK, take the following steps.

1. Clear bit 3 of the ISA Clock Control register.
2. Program bits 2–0, the ISA Bus Clock Select bits of this register, writing the value selected from Table 8.
3. Set bit 3 of the ISA Clock Control register.

**Table 8. ISA Bus Clock Select Bit Programming**

| Bit 2 | Bit 1 | Bit 0 | BCLK Value |
|---|---|---|---|
| 0 | 0 | 0 | PCLK / 3 (default) |
| 0 | 0 | 1 | PCLK / 2 |
| 0 | 1 | 0 | PCLK / 4 |
| 0 | 1 | 1 | PCLK / 6 |
| 1 | 0 | 0 | PCLK / 5 |
| 1 | 0 | 1 | PCLK / 10 |
| 1 | 1 | 0 | PCLK / 12 |
| 1 | 1 | 1 | OSC / 2 |

# 5.9 Resets

PWRGD (power good) from the power supply is the main source of reset for the AMD-756 VDD3 logic. From this signal, PCIRST#, RSTDRV, and CPURST are derived. The normal power-up sequence is shown in Figure 21.

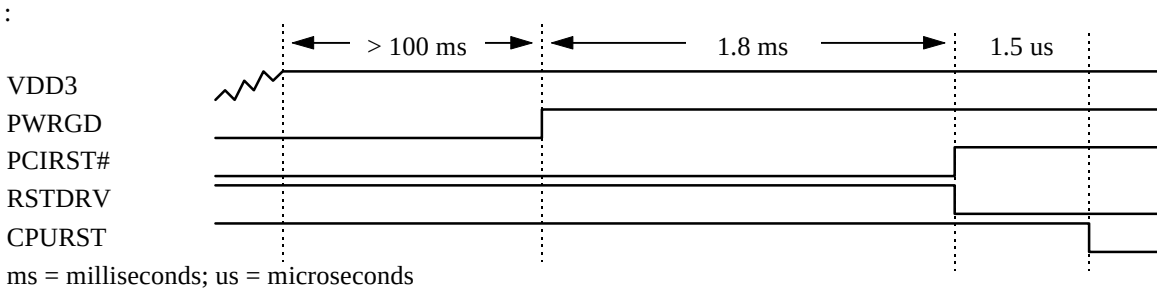


**Figure 21. Normal Power-Up Sequence**

A 1.8-msec delay occurs between the start of PWRGD and the end of PCIRST# and RSTDRV. This delay can be shortened (for simulation and test purposes) by forcing the IOCHRDY pin low while the PWRGD signal transitions from low to high. If this procedure is followed, the delay from PWRGD to PCIRST#/RSTDRV is reduced to approximately 1.5 microseconds.

The system can be reset by writing a 1 to function 0 offset 47h bit [SWPCIR]. This resets most of the internal registers and generates a pulse on PCIRST#, RSTDRV, and CPURST. The pulse generated on PCIRST# and RSTDRV is four PCLK cycles long. At the end of this pulse, CPURST remains asserted for three additional PCLK cycles.

The system can also be reset via the keyboard controller, PORT92 (system control register, fixed-address I/O port 92h), and a shutdown command. These all generate pulses that are 1.0 to 1.5 microseconds long. If function 0 offset 47 bit [CPURS] is high, these reset commands generate pulses on the INIT# pins rather than CPURST.

# 5.10 Direct Memory Access

The DMA controllers are 8237-compatible, have internal latches for latching the middle address bits output by the 8237 megacells on the data bus, and have 74LS612 equivalent memory mappers to generate the upper address bits.

The DMA logic controls transfers between an I/O channel and on-board or off-board memory. This logic generates a bus request to the PCI bus when an I/O channel requests a DMA operation. Once a bus grant has been issued, and any pending access to the ISA bus is completed, the DMA controller drives the PCI address bus and the slot address bus. DMA transfers can occur over the full 16 Mbyte range available on the slot bus and the entire 32-bit address range of the PCI bus.

## 5.10.1 DMA Controllers

The AMD-756 controller supports seven DMA channels using two 8237 equivalent megacells capable of running at BCLK. This option is programmable via the Type F DMA Control register (Function 0, offset 45h). DMA controller 1 contains channels 0 through 3. These channels support 8-bit I/O adapters. They are used to transfer data between 8-bit peripherals and 8-bit or 16-bit memory. Each channel can transfer data in 64-Kbyte pages within the first 16 Mbytes of the PCI memory space.

DMA controller 2 contains channels 4 through 7. Channel 4 is used to cascade DMA controller 1, so it is not available externally. Channels 5 through 7 support 16-bit I/O adapters to transfer data between these adapters and 16-bit system memory. Each channel can transfer data in 128-Kbyte pages within the first 16 Mbytes of the PCI memory space. Channels 5, 6, and 7 are meant to transfer 16-bit words only and cannot address odd bytes in system memory.

## 5.10.2 DMA Controller Registers

The 8237 megacells can be programmed anytime PGNT# is negated, i.e., when DMA controllers are not in operation. Table 9 on page 92 lists the I/O addresses of all target and master

DMA controller registers that can be read or written in the 8237 megacells. Channels 0–3 of the master and target DMA Controllers control system DMA Channels 0–3. There are 16 master and target DMA controller registers.

### Target and Master DMA Controllers Ports C0h–DFh

The target and master DMA controller ports are in Table 9. When writing to a channel address or word count register, the data is written into both the base register and current register simultaneously. When reading a channel address or word count register, only the current address or word count can be read. The base address and base word count are not accessible for reading.

The address and word count registers for each channel are 16-bit registers. The value on the data bus is written into the upper byte or lower byte, depending on the state of the internal addressing flip-flop. This flip-flop can be cleared by the Clear Byte Pointer Flip-Flop command. Following this command, the first read/write to an address or word count register will read or write to the least significant byte of the 16-bit register and the byte pointer flip-flop toggles back to zero.

**Table 9. Ports 00h–0Fh Master DMA Controller**

| Target I/O Address Bits | Master I/O Address Bits | Register Name | Access |
|---|---|---|---|
| 0000 0000 1100 000x | 0000 0000 000x 0000 | Ch 0 Base/Current Address | RW |
| 0000 0000 1100 001x | 0000 0000 000x 0001 | Ch 0 Base/Current Count | RW |
| 0000 0000 1100 010x | 0000 0000 000x 0010 | Ch 1 Base/Current Address | RW |
| 0000 0000 1100 011x | 0000 0000 000x 0011 | Ch 1 Base/Current Count | RW |
| 0000 0000 1100 100x | 0000 0000 000x 0100 | Ch 2 Base/Current Address | RW |
| 0000 0000 1100 101x | 0000 0000 000x 0101 | Ch 2 Base/Current Count | RW |
| 0000 0000 1100 110x | 0000 0000 000x 0110 | Ch 3 Base/Current Address | RW |
| 0000 0000 1100 111x | 0000 0000 000x 0111 | Ch 3 Base/Current Count | RW |
| 0000 0000 1101 000x | 0000 0000 000x 1000 | Status/Command | RW |
| 0000 0000 1101 001x | 0000 0000 000x 1001 | Write Request | WO |
| 0000 0000 1101 010x | 0000 0000 000x 1010 | Write Single Mask | WO |
| 0000 0000 1101 011x | 0000 0000 000x 1011 | Write Mode | WO |
| 0000 0000 1101 100x | 0000 0000 000x 1100 | Clear Byte Pointer F/F | WO |
| 0000 0000 1101 101x | 0000 0000 000x 1101 | Master Clear | WO |
| 0000 0000 1101 110x | 0000 0000 000x 1110 | Clear Mask | WO |
| 0000 0000 1101 111x | 0000 0000 000x 1111 | R/W All Mask Bits | RW |

*Note: Not all address bits are decoded.*

The 8237 DMA controller megacells allow the user to program the asserted level of the DREQ and DACK# signals to be active low or high. Because the two megacells are cascaded together internally on the chip, DREQ must always be programmed active high and DACK# active low.

When programming the 16-bit channels (DMA controller 2, channels 5, 6, and 7), the address written to the base register must be the real address divided by two. The base word count for these channels is the number of 16-bit words to be transferred, not the number of bytes, as is the case for the 8-bit channels (DMA controller 1, channels 0, 1, 2, and 3). It is recommended that all internal locations in the 8237 megacells, especially the mode registers, must be loaded with some valid value, even if the channels are not used.

### 5.10.3 Middle Address Bit Latches

The middle DMA address bits are held in an internal 8-bit register. The DMA controller drives the value to be loaded onto the internal data bus, then issues an address strobe signal to latch the data bus value into this register. An address strobe is issued at the beginning of a DMA cycle and any time the lower 8-bit address increments across the 8-bit subpage boundary during block transfers. This register cannot be read or written to externally. It is loaded only from the address strobe signals from the megacells, and the outputs go only to the AD[16:8] pins.

### 5.10.4 Page Registers

The AMD-756 controller uses two 74LS612 cells to generate the page registers for each DMA channel. The page registers provide the upper address bits during DMA cycles. DMA addresses do not increment or decrement across page boundaries. Page boundaries for the 8-bit channels (channels 0, 1, 2, and 3) are every 64 Kbytes. Page boundaries for the 16-bit channels (channels 5, 6, and 7) are every 128 Kbytes. There are 32 8-bit registers between the 612 megacells.

Page registers must be written at the I/O addresses shown in Table 10 to select the correct page for each DMA channel before any DMA operations are performed. Address locations

between 080h and 08Fh other than those shown in the table are not used by the DMA channels, but can be read or written to by a PCI bus master.

**Table 10. Ports 80h–8Fh DMA Page Register Access**

| Page Register Address | DMA Channel | I/O Address Bits 15–0 | Register Name | |
|---|---|---|---|---|
| 87h | 0 | 0000 0000 1000 0111 | Ch 0 DMA Page M[0] | RW |
| 83h | 1 | 0000 0000 1000 0011 | Ch 1 DMA Page M[1] | RW |
| 81h | 2 | 0000 0000 1000 0001 | Ch 2 DMA Page M[2] | RW |
| 82h | 3 | 0000 0000 1000 1101 | Ch 3 DMA Page M[3] | RW |
| 8Bh | 5 | 0000 0000 1000 1111 | Ch 5 DMA Page M[5] | RW |
| 89h | 6 | 0000 0000 1000 1011 | Ch 6 DMA Page M[6] | RW |
| 8Ah | 7 | 0000 0000 1000 1001 | Ch 7 DMA Page M[7] | RW |
| 8Fh | 4 | 0000 0000 1000 1010 | Ch 4 DMA Page M[4] | RW |

The page register is used to set the values for AD[23:16] bus lines. In normal operation, zeroes are driven onto PCI address bits AD[31:24] during DMA cycles, making the AMD-756 backward-compatible with the PC/AT standard.

### 5.10.5 DMA Address Generation

DMA addresses are organized as upper, middle, and lower address portions.

The upper address portion selects a specific page, and is generated by the page registers in the 74LS612 megacells. The page registers for each channel must be set up by the system before a DMA operation. DMA addresses do not increment or decrement across page boundaries. Page sizes are 64 Kbytes for 8-bit channels 0 through 3, and 128 Kbytes for 16-bit channels 5 through 7. The DMA page register values are output on PCI address bus AD[31:16] (8-bit channels) and AD[31:17] (16-bit channels).

The middle address portion, which selects a block within the page, is generated by the 8237 megacells at the beginning of a DMA operation and any time the DMA address increments or decrements through a block boundary. The block size of an 8-bit channel is 256 bytes, while that of a 16-bit channel is 512 bytes. The middle address portion is output by the 8237 megacells onto the internal data bus were it is latched. The

latched address are then driven on PCI address bits AD[15:8] for 8-bit channels and AD[16:9] for 16-bit channels.

The lower address portion is generated directly by the 8237 megacells during DMA operations, and the lower address bits are output on PCI address bits AD[7:0] for 8-bit channels and AD[8:1] for 16-bit channels.

SBHE# is configured as an output during all DMA operations It is driven as the inversion of AD0 during 8-bit cycles, and forced low for all 16-bit DMA cycles. Table 11 shows the mapping from the DMA subsystem signals to slot bus signals. Table 12 on page 96 shows the mapping of the AMD-756 controller DMA subsystem signals to PCI address bus signals.

**Table 11. DMA Addressing for ISA Bus Accesses (DMA/Slot Bus)**

| Page Register Outputs | Middle Address Latch Outputs | 8237 Address Outputs | DMA1 ISA Address Bits | DMA2 ISA Address Bits |
|---|---|---|---|---|
| M[7] | | | LA[23] | LA[23] |
| M[6] | | | LA[22] | LA[22] |
| M[5] | | | LA[21] | LA[21] |
| M[4] | | | S/LA[20] | S/LA[20] |
| M[3] | | | S/LA[19] | S/LA[19] |
| M[2] | | | S/LA[18] | S/LA[18] |
| M[1] | | | S/LA[17] | S/LA[17] |
| M[0] | | | S/LA[16] | – |
| | D[7] | | S/LA[15] | S/LA[16] |
| | D[6] | | S/LA[14] | S/LA[15] |
| | D[5] | | S/LA[13] | S/LA[14] |
| | D[4] | | S/LA[12] | S/LA[13] |
| | D[3] | | S/LA[11] | S/LA[12] |
| | D[2] | | S/LA[10] | S/LA[11] |
| | D[1] | | S/LA[9] | S/LA[10] |
| | D[0] | | S/LA[8] | S/LA[9] |
| | | A[7] | S/LA[7] | S/LA[8] |
| | | A[6] | S/LA[6] | S/LA[7] |
| | | A[5] | S/LA[5] | S/LA[6] |
| | | A[4] | S/LA[4] | S/LA[5] |
| | | A[3] | S/LA[3] | S/LA[4] |

**Table 11. DMA Addressing for ISA Bus Accesses (DMA/Slot Bus) (continued)**

| Page Register Outputs | Middle Address Latch Outputs | 8237 Address Outputs | DMA1 ISA Address Bits | DMA2 ISA Address Bits |
|---|---|---|---|---|
| | | A[2] | S/LA[2] | S/LA[3] |
| | | A[1] | S/LA[1] | S/LA[2] |
| | | A[0] | S/LA[0] | S/LA[1] |
| | | VSS | – | S/LA[0] |
| | | A[0]# | SBHE# | – |
| | | VSS | – | SBHE# |

**Table 12. DMA Addressing for ISA Bus Accesses (DMA/PCI AD Bus)**

| Page Register Outputs | Middle Address Latch Outputs | 8237 Address Outputs | DMA1 ISA Address Bits | DMA2 ISA Address Bits |
|---|---|---|---|---|
| 0 | | | AD[31] | AD[31] |
| 0 | | | AD[30] | AD[30] |
| 0 | | | AD[29] | AD[29] |
| 0 | | | AD[28] | AD[28] |
| 0 | | | AD[27] | AD[27] |
| 0 | | | AD[26] | AD[26] |
| 0 | | | AD[25] | AD[25] |
| 0 | | | AD[24] | AD[24] |
| M[7] | | | AD[23] | AD[23] |
| M[6] | | | AD[22] | AD[22] |
| M[5] | | | AD[21] | AD[21] |
| M[4] | | | AD[20] | AD[20] |
| M[3] | | | AD[19] | AD[19] |
| M[2] | | | AD[18] | AD[18] |
| M[1] | | | AD[17] | AD[17] |
| M[0] | | | AD[16] | – |
| | D[7] | | AD[15] | AD[16] |
| | D[6] | | AD[14] | AD[15] |
| | D[5] | | AD[13] | AD[14] |
| | D[4] | | AD[12] | AD[13] |
| | D[3] | | AD[11] | AD[12] |
| | D[2] | | AD[10] | AD[11] |
| | D[1] | | AD[9] | AD[10] |

**Table 12. DMA Addressing for ISA Bus Accesses (DMA/PCI AD Bus) (continued)**

| Page Register Outputs | Middle Address Latch Outputs | 8237 Address Outputs | DMA1 ISA Address Bits | DMA2 ISA Address Bits |
|---|---|---|---|---|
| | D[0] | | AD[8] | AD[9] |
| | | A[7] | AD[7] | AD[8] |
| | | A[6] | AD[6] | AD[7] |
| | | A[5] | AD[5] | AD[6] |
| | | A[4] | AD[4] | AD[5] |
| | | A[3] | AD[3] | AD[4] |
| | | A[2] | AD[2] | AD[3] |
| | | A[1] | – | AD[2] |
| | | A[0] | – | BE[1], BE[0] |
| | | A[0]# | – | BE[3], BE[2] |
| | | A[1] + A[0] | BE[0]# | – |
| | | A[1] + A#[0] | BE#[1] | – |
| | | A[1] + A[0] | BE#[2] | – |
| | | A#[1] + A#[0] | BE#[3] | – |

## 5.10.6 Type F DMA

Type F DMA is supported on all channels. The channels can be individually enabled to provide Type F DMA timing, using the Type F DMA control register (Function 0, offset 45h) as shown in Table 13 on page 97. Therefore, configuration software needs to detect Type F-capable devices and configure their channels only once after reset.

**Table 13. Type F DMA Control**

| Offset 45h | Type F DMA Control | Default |
|---|---|---|
| Bit 7 = 1 | ISA Master/DMA to PCI Line Buffer | 0 |
| Bit 6 = 1 | Enable DMA Type F Timing on Channel 7 | 0 |
| Bit 5 = 1 | Enable DMA Type F Timing on Channel 6 | 0 |
| Bit 4 = 1 | Enable DMA Type F Timing on Channel 5 | 0 |
| Bit 3 = 1 | Enable DMA Type F Timing on Channel 3 | 0 |
| Bit 2 = 1 | Enable DMA Type F Timing on Channel 2 | 0 |
| Bit 1 = 1 | Enable DMA Type F Timing on Channel 1 | 0 |
| Bit 0 = 1 | Enable DMA Type F Timing on Channel 0 | 0 |

When Type F DMA is enabled for a channel, Type F DMA transfers occur during the DACK# for that channel. That is, the programmed timing parameters are ignored, DMA cycles occur with zero wait states, and the DMA clock is set equal to BCLK.

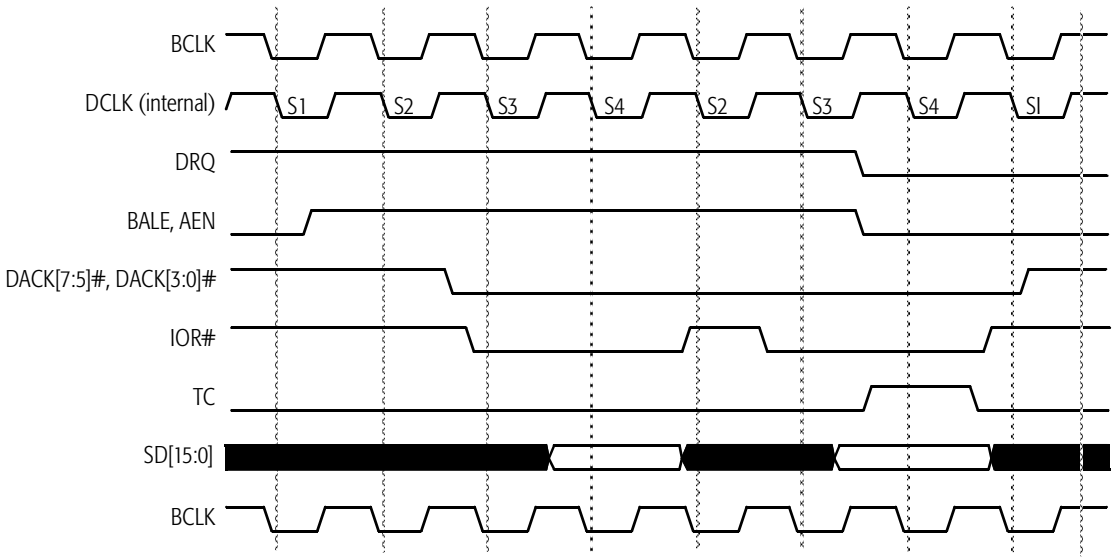


**Figure 22. Type F DMA Timing**

## 5.10.7 DMA Channel Mapping Registers

DMA channel mapping allows the selection of any DMA channel number for each Plug-n-Play DMA request/acknowledge signal pair. The mapping register allows each Plug-n-Play DMA pin pair to be connected to any DMA channel. When a Plug-n-Play DMA pin pair is connected to a DMA channel, that channel's normal ISA pin pair is disabled so that the DRQ is ignored and the DACK# is driven high.

## 5.10.8 Ready Control Logic

The Ready input to each of the 8237 megacells is driven from the same source within the ready control logic. The AMD-756 controller ready control logic forces the preprogrammed number of wait states on every DMA transfer.

If needed, the external signal IOCHRDY goes into the ready control logic to extend transfer signals. To add extra wait states, an external device can pull IOCHRDY low within the

setup time before the second phase of the internal DMA clock no later than the last forced wait state cycle. The current DMA cycle is then extended by inserting wait states until IOCHRDY is returned high. IOCHRDY going high must meet the setup time at the beginning of a wait state or an extra wait state is inserted before the DMA controller transitions to state S4.

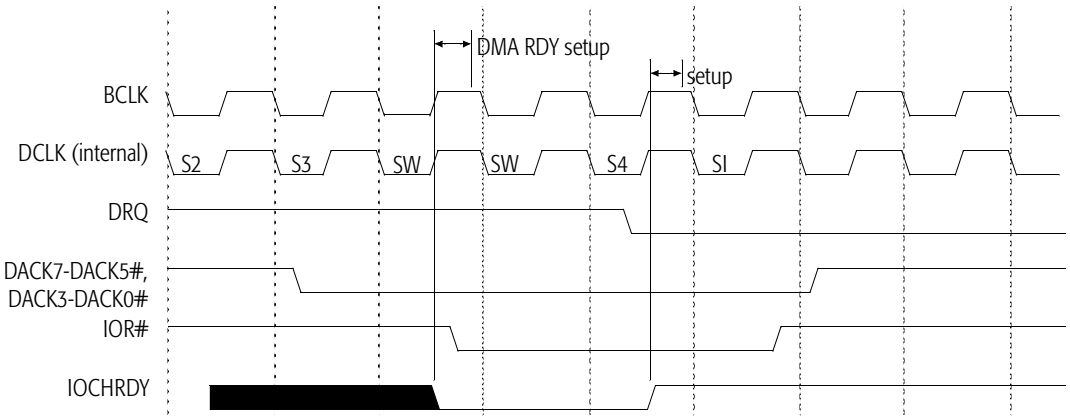


**Figure 23. DMA Ready Timing**

## 5.10.9 External Cascading

An external DMA controller or bus master can be attached to an AT-compatible design through the AMD-756 DMA controllers. To add an external DMA controller, one of the seven available DMA channels must be programmed in the cascade mode. This channel's DRQ signal is then connected to the external DMA controller's HLDA input. When one of the seven channels is programmed in the cascade mode and that channel is acknowledged, the AMD-756 controller does not drive the data bus, the command signals, or the address bus.

An external device can become a bus master and control the system address, data, and command buses in much the same manner. To enable this control, one of the external channels must be programmed in the cascade mode. The external device then asserts the DRQ line for that channel. When the channel's DACK# line goes asserted, the external device can then pull the MASTER# signal low. As in the DMA controller cascading, the AMD-756 does not drive the address, data, and command signals while the cascaded channel's DACK# signal is asserted.

# 5.11 Distributed DMA Support

Distributed DMA is PCI bus mastering with a legacy-compatible programming mode. It offers upward compatibility for ISA legacy devices in PCI bus systems, providing a vast improvement in performance over previous generation devices.

Each channel in the 8237 DMA controller is mapped to an individual DMA slice. The channel 0 base address register, current address, base count and current count, command, status, request etc. are mapped to DMA Slice DMA0. Each slice exists in a separate, non-overlapping I/O address space in the PCI bus space.

The Distributed DMA control register is located in Function 0, offset 60h–6Fh. Each channel base address can be individually programmed and enabled.

## 5.11.1 Target DMA Channel

Each target DMA channel has a block of sixteen 8-bit registers which are defined in Table 14. This block is locatable anywhere in the Legacy 64K I/O space by programming the Target DMA Configuration Register. All target DMA channels must have an identical programming model. The master DMA is programmed with the base address of each target DMA by having a matching base address register for each channel.

**Table 14. Programming Model for Single Target DMA Channel**

| Target Address | Read/ Write | Register Name | Byte DMA Address | Word DMA Address | POR Value |
|---|---|---|---|---|---|
| b + 0h | W | Base Address 0–7 | CH0 =0000h<br>CH1 =0002h<br>Ch2 =0004h<br>CH3 =0006h | CH4 =00C0h<br>CH5 =00C4h<br>CH6 =00C8h<br>CH7 =00CCh | XXh |
| b + 0h | R | Current Address 0–7 | CH0 =0000h<br>CH1 =0002h<br>Ch2 =0004h<br>CH3 =0006h | CH4 =00C0h<br>CH5 =00C4h<br>CH6 =00C8h<br>CH7 =00CCh | XXh |

**Table 14. Programming Model for Single Target DMA Channel (continued)**

| Target Address | Read/ Write | Register Name | Byte DMA Address | Word DMA Address | POR Value |
|---|---|---|---|---|---|
| b + 1h | W | Base Address 8–15 | CH0 =0000h<br>CH1 =0002h<br>Ch2 =0004h<br>CH3 = 0006h | CH4 =00C0h<br>CH5 =00C4h<br>CH6 =00C8h<br>CH7 =00CCh | XXh |
| b + 1h | R | Current Address 8–15 | CH0 =0000h<br>CH1 =0002h<br>Ch2 =0004h<br>CH3 =0006h | CH4 =00C0h<br>CH5 =00C4h<br>CH6 =00C8h<br>CH7 =00CCh | XXh |
| b + 2h | W | Base Address 16–23 | Ch0 =0087h<br>CH1 =0083h<br>Ch2 =0081h<br>CH3 =0082h | CH4 =N/A<br>CH5 =008Bh<br>CH6 =0089h<br>Ch7 =008Ah | XX |
| b + 2h | R | Current Address 16–23 | Ch0 =0087h<br>CH1 =0083h<br>Ch2 =0081h<br>CH3 =0082h | CH4 =N/A<br>CH5 =008Bh<br>CH6 =0089h<br>Ch7 = 008Ah | XXh |
| b + 3h | W | Base Address 24–31 | N/A | N/A | |
| b + 3h | R | Current Address 24–31 | N/A | N/A | |
| b + 4h | W | Base Word Count 0–7 | Ch0 =0001h<br>Ch1 =0003h<br>Ch2 =0005h<br>Ch3 =0007h | CH4 =00C2h<br>CH5 =00C6h<br>Ch6 =00CAh<br>Ch7 =00CEh | XXh |
| b +4h | R | Current Word Count 0–7 | Ch0 =0001h<br>Ch1 =0003h<br>Ch2 =0005h<br>Ch3 =0007h | CH4 =00C2h<br>CH5 =00C6h<br>Ch6 =00CAh<br>Ch7 =00CEh | XXh |
| b + 5h | W | Base Word Count 8–15 | Ch0 =0001h<br>Ch1 =0003h<br>Ch2 =0005h<br>Ch3 =0007h | CH4 =00C2h<br>CH5 =00C6h<br>Ch6 =00CAh<br>Ch7 =00CEh | XXh |
| b + 5h | R | Current Word Count 8–15 | Ch0 =0001h<br>Ch1 =0003h<br>Ch2 =0005h<br>Ch3 =0007h | CH4 =00C2h<br>CH5 =00C6h<br>Ch6 =00CAh<br>Ch7 =00CEh | XXh |
| b + 6h | W | Base Word Count 16–23 | N/A | N/A | |
| b + 6h | R | Current Word Count 16–23 | N/A | N/A | |
| b + 7h | N/A | Reserved (note 1) | | | |
| b + 8h | W | Command | 0008h | 00D0h | 00h |
| b + 8h | R | Status | 008h | 00D0h | X0h |
| b + 9h | W | Request | 0009h | 00D2h | 00h |
| b + Ah | N/A | Reserved (note 1) | | | |
| b + Bh | W | Mode | 000Bh | 00D6h | 00h |
| b + Ch | W | Reserved (note 1) | | | |

**Table 14. Programming Model for Single Target DMA Channel (continued)**

| Target Address | Read/ Write | Register Name | Byte DMA Address | Word DMA Address | POR Value |
|---|---|---|---|---|---|
| b + Dh | W | Master Clear | 000Dh | 00DAh | N/A |
| b + Eh | N/A | Reserved (note 1) | | | |
| b + Fh | W | Single-Channel Mask | 000Ah | 00D4h | 00h |
| b + Fh | R | Single-Channel Mask | config CFh | config EFh | 00h |

***Note:***
*1. Reads return all zeroes. Writes have no effect.*

## 5.11.2 DMA Control Registers

There are two physical DMA controllers in a Legacy PC system, one for byte transfers and one for word transfers, so there are at least two possible control registers for each register defined. The byte transfer channels are channels 0–3, and their registers are mapped to the byte DMA control registers. The word transfer channels are channels 4–7, and their registers are mapped to the word DMA control registers. Channel 4 is used to connect the two DMA devices together in an ISA system, so it is not available as a separate channel.

**Command Register**

The functionality of this register is identical to the legacy DMA controller, so data is passed through unchanged.

**Mode Register**

Data bits 1–0 are reserved. They are written undefined by the master DMA. The legacy DMA controller expects the channel number encoded in these bits. Each target DMA channel encodes the lower two bits of its channel number into the lower two bits of the data, replacing the two undefined bits.

The functionality of the remainder of this register is identical to the legacy DMA controller, so data is passed through unchanged.

**Request Register**

Data bits 1–0 are reserved. They are written undefined by the master DMA. The legacy DMA controller expects the channel number encoded in these bits. Each target DMA channel encodes the lower two bits of its channel number into the lower two bits of the data, replacing the two undefined bits. The functionality of the remainder of this register is identical to the legacy DMA controller, so data is passed through unchanged.

**Single-Channel Mask Register**

In writes to this register, the master DMA writes the new mask value in data bit 0. Data bits 1, 2, and 3 are reserved and are undefined by the master DMA. The legacy DMA controller expects the channel number encoded in bits 1–0 and the mask bit passed in bit 2. Each target DMA channel encodes the lower two bits of its channel number into the lower two bits of the data, replacing bits 1–0. The mask bit written in bit 0 is copied intact to bit 2 and bit 3 is cleared. The functionality of the remainder of this register is identical to the legacy DMA controller, so data is passed through unchanged.

In reads of this register, the master DMA reads the current mask value in bit 0. The legacy DMA controller's single-channel mask register is write-only, therefore the multi-channel mask shadow register is read. It returns the mask bits for all four channels in the DMA controller in such a way that the channel 0 mask is returned in bit 0, the channel 1 mask in bit 1, the channel 2 mask in bit 2, and the channel 3 mask in bit 3. The bit corresponding to the target channel number is copied to bit 0 and the remaining bits are cleared.

**Status Register**

The master DMA reads the current terminal count (TC) status value replicated four times in data bits 0–3 and the current channel request (DRQ) status value replicated four times in data bits 4–7. The legacy DMA controller's status register returns the terminal count status and request bits for all four channels in the DMA controller. The TC bit corresponding to the target channel number is copied to bits 0–3, and the DRQ bit corresponding to the target channel number is copied to bits 4–7.

### 5.11.3 DMA Software Commands

**Master Clear**

The functionality of this register is identical to the legacy DMA controller, so data is passed through unchanged.

### 5.11.4 DMA Addressing

Each legacy DMA channel has two legacy addresses defined to store the base memory address and count information. Located at these byte legacy addresses are 16-bit registers. The state of the first/last flip-flop determines which byte (high or low) is being accessed. The target DMA does not suffer this problem

because it has fully decoded these registers. Table 15 on page 104 shows the relationship between legacy DMA addressing for Base, Count, and Memory Page registers. It also shows where this information is programmed into the target DMA. For the byte legacy DMA, bits 0–7 represent address 0–7. However, for the word legacy DMA, bits 0–7 represent address 1–8. This carries forward to the next address byte. The memory page register re-aligns the bit position to the address. This relationship is maintained in the target DMA. A target DMA can be programmed to be in 8-bit/16-bit transfer mode from its PCI configuration space. This mode information defines how the target DMA treats the data in the registers. Table 15 also defines optional non-legacy addressing extensions for the target.

**Table 15. DMA Registers**

| Legacy Channel | Base Address | Base Address | Memory Page | Count Address | Count Address |
|---|---|---|---|---|---|
| Channel 0 | 0000h | 0000h | 0087h | 0001h | 0001h |
| Channel 1 | 0002h | 0002h | 0083h | 0003h | 0003h |
| Channel 2 | 0004h | 0004h | 0081h | 0005h | 0005h |
| Channel 3 | 0006h | 0006h | 0082h | 0007h | 0007h |
| | **Address 1–8** | **Address 9–16** | **Address 17–23** | **Address 1–8** | **Address 9–16** |
| Channel 4 | 00C0h | 00C0h | N/A | 00C2h | 00C2h |
| Channel 5 | 00C4h | 00C4h | 008Bh | 00C6h | 00C6h |
| Channel 6 | 00C8h | 00C8h | 0089h | 00CAh | 00CAh |
| Channel 7 | 00CCh | 00CCh | 008Ah | 00CEh | 00CEh |
| **Above Channels Map to Target Address** | **Base + 0h** | **Base + 1h** | **Base + 2h** | **Base + 4h** | **Base + 5h** |
| 8-Bit Mode | Address 0–7 | Address 8–15 | Address 16–23 | Address 0–7 | Address 8–15 |
| 16-Bit Mode | Address 1–8 | Address 8–16 | Address 17–23 | Address 1–8 | Address 8–16 |
| Non-Legacy Target DMA Addressing Extensions | Base Address Base + 3h | | | Count Address Base + 6h | |
| 8-Bit Mode | Address 24–31 | | | Address 16–23 | |
| 16-Bit Mode | Address 24–31 | | | Address 17–23 | |

***Notes:***

1. *Any target DMA that does not support the non-legacy extensions must always return a value of 00h from these locations when read.*
2. *It is the responsibility of the master DMA to support the reserved memory page registers. Because the AMD-756 controller implements subtractive decoding for these registers, master DMA blocks that implement them behave as expected by the distributed DMA specification.*

### 5.11.5 PCI Target DMA Configuration Registers

There must be one target configuration register for each target channel in a device, with bit 0 being the channel enable bit. The target base address, along with a matching base address in the master DMA indicates the DMA channel to which the target DMA is mapped. No two target DMA channels can be programmed with the same target base address, because bits 6–4 of the base address are read-only values that equal the channel number.

The target DMA is only required to support at least one transfer size. The first four target DMA channels only support 8-bit transfers, so bits 2 and 1 always read 00b. The second four target DMA channels only support 16-bit transfers, so bits 2 and 1 always read 01b. No other transfer sizes are supported.

Non-legacy extended addressing is not supported. The DMA target channel accepts writes to bits 31–24 of the address register and bits 23–16 of the count register, with reads from those bits returning zeroes for data.

## 5.12 ISA Bus Refresh Cycle Types

The AMD-756 controller supports decoupled refresh mode only. The PC/AT-compatible refresh period of 15.625 microseconds is supported by dividing the OSC signal. The AMD-756 supports only off-board refresh timing. Data in DRAM on the ISA bus is refreshed every 15.64 microseconds.

A refresh request can be generated by either the AMD-756 controller in PCI bus master mode, or by an add-on card in ISA master mode. The only difference between the refresh requests is that the requester drives the REFRESH# pin. The refresh address is put on SA[8:0] by the AMD-756 (regardless of which master currently owns the bus) in response to a low REFRESH# signal. The SA[16:9] addresses are tri-stated. SA[19:17] are driven low. MEMR# is asserted by the AMD-756 controller one BCLK cycle after REFRESH# goes asserted. MEMR# remains low for two BCLK cycles. The REFRESH# signal is negated one BCLK period after MEMR# negates.

# 5.13 Fast IDE/EIDE Interface

The AMD-756 controller includes an IDE controller that connects to the primary and secondary port pins as shown in Figure 24. The IDE controller is accessed through function 1 PCI configuration registers.

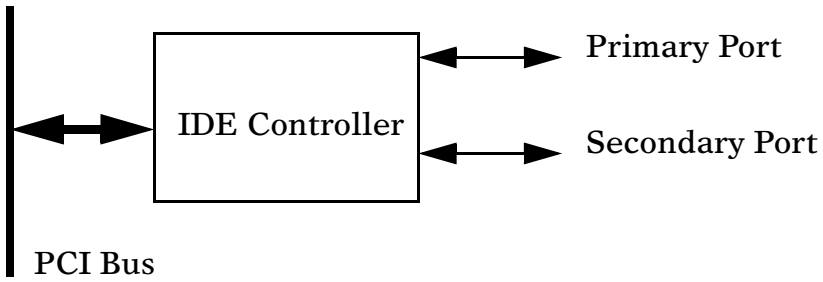


**Figure 24. IDE Controller Connections**

## 5.13.1 IDE Drive Registers

The IDE registers are 1F0h through 1F7h for the primary channel and 170h through 177h and 376h for the secondary channel. These registers are not resident in the AMD-756, but are incorporated into the actual drive mechanism. The contents of the IDE registers are relatively straightforward, but the legacy ATA registers are detailed here for completeness. The address map for these registers is shown in Table 16.

## 5.13.2 IDE Configuration Registers

Each IDE channel has a complete and independent set of configuration registers. The registers for the primary channel and the secondary channel are identical except for their addresses in PCI configuration space Function 1. The primary channel registers are located at offset 10h–1Bh. The secondary channel registers are located at offset 18h–1Fh.

**Table 16. IDE Register Map**

| Channel 0 | Channel 1 | Type | Description |
|---|---|---|---|
| 1F0h | 170h | Read/Write | Data register (16-bit) |
| 1F1h | 171h | Read-Only<br>Write-Only | Error Register (8-bit)<br>Features Register (8-bit) (former Write Compensation Register) |
| 1F2h | 172h | Read/Write | Sector Count Register (8-bit) |
| 1F3h | 173h | Read/Write | Sector Number Register |
| 1F4h | 174h | Read/Write | Low Cylinder Number Register (8-bit) |
| 1F5h | 175h | Read/Write | High Cylinder Number Register (8-bit) |
| 1F6h | 176h | Read/Write | Drive/Head Register (8-bit) |
| 1F7h | 177h | Read-Only<br>Write-Only | Status Register (8-bit)<br>Command Register (8-bit) |
| 3F6h | 376h | Read-Only<br><br>Write-Only | Alternate Status Register (8-bit)—Contains the same information as the status register at offset 1F7h but does not clear the interrupt or imply interrupt acknowledge<br>Device Control Register (8-bit)—Bit 2 is the software reset bit. Bit 1 is the enable bit for the drive interrupt to the host. |

## 5.13.3 Ultra DMA Support

Ultra DMA is a data transfer protocol for ATA/ATAPI-4 to be used with READ DMA and WRITE DMA commands and data transfers for PACKET commands. The AMD-756 controller supports Ultra DMA/33 transfer mode 0, 1, 2. It also supports mode 3 and 4 defined in a new proposal for Ultra DMA-66, which doubles the transfer rates to 66 Mbytes/seconds. The UDMA-66 protocol enhances data integrity by use of an 80-pin conductor cable and by CRC checking. Table 17 lists the cycle times and throughput for the different modes in the UDMA protocol.

**Table 17. Ultra DMA Protocol Modes**

| Mode | Cycle Time | Source | Ultra DMA |
|---|---|---|---|
| 0 | 235 ns | 16Mbytes/second | UDMA-33 |
| 1 | 160 ns | 24Mbytes/second | |
| 2 | 120 ns | 33Mbytes/second | |
| 3 | 90 ns | 44Mbytes/second | UDMA-66 |
| 4 | 60 ns | 66Mbytes/second | |

The Ultra DMA protocol uses the existing signal pins in the IDE connector. A number of standard IDE signals are redefined when the device is in the UDMA mode. Table 18 lists the IDE drive cable interface that are redefined as the Ultra DMA interface signals.

**Table 18. Ultra DMA Interface Signal Redefinition**

| Standard IDE Signal | Driven By | UDMA/33/66 Read Definition | UDMA/33/66 Write Definition | AMD-756™ Signal |
|---|---|---|---|---|
| DIOR# | Host | DMARDY# | STROBE | |
| DIOW# | Host | STOP# | STOP# | |
| IORDY | Device | STROBE | DMARDY# | |

**The DIOR#** signal is redefined as DMARDY# during a read transaction for transferring data from the UDMA device to the AMD-756 controller. The AMD-756 uses the DMARDY# 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 during a write transaction when transferring data from the AMD-756 controller to the UDMA device. This signal is driven by the AMD-756 and data is transferred during each rising and falling edge transition.

**DIOW#** signal is redefined as STOP during read and write transaction. This signal is always driven by the AMD-756 controller and is used to request that a transfer be stopped.

**IORDY** signal is redefined as STROBE during a read transaction for transfer. This signal is driven by the UDMA device and data is transferred during each rising and falling edge transition. DIOR# signal is redefined as DMARDY# during a write transaction when transferring data from the AMD-756 to the UDMA device. The UDMA device uses DMARDY# to signal when it is ready to transfer data and to add wait states to the current transaction.

A READ DMA or WRITE DMA command or data transfer for a PACKET command is accomplished through a series of input or output data bursts. Each burst has three phases of operation, the burst initial phase, the data transfer phase, and the burst termination phase.

The burst initial phase begin with the assertion of DMARQ by the device and ends when the sender toggles STROBE to transfer the first data word. The data transfer phase is then in effect until the burst termination phase, which begins either when the host asserts STOP or the device negates DMARQ.

The device asserts DDRQ to initiate a burst. The host asserts DDACK# when it is ready to begin the requested burst. For read cycles, the host releases DATA, the device asserts DSTROBE, and the host negates STOP and asserts DMARDY#. The device then drives the first word of the data transfer onto DATA. The data is transferred when the device negates DSTROBE. For write cycles, the device asserts DDMARDY# after the host has negated STOP. The host drives the first word of the data transfer onto DATA. The data is transferred when the host toggles HSTROBE.

The device (reads) or host (writes) continues to drive a data word onto DATA and toggles DSTROBE to latch the data until the data transfer is complete or the burst is paused. Either the device or the host can pause a burst transfer. The device pauses the read DMA burst by halting DSTROBE toggling, and resumes the burst by toggling DSTROBE again. The host pauses a read burst by negating HDMARDY# and resumes the burst by reasserting HDMARDY#.

Either the device or the host can terminate a burst. A burst must be paused before it can be terminated. The host responds by asserting STOP and with the device negating DMARDY#. The device can then stop the burst by negating DDRQ and host acknowledges by asserting STOP. The transmitter then drives the STROBE signal to a high level. The host then places the result of its CRC (Cyclic Redundancy Check) on DATA and negates DDACK#. The data is latched in the device at the negating edge of DDACK#.

### UDMA33/66 Protocol Changes

Due to the shorter cycle times of the UDMA-66 protocol, more data words can be sent after the DMARDY signal is set to Low. With UDMA-33, modes 2, 1, and 0 could receive at least two data words whenever the receiver entered or resumed an Ultra DMA burst. With UDMA-66, modes 3 and 4 can receive at least three data words whenever the receiver enters or resumes an Ultra DMA burst.

The UDMA-66 proposal specifies a new 80-pin conductor cable, which is required for modes 3 and 4. This conductor cable reduces crosstalk by adding 40 additional grounds between the 40 standard ATA signal and ground lines. To avoid running at 66 Mbytes/second without the proper cable, the host must verify the cable type for modes 3 and 4.

The 80-pin cable can be detected either by using the state of the PDIAG#:CBLID# signal or through the Identify Device command or Identify Packet Device command. To detect the state of the of the PDIAG#:CBLID# signal, the host checks to see if PDIAG#:CBLID# is grounded or pulled up. If this signal is grounded, an 80-pin conductor cable is installed, because this signal is grounded in the 80-pin conductor cable host connector. If this signal is pulled up, a 40-pin conductor cable is installed, because this signal is connected to the UDMA-66 devices and is pulled up through a 10-Kohm resistor at each device.

To detect the 80-pin cable with the Identify Device command or Identify Packet Device command, a capacitor is installed from PDIAG#:CBLID# to ground. The UDMA-66 drive detects the presence or absence of this capacitor after receiving the Identify Device command or Identify Packet Device command. The UDMA-66 drive then discharges the capacitor and measures the signal line. If an 80-pin conductor cable is installed, the UDMA-66 drive detects this signal as pulled up, because this signal is not connected to the UDMA-66 drive in the 80-pin cable. If a 40-pin cable is installed, the rise time of the signal is slow enough that it can be sampled by the UDMA-66 drive while it is below $V_{IL}$. The capacitor test results are reported to the host in the data returned by the Identify Device command or Identify Packet Device command.

# 5.14 Power Management Support

## 5.14.1 Power Management Subsystem

The power management function of the AMD-756 controller is indicated in the following block diagram. This block includes logic for most of the multiplexed-function pins-such as general-purpose I/O (GPIO) pins, the power management (PM) pins, system management bus (SMBus) pins, and the plug and play (PNP) pins-as well as the logic for ACPI-compliant power management for desktop systems. Register access to most of this logic is contained in the configuration space for function 3 and the 256-byte I/O space (defined by the configuration space) called I/O Mapped Power Management +xx. Here is a general diagram of this block.

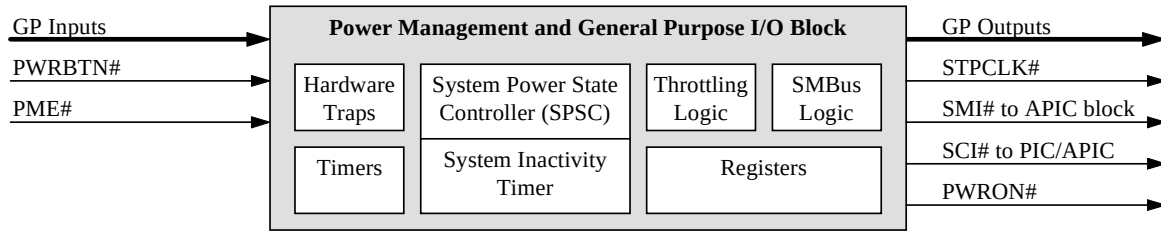


**Figure 25. Power Management and General Purpose I/O**

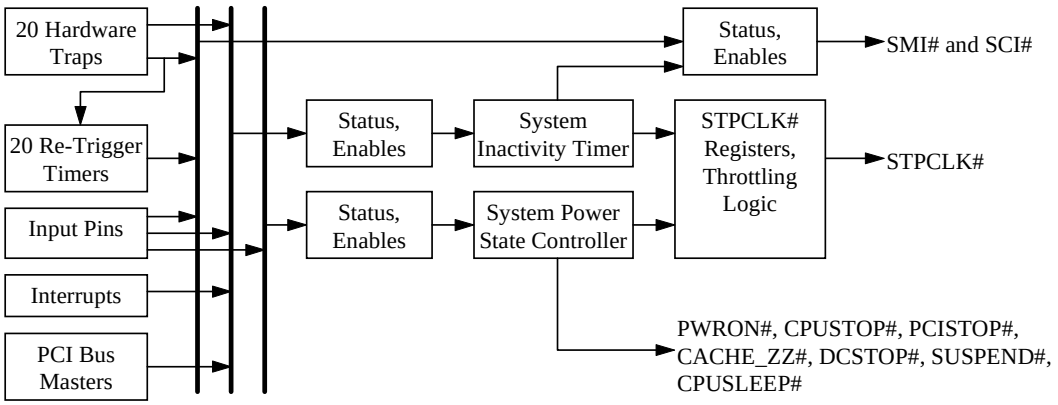


**Figure 26. Basic Power Management Block Diagram**

## 5.14.2 Power Plane Management

The following table summarizes the events that can be detected by the system management logic and the registers in which hardware responses can be enabled for each. The columns are STS, where the status bits are accessible, EVT, where the grouped statuses can be read, SCI/SMI EN, where the ACPI interrupts can be enabled, SMI_EN, where the SMI interrupts can be enabled, SIT_EN, where the events can be enabled to reload the system inactivity timer, and the resume columns which show where the registers to enable the resume events from C2, C3, POS, and SOFF to FON (see the system power state controller section, below).

**Table 19. SMM Events**

| Events | STS | Event EVT | SCI/SMI EN | SMI only SMI_EN | Reload SIT_EN | Resume POS EN | Resume C2-C3 EN | Resume SOFF EN |
|---|---|---|---|---|---|---|---|---|
| SMBus host complete/error | PME0 | I/O Mapped Power Management +28h | I/O Mapped Power Management +E2h | I/O Mapped Power Management +2Ah | | | | |
| SMBus host as target transfer | I/O Mapped Power Management +E0h | | I/O Mapped Power Management +E2h | | | | I/O Mapped Power Management +16h | I/O Mapped Power Management +26h |
| SMBus snoop match | I/O Mapped Power Management +E0h | | I/O Mapped Power Management +E2h | | | | I/O Mapped Power Management +16h | I/O Mapped Power Management +26h |
| SMBALERT# pin | I/O Mapped Power Management +E0h | | I/O Mapped Power Management +E2h | | | | I/O Mapped Power Management +16h | |
| | | | | | | | | |
| IRQ[15:0] | | | | | I/O Mapped Power Management +B4h | | | |
| INTR (unmasked IRQs) | | | | | I/O Mapped Power Management +B4h bit[2] | | I/O Mapped Power Management +16h | |
| PCI bus masters | I/O Mapped Power Management | | | | I/O Mapped Power Management +B4h | | I/O Mapped Power Management +04h bit[1] (C3) | |
| | | | | | | | | |

**Table 19. SMM Events (continued)**

| Events | STS | Event EVT | SCI/SMI EN | SMI only SMI_EN | Reload SIT_EN | Resume POS EN | Resume C2-C3 EN | Resume SOFF EN |
|---|---|---|---|---|---|---|---|---|
| 20 Hardware traps | I/O Mapped Power Management +A8h | I/O Mapped Power Management +28h | I/O Mapped Power Management +ACh | I/O Mapped Power Management +2Ah | I/O Mapped Power Management +B0h | | | |
| 20 Re-trigger timers time out | I/O Mapped Power Management +A0h | I/O Mapped Power Management +28h | I/O Mapped Power Management +A4h | I/O Mapped Power Management +2Ah | | | | |
| | | | | | | | | |
| System inactivity timer time out | I/O Mapped Power Management +28h | | I/O Mapped Power Management +22h | I/O Mapped Power Management +2Ah | | | I/O Mapped Power Management +16h | |
| | | | | | | | | |
| USB bus resume event | I/O Mapped Power Management +24h | I/O Mapped Power Management +28h | I/O Mapped Power Management +25h | I/O Mapped Power Management +2Ah | | | I/O Mapped Power Management +16h | I/O Mapped Power Management +26h |
| 4 USB transaction types | I/O Mapped Power Management +24h | | I/O Mapped Power Management +25h | | | | | |
| | | | | | | | | |
| ACPI timer overflow | I/O Mapped Power Management | | I/O Mapped Power Management +02h | | | | | |
| | | | | | | | | |

**Table 19. SMM Events (continued)**

| Events | STS | Event EVT | SCI/SMI EN | SMI only SMI_EN | Reload SIT_EN | Resume POS EN | Resume C2-C3 EN | Resume SOFF EN |
|---|---|---|---|---|---|---|---|---|
| Power button override | I/O Mapped Power Management | | | | | | | I/O Mapped Power Management +26h (off) |
| BIOS-OS lock | I/O Mapped Power Management, bit[28] | | I/O Mapped Power Management +02h | I/O Mapped Power Management +2Ah | | | | |
| Software SMI | I/O Mapped Power Management +28h | | | I/O Mapped Power Management +2Ah | | | | |
| | | | | | | | | |
| 18 GPIO inputs | I/O Mapped Power Management +D4h | I/O Mapped Power Management +28h | I/O Mapped Power Management +D8h | I/O Mapped Power Management +2Ah | | | | |
| Real-time clock IRQ | I/O Mapped Power Management | | I/O Mapped Power Management +02h | | | | I/O Mapped Power Management +16h | I/O Mapped Power Management +26h |
| PWRBTN# pin | I/O Mapped Power Management | | I/O Mapped Power Management +02h | I/O Mapped Power Management +2Ah | | | I/O Mapped Power Management +16h | I/O Mapped Power Management +26h |
| EXTSMI# pin | I/O Mapped Power Management +28h | | I/O Mapped Power Management +22h | I/O Mapped Power Management +2Ah | | | I/O Mapped Power Management +16h | I/O Mapped Power Management +26h |

**Table 19. SMM Events (continued)**

| Events | STS | Event EVT | SCI/SMI EN | SMI only SMI_EN | Reload SIT_EN | Resume POS EN | Resume C2-C3 EN | Resume SOFF EN |
|---|---|---|---|---|---|---|---|---|
| PME# pin | I/O Mapped Power Management +28h | | I/O Mapped Power Management +22h | I/O Mapped Power Management +2Ah | | | I/O Mapped Power Management +16h | I/O Mapped Power Management +26h |
| RI# pin | I/O Mapped Power Management +28h | | I/O Mapped Power Management +22h | I/O Mapped Power Management +2Ah | | | I/O Mapped Power Management +16h | I/O Mapped Power Management +26h |
| SLPBTN# pin | I/O Mapped Power Management | | I/O Mapped Power Management +02h | I/O Mapped Power Management +2Ah | | | I/O Mapped Power Management +16h | I/O Mapped Power Management +26h |
| THERM# pin | I/O Mapped Power Management +28h | | I/O Mapped Power Management +22h | I/O Mapped Power Management +2Ah | | | | |

### 5.14.3 SCI and SMI Control

There are three categories of events that generate SCI/SMI interrupts. They are time-outs of device monitors-when a peripheral has not been accessed for a period of time-hardware traps, and input pins. Figure 27 shows how the control for the SCI/SMI interrupts is configured.

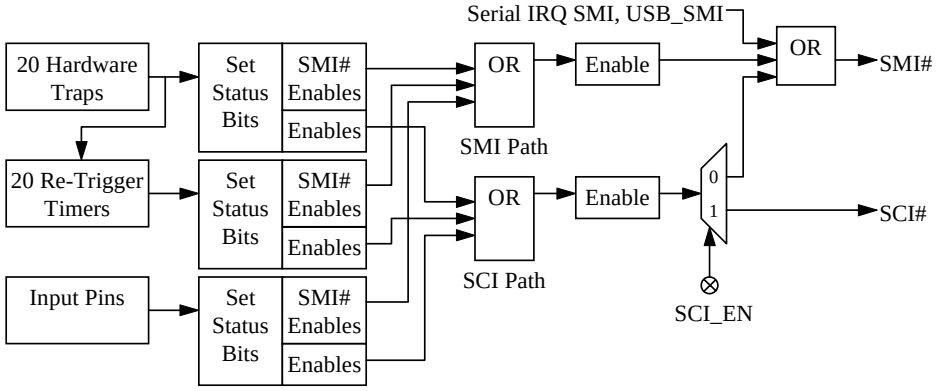


**Figure 27. SCI/SMI Control**

The AMD-756 controller USB block also provides a source of SMI that is ORed into the logic. The serial IRQ logic can generate SMIs as well. SMI# and SCI# go to the PCI/APIC blocks to generate CPU interrupts.

### 5.14.4 System Inactivity Timer

Any of the hardware traps, IRQ lines, or PCI bus master activity can reload the system inactivity timer. If the timer is enabled and decrements to zero, an interrupt is generated.

### 5.14.5 Throttling Logic

Throttling refers to the act of asserting STPCLK# to the CPU for a specified percentage of time in order to reduce the power being consumed by the CPU. Once throttling is started, STPCLK# looks like a clock with a period of 244 microseconds (based on 8 cycles of the 32.768 KHz. clock) and a duty cycle as

specified by control registers function 3 offset 50h and I/O Mapped Power Management + 10h.

Two types of throttling are possible: normal and thermal. Normal throttling is controlled by software. Thermal throttling is controlled by the THERM# pin. If both are enabled to occur at the same time, then the duty cycle specified for thermal throttling is used. Throttling is ignored in the SOFF, C2, C3, and POS power states. If throttling is enabled when entering these states, it stops, once the state is entered. After exiting the state, throttling continues.

### 5.14.6 System Power State Controller (SPSC)

The system power state controller (SPSC) supports the states shown in Table 20. Figure 28 shows the power state transitions.

**Mechanical off: MOFF.** MOFF is the state when VDD_SOFT is not powered. This can happen at any time, from any state, due to the loss of power to the VDD_SOFT plane (e.g., a power outage, the power supply is unplugged, or the power supply's mechanical switch). When power is applied to this plane, then the system transitions to either SOFF or FON, depending on whether the PWRON# pin has a pull-up or a pull-down resistor on it. The only resume event that is enabled after the transition from MOFF to SOFF is from PWRBTN#.

**Table 20. Power States**

| State | VDD3 | VDD_SOFT | VDD_RTC | Notes |
|---|---|---|---|---|
| Full on (FON) | On | On | On | Low-power state initiated by CPU. |
| C2 | On | On | On | All resume events available. |
| C3 | On | On | On | All resume events available. |
| Power on suspend (POS) | On | On | On | All resume events available. |
| Soft off (SOFF) | Off | On | On | USB transaction and IRQ resume events not available (however USB resume event available). |
| Mechanical off (MOFF) | Off | Off | On | No resume events available. |

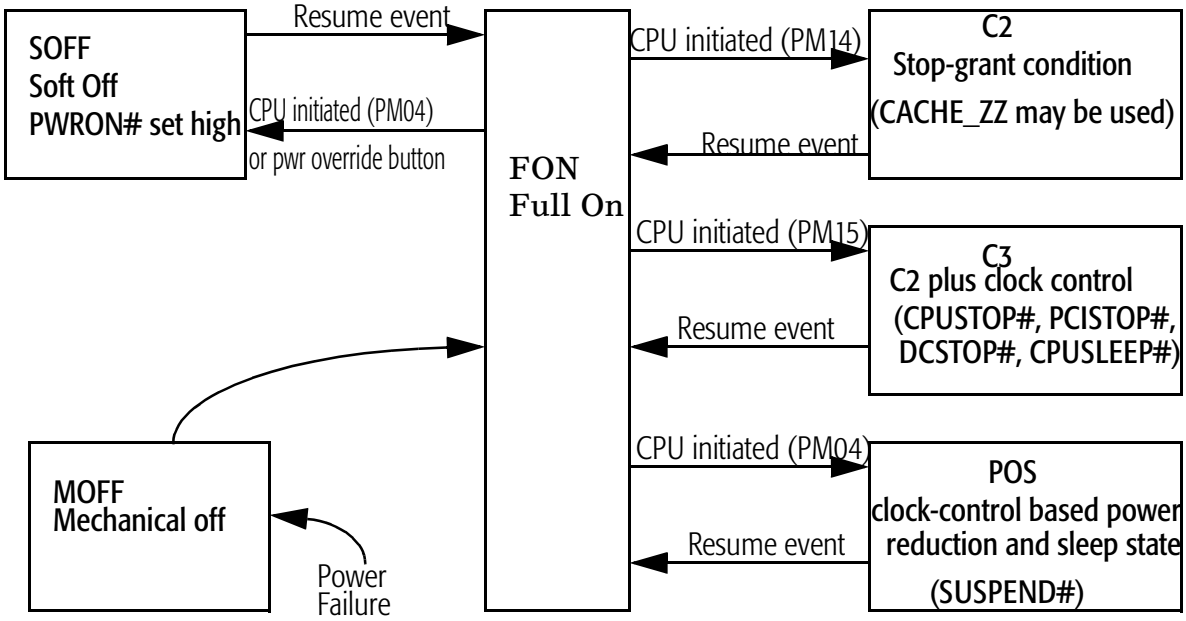


**Figure 28. Power State Transitions**

**Soft off: SOFF.** The soft off state appears to be “off” to the user. AMD-756 controller's VDD_SOFT plane is powered, but VDD3 is not. The system normally uses PWRBTN# to transition from SOFF to FON. But AMD-756 also allows SMBus activity, USB resume events, the real-time clock alarm, the EXTSMI# pin, the SLPBTN# pin, and the PME# pin to be enabled to cause this transition. All of the SOFF-to-FON enables are found in I/O Mapped Power Management +26h.

**Snoop-capable clock control: C2**. In C2, the CPU's internal clock is disabled via STPCLK#. The state is controlled by I/O Mapped Power Management +50h bits[7:0]. Per the ACPI specification, the CPU's cache can be snooped while in this state (however, if the CACHE_ZZ function is enabled, the L2 cache can not be snooped when in C2). The hardware exits this state when any of the enabled resume events specified by I/O Mapped Power Management+16h occur. These include SMBus activity, various pin transitions, INTR interrupts, SMI interrupts, system inactivity timer time out, and the USB-defined resume event.

**Snoop-disabled clock control: C3.** In C3, STPCLK# is asserted along with a few other clock control pins that result in a state

that prevents the processor's cache's from being snooped. The state is controlled by I/O Mapped Power Management +50h bits[15:8]. These include CPUSTOP#, which halts the CPU's clock via the system PLL chip, PCISTOP#, which halts the system's PCI clocks (except the one to the south bridge), DCSTOP#, which causes the DRAM controller to prepare for its host clock to stop by initiating self refresh cycles to system memory, and CPUSLEEP# which places the CPU into deep sleep. The resume events for C3 are the same as C2 except that PCI bus master requests are added.

**Power on suspend: POS.** The POS state is treated similarly to C3 by the AMD-756 controller. The state is controlled by I/O Mapped Power Management +50h bits[23:16]. The SUSPEND# pin is typically in this state, asserted along with all the enabled clock controls, which can be used control an external power plane. The POS resume events are the same as for C2 and C3, specified by I/O Mapped Power Management +16h.

### MOFF to FON and SOFF to FON

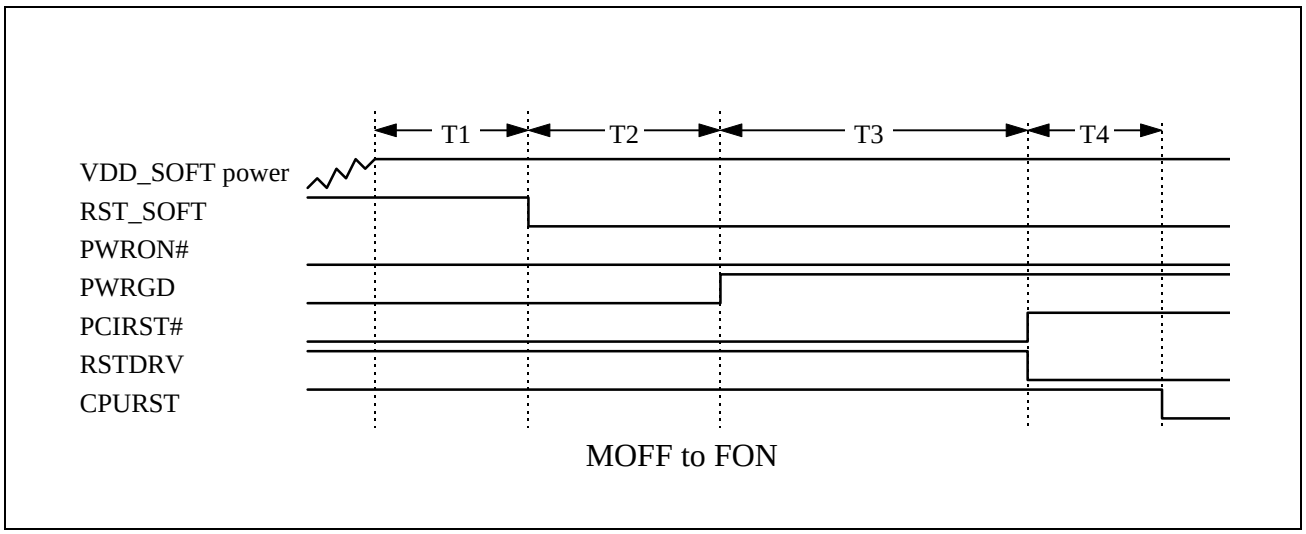


**Figure 29. Mechanical Off to Full On**

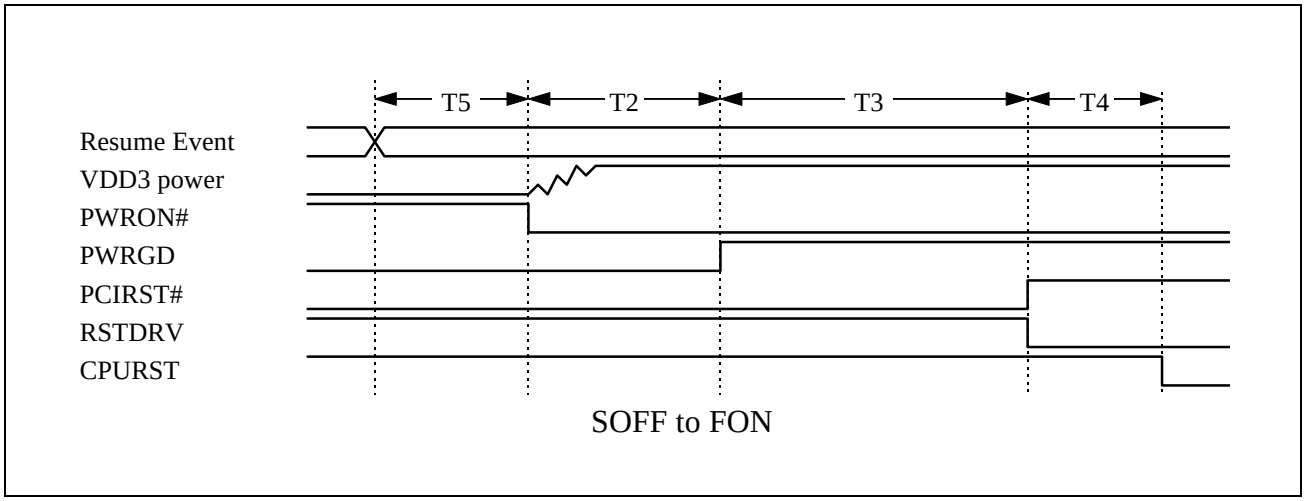


**Figure 30. Soft Off to Full On**

**Table 21. Power Transition Times for Figures Above**

| T | Minimum | Maximum |
|---|---|---|
| T1 | 25ms | 50 ms |
| T2 | 50 ms | |
| T3 | 1.8 ms | |
| T4 | 1.5 ms | |
| T5 | 32 µs | 64 µs |

**MOFF to SOFF**

The timing diagram for this sequence is similar to the first part of the MOFF to FON sequence. However, PWRON# rises with VDD_SOFT and is held high for the entire duration; thus, the other reset signals, PWRGD, PCIRST#, RSTDRV, and CPURST, never transition.

**C2, C3, or POS to SOFF**

This transition can be initiated by software via I/O Mapped Power Management +04h or by the power-button override event. These both cause PWRON# to go high. The assumption is that the power supply will respond by de-asserting PWRGD and dropping power to the system's main power supplies,

including VDD3. VDD_SOFT must remain valid throughout this transition. Immediately after PWRGD goes low, PCIRST#, RSTDRV, and CPURST become asserted. As VDD3 drops, these reset signals become invalid

### Transitions Between FON and C2, C3, or POS

Function 3 Offset 50h specifies the definition of these transitions as enables for the following eight signals: CPURST, SUSPEND#, CPUSLEEP#, PCISTOP#, CPUSTOP#, DCSTOP#, CACHE_ZZ. Any of these signals can be enabled for the transition to any of the three low-power states. The transition to C2, C3, and POS occurs as follows, for each of the enabled pin controls:

- **CPU initiation.** First, software commands that the proper state transition take place as follows: the transition to C2 is initiated by reading I/O Mapped Power Management + 14h, the transition to C3 is initiated by reading I/O Mapped Power Management + 15h, and the transition to POS is initiated by setting I/O Mapped Power Management + 04h bit [SLP_TYP] to 'b010 and writing I/O Mapped Power Management + 04h bit [SLP_EN] high.
- **Stop grant.** The AMD-756 controller then asserts STPCLK# and waits for the stop-grant cycle from the PCI bus (a special cycle as specified by function 3 offset 41h bit [STPGNT]) to be completed.
- **CACHE_ZZ.** Four PCLK cycles after stop-grant, CACHE_ZZ is asserted.
- **DCSTOP#**. Eight PCLK cycles after stop-grant, DCSTOP# is asserted.
- **CPUSLEEP#.** 64 PCLK cycles after stop-grant, CPUSLEEP# is asserted.
- **CPUSTOP# and PCISTOP#.** 68 PCLK cycles after stop-grant, CPUSTOP# and PCISTOP# are asserted.
- **SUSPEND#.** 80 PCLK cycles after stop-grant, SUSPEND# is asserted.

Resume events are enabled by I/O Mapped Power Management + 16h and, in the case of C3, I/O Mapped Power Management +03h bit[1]. Here is the resume sequence, once an enabled resume event occurs, if the SUSPEND# pin is utilized:

- **SUSPEND#.** Immediately after the resume event, the SUSPEND# pin is de-asserted along with CPUSLEEP# and

STPCLK#. If CPURST is enabled to be asserted by function 3 offset 50h, it is asserted at this time as well.

- **CPUSTOP# and PCISTOP#.** 17.7 milliseconds after SUSPEND#, CPUSTOP# and PCISTOP# are de-asserted.
- **DCSTOP#.** About one millisecond after CPUSTOP# and PCISTOP are de-asserted, DCSTOP# is de-asserted.
- **CPURST.** 4 PCLK cycles after DCSTOP#, CACHE_ZZ and CPURST are de-asserted.

Here is the resume sequence, once an enabled resume event occurs, if the SUSPEND# pin is not utilized, but DCSTOP#, PCISTOP# or CPUSTOP# is utilized:

- **CPUSTOP# and PCISTOP#.** CPUSTOP# and PCISTOP# are de-asserted immediately after the resume event.
- **DCSTOP#.** About one millisecond after the resume event, DCSTOP#, CPUSLEEP#, and CACHE_ZZ are de-asserted.
- **STPCLK#.** 4 PCLK cycles after DCSTOP#, STPCLK# is de-asserted.

Here is the resume sequence, once an enabled resume event occurs, if the SUSPEND#, PCISTOP#, CPUSTOP#, and DCSTOP# pins are not utilized:

- **CACHE_ZZ.** 4 PCLK cycles after the resume event, CACHE_ZZ is de-asserted.
- **STPCLK#.** 4 PCLK cycles after CACHE_ZZ, STPCLK# is de-asserted.

### 5.14.7 Serial IRQ Protocol

The AMD-756 controller supports the serial IRQ protocol. This logic controls the SERIRQ pin and outputs IRQs to the PIC/APIC blocks. This logic runs off of PCLK. It is specified by function 3 offset 4Ah. The serial IRQ logic does not provide support for generating IRQ0, IRQ2, IRQ8, or IRQ13.

### 5.14.8 SMBus

The AMD-756 controller includes a complete system management bus, or SMBus, interface. SMBus is a two-wire serial interface used to communicate with system devices such

as temperature sensors, clock chips, and batteries. The registers that specify this bus are I/O mapped power management +E0h through I/O mapped power management +EFh.

The SMBus interface includes a host controller and a host-as-target controller.

### Host Bus Controller

The host can generate cycles over the SMBus as a master. Software accomplishes this by setting up I/O Mapped Power Management +E2h bit[CYCTYPE] to specify the type of SMBus cycle desired and then (or concurrently) writing a 1 to I/O Mapped Power Management +E2h bit[HOSTST]. Then a cycle is generated with the various address, command, and data fields as specified by the registers called out in I/O Mapped Power Management +E2h bit[CYCTYPE].

Writes to the host controller registers I/O Mapped Power Management +E2h bits[3:0], I/O Mapped Power Management +E4h, I/O Mapped Power Management +E8h, and I/O Mapped Power Management +E9h are illegal while the host is busy with a cycle. If a write occurs to I/O Mapped Power Management +E2h while I/O Mapped Power Management +E0h bit[HST_BSY] is asserted, then the four LSBs be ignored. Writes to I/O Mapped Power Management +E4h, I/O Mapped Power Management +E8h, and I/O Mapped Power Management+E9h while I/O Mapped Power Management +E0h bit[HST_BSY] is asserted are ignored (the PCI cycle is completed, but no data is transferred).

If an SMBus-defined time out occurs while the host is master of the SMBus, then the host logic attempts to generate a SMBus stop event to clear the cycle. Also, I/O Mapped Power Management +E0h bit [TO_STS] is set to indicate what happened.

### Host-as-Target Controller

The host-as-target controller responds to word-write accesses to either the host address specified by I/O Mapped Power Management +EEh or the snoop address specified by I/O Mapped Power Management +EFh. In either case, if the address matches, then the subsequent data is placed in I/O Mapped Power Management +ECh and I/O Mapped Power Management +EAh. In the case of snoop accesses, the command information is stored in I/O Mapped Power Management +ECh bits[7:0] and the data is stored in I/O Mapped Power Management +EAh bits[15:0]. In the case of

addresses that match the I/O Mapped Power Management +EEh host-as-target address register, then the SMBus master's address is stored in I/O Mapped Power Management +ECh bits[7:1]-if the master has a 7-bit address-or I/O Mapped Power Management +ECh bits[15:1]-if the master has a 10-bit address. After the address match is detected, the target logic waits for the subsequent stop command before setting the appropriate status bit in I/O Mapped Power Management +E0h bits[HSLV_STS, SNP_STS]; however, if a time out occurs during the cycle, after the address match is detected, then the appropriate bit in I/O Mapped Power Management +E0h bits [HSLV_STS, SNP_STS] will not be set.

If one of the target status bits, I/O Mapped Power Management +E0h bits[HSLV_STS, SNP_STS], are set and another access to the host target controller is initiated, then it is not acknowledged (via the first SMBus acknowledge cycle) until the status bit is cleared.

The host-as-target controller operates in the C2, C3, POS, and SOFF modes; it can be used to generate interrupts that wake the system and place it into FON.

SMBALERT. The host controller supports the SMBALERT# signal. If this signal is asserted, then, according to the SMBus specification, software must determine the source by sending a host read cycle to the alert response address, 'b0001_100. If the SMBus host controller detects this address for a read cycle with I/O Mapped Power Management +E2h bit[CYCTYPE] set to receive byte ('b001), then it stores the address returned by the SMBALERT# target in I/O Mapped Power Management +E6h bits[7:0]. If bits[7:1] of this address are 'b1111_0xx, indicating a 10-bit address, then it stores the next byte from the target in I/O Mapped Power Management +E6h bits [15:8].

### 5.14.9 Plug and Play

The AMD-756 controller supports three PNP IRQs, two PNP chip selects, and one PNP DMA select. The registers that specify these are function 3 offsets 44h and 46h. The PNP pins are multiplexed with other functions; the control registers that specify the functions (the GPIO control registers I/O Mapped Power Management +C0h through I/O Mapped Power

Management +D1h) must be set up appropriately for the PNP functions to operate.

### 5.14.10 General-Purpose I/O

The general-purpose I/O pins, GPIO[17:0], can be assigned to be inputs, outputs, interrupt generators, or bus controls. These pins can be programmed to be general-purpose I/O or to be a pre-determined alternate function (see the pin summary section of this document). These pins are all named after one of their alternate functions. There is one register for each pin, I/O Mapped Power Management +[C0h:D1h], that controls the state of each. IRQ status and enables are available for each pin in registers I/O Mapped Power Management +D4h and I/O Mapped Power Management +D8h.

### 5.14.11 General-Purpose I/O Functions

As a general-purpose I/O pin, these pins have the following options:

- Outputs.
  - Can be set high or low.
  - Can be controlled by GPIO output clocks 0 or 1 (see I/O Mapped Power Management +DCh).
- Inputs.
  - Active high or active low programmable.
  - SCI or SMI IRQ capable.
  - Can be latched or not latched.
  - Inputs can be debounce protected.

Here is the basic format for all the general-purpose I/O pins. The input path is not disabled when the output path is enabled or the pin is used for an alternate function. However, when a GPIO function is selected, the alternate function signal is forced to the negated state.

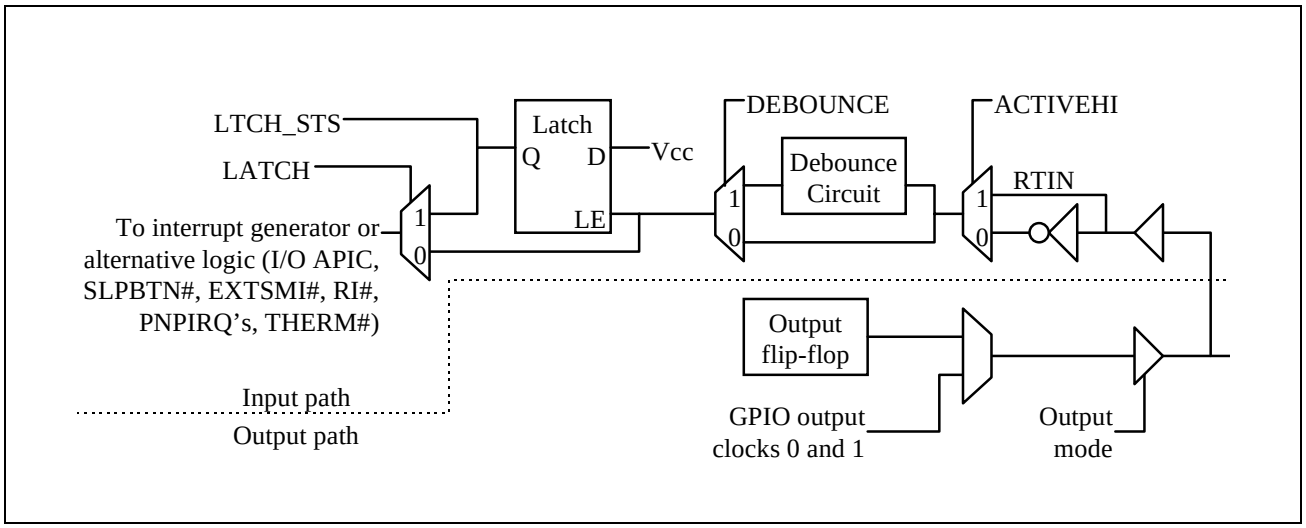


**Figure 31. General-Purpose I/O Logic**

**Debounce**

The debounce circuit consists of a three bit counter clocked off of C256HZ, a 4-millisecond cycle-time clock. The input signal to the circuit is the output of the active-high-active-low multiplexer. The counter is asynchronously cleared by the low state of the input signal and allowed to count when the input signal is high. If the counter increments from 'h0 to 'h4, then the counter stops and the output goes high. Thus, the input signal must be high for 12 to 16 milliseconds before the output signal is asserted.

**GPIO Output Clocks**

There are two GPIO output clocks, numbered 0 and 1. They are specified by I/O Mapped Power Management +DCh. Each output clock includes a 7-bit programmable high time, a 7-bit programmable low time, and the counter can be clocked by one of four frequencies. See Table 22.

The output of the two GPIO output clocks can be selected to drive the output of any of the GPIO pins. They can be used to blink LEDs or for other functions.

**Table 22. GPIO Output Clock Options**

| I/O Mapped Power Management +DCh CLK[1,0]BASE | Base Clock Period | Output High Time Range | Output Low Time Range |
|---|---|---|---|
| 'b00 | 250 microseconds | 250 μ S to 32 ms | 250 μ S to 32 ms |
| 'b01 | 2 milliseconds | 2 mS to 256 ms | 2 mS to 256 ms |
| 'b10 | 16 milliseconds | 16 mS to 2 sec | 16 mS to 2 sec |
| 'b11 | 128 milliseconds | 128 mS to 16.4 sec | 128 mS to 16.4 sec |

Square wave generator. The INTIRQ8# pin, controlled by I/O Mapped Power Management +D0h, includes one additional potential functions, SQWAVE. This is a square wave output, the frequency for which is specified by function 3 offset 4Eh. The square wave generator uses a counter that is clocked by the 32 KHz. clock.

## 5.14.12 AMD-751™ Controller Power Management

The AMD-751 system controller flushes all activity before allowing the stop-grant cycle onto the PCI bus to the AMD-756. After that, the AMD-756 controller can force the AMD-751 device into a clock-stopped mode with DCSTOP#.

The AMD-751 includes self refresh mode so that the host clock can be allowed to stop.

The AMD-751 includes the PCI arbiter disable registers described in the ACPI specification in order to support the C3 low-power state.

## 5.14.13 VDD_SOFT Registers and Logic

The following is a list of the register bits and logic on the VDD_SOFT power plane:

**Registers**

Function 3 offset 48h bits[INTRTC, PWRON]; I/O Mapped Power Management bits [PBOR_STS, RTC_STS, SLPBTN_STS, PWRBTN_STS]; I/O Mapped Power Management +1Ch bits[7:0]; I/O Mapped Power Management +24h bits[USB_RSM_STS]; I/O Mapped Power Management +26h

(all); I/O Mapped Power Management +28h bits [RI_STS, EXTSMI_STS, PME_STS]; I/O Mapped Power Management + C0h; I/O Mapped Power Management + C1h; I/O Mapped Power Management + C3h; I/O Mapped Power Management + CCh; and I/O Mapped Power Management + CEh.

**Logic**

GPIO output blink clocks; the logic associated with pins PWRBTN#, PWRON#, PME#, RI#, SMBUS[C,D], EXTSMI#, SLPBTN#, and PWRGD (including GPIO logic); the state machine to transition between the SOFF and FON states; SMBus target logic; and the USB resume-event detect logic.

This logic is functional and the registers are preserved when in any power state, including SOFF, except MOFF.

### 5.14.14 RTC and CMOS Memory

The real-time clock includes a 32 KHz. oscillator, a clock and calendar timer, an alarm (which generates an interrupt when a specified time occurs), and 256 bytes of non-volatile RAM. It is register compatible with the real-time clock found in the original AT design (which used the MC146818). Also, it is updated to meet ACPI's real-time clock requirements.

**Power**

The real-time clock includes its own power plane, VDD_RTC, which is expected to be powered by an external 3.3-volt lithium battery. When the VDD_SOFT plane is powered, the system is required to power the real-time clock circuitry with that plane such that the VDD_RTC leakage current is less than 0.5 microamps. It is a requirement that CPU accesses to the real-time clock logic be disabled when the VDD_SOFT plane is not powered. It is also a requirement that the VDD_SOFT plane be capable of being arbitrarily powered up and down without the possibility of disturbing the RTC logic or causing erroneous writes to the real-time clock register set.

The VDD_RTC power plane is required to not be damaged when it is not powered while the other Cobra power planes are powered. Conversely, VDD_RTC is allowed to be powered while any other planes are not powered.

When operating on the battery's supply, the RTC section is required to consume no more than 0.5 microamps of current

(this gives 11.4 years of life with a typical 50 milliamp-hour battery).

**Oscillator**

The real-time clock includes a 32.768 KHz. oscillator that is used to keep time. The oscillator circuit is designed to be accurate to within 10 parts per million over the external temperature and capacitance ranges. This provides for a time loss of less than 30 seconds per month. Switching the real-time clock power source between VDD_RTC and VDD_SOFT does not affect the operation or frequency of the oscillator. This oscillator is required to be present regardless of whether the internal or an external real-time clock is used.

**VDD_SOFT Reset**

The RTC logic generates a reset signal to VDD_SOFT power plane when that plane is detected to power up. It is called RST_SOFT. RST_SOFT is required to last between 25 and 50 milliseconds after VDD_SOFT is detected to be greater than 2.5 volts. This reset is required to occur when the VDD_RTC plane is powered as well; the RTC battery may be dead or the external RTC may be being used so the RTC power plane would always power up with the VDD_SOFT plane.

**Self Reset**

There is no external reset signal for the logic powered by VDD_RTC. The circuitry generates its own internal reset signal when VDD_RTC power is applied to the IC such that it is guaranteed to power up in a functional state. The application of VDD_RTC power is allowed to be very noisy—rapidly going up and down—as a battery is being inserted into a socket. The self reset signal is allowed to be up to two seconds long. The self reset is applied to the reset for the VDD_SOFT plane, if that plane is powered (i.e., if the battery is missing or dead and VDD_SOFT is powered, then the logic is capable of resetting both planes properly).

**External Real-Time Clock**

This logic is also required to be able to operate properly when an external real-time clock is used. In this situation, no battery is connected to the VDD_RTC plane. Instead, the VDD_RTC plane is connected to VDD_SOFT. After PWRGD reset, the system detects that an external real-time clock is present and selects the pins for this function. See function 3 offset 48h bit[INTRTC].

**New ACPI Requirements**

Section 4.7.2.4 of the revision 1.0 ACPI specification describes the new requirements for the real-time clock. Some of these are

listed as optional. However, the AMD-756 controller implements all of these features.

The day alarm, month alarm, and centenary value are stored in CMOS RAM address space.

**Table 23. RTC CMOS Addresses**

| CMOS RAM Offset | Function | Range for Binary Mode | Range for BCD Mode |
|---|---|---|---|
| 7Dh | Date alarm | 0h-1Fh | 01h-31h |
| 7Eh | Month alarm | 01h-0Ch | 01h-12h |
| 7Fh | Century field | 13h-63h | 19h-99h |

**PRDY**

The RTC's counter chain is frozen from counting, if enabled to do so in function 3 offset 4Ch bit[RTC_DIS], when PRDY is asserted.

A complete CMOS table can be found in chapter 7.

## 5.15 Universal Serial Bus Controller (USBC)

**Introduction**

The USB Controller, USBC, is a complete Host Controller as defined by the Universal Serial Bus, USB, 1.0 specification and the OpenHCI standard developed by Compaq, Microsoft, and National Semiconductor. OpenHCI specifies the interface between the Host Controller Driver, HCD, and the Host Controller, HC. USBC contains an integrated Root Hub with 4 USB ports, PCI interface, and a Host Controller core. Keyboard and Mouse legacy support are also included for DOS compatibility with USB devices.

**PCI Interface**

The OpenHCI Host Controller is connected to the system by the PCI bus. The design requires both master and target bus operations. As a master, the USBC runs cycles on the PCI bus to access Endpoint Descriptors, EDs, and Transfer Descriptors, TDs, as well as transferring data between memory and the local data buffer. the USBC is a PCI target when it decodes cycles to its internal PCI Configuration Registers or to its internal PCI Memory Mapped I/O Registers. The following table identifies the cycle types initiated by the USBC as a master and accepted by the USBC as a target.

**Table 24. PCI Commands Supported by the USBC**

| C/BE[3:0]# | Command Type | Target Support | Master Support |
|---|---|---|---|
| 0010 | I/O Read | Yes | No |
| 0011 | I/O Write | Yes | No |
| 0110 | Memory Read | Yes | Yes |
| 0111 | Memory Write | Yes | Yes |
| 1010 | Configuration Read | Yes | No |
| 1011 | Configuration Write | Yes | No |
| 1100 | Memory Read Multiple | Yes* | Yes |
| 1110 | Memory Read Line | Yes* | Yes |
| 1111 | Memory Write and Invalidate | Yes* | Yes |

*Note: *Only a single doubleword is accepted after which the cycle is retried.*

### PCI Master

For write cycles, the command issued is Memory Write unless the requested transfer is exactly one cache line and the cache line size is set to 32 bytes, in which case the command is Memory Write and Invalidate. The byte enables are synthesized for the first and last doublewords using the begin address and number of bytes to calculate them. All doublewords between the first and last have all byte enables asserted.

For read cycles, the command issued depends on the transfer size and the value of the cache line size within the configuration space. Transfers that cross a cache line boundary use the Memory Read Multiple command. Transfers that cross a quadword boundary use the Memory Read Line command. All other transfers use the Memory Read command.

The address used in the master cycle is stored in a register at the initial request of the cycle. When the PCI cycle commences, the upper 30 bits of the register are sent to the PCI Bus, with the lower two bits always being 0h. After each successful PCI transfer the address register is incremented by the number of bytes transferred. Thus, in case of a disconnect, retry, or time-out of the cycle the address is ready for the resumption of the cycle.

### PCI Target

The USBC Target State Machine and interface logic support fast back-to-back cycles. The USBC does not require an idle cycle between a write to another target and a USBC target access.

### Device Selection

The USBC uses medium decode timing when asserting DEVSEL# to claim a PCI transaction whose address

corresponds to the base address and offset of an internal address. When the USBC asserts DEVSEL#, it does not negate DEVSEL# until the transaction ends on the bus. The end of the bus transaction is indicated by FRAME# sampled negated, IRDY# sampled asserted, and either TRDY# or STOP# sampled asserted.

### Write Cycles

The USBC interprets command codes C/BE[3:0]# = 0111 (memory write) and 1111 (memory write and invalidate) as write cycles. The USBC does not perform write gathering for PCI write cycles; all writes are posted and strong write ordering is maintained.

While a posted write cycle is still in process, subsequent target read or write cycles are either held off with wait states until the posted write completes, or the cycle is "retried" if it requires more than 16 waitstates.

If a master attempts to burst more than one doubleword to the USBC in a cycle, the USBC disconnects the cycle after the first doubleword is written.

The PCI Specification allows any contiguous or non-contiguous combination of byte enables. As a target, the USBC ignores the byte enables for memory write cycles. All memory-mapped registers within the USBC are defined as doublewords.

### Read Cycles

The USBC interprets command codes C/BE[3:0]# = 0110 (memory read), 1100 (memory read multiple), and 1110 (memory read line) as a read cycle. Read cycles are not posted. If the cycle can be accepted, the USBC claims the cycle by asserting DEVSEL#, and then holds the PCI bus in wait states by holding TRDY# negated until the data is ready.

If a master attempts to burst more than one doubleword from the USBC in a cycle, the USBC disconnects the cycle after the first doubleword is read.

### Configuration Cycles

The USBC interprets command codes C/BE[3:0]# = 1010 and 1011 as configuration read and configuration write cycles respectively. The USBC can accept any combination of byte enables for configuration write cycles.

### Locked Cycles

The USBC does not implement lock cycles.

**Abort Cycles**

The USBC PCI Target state machine performs PCI target abort cycles for any I/O cycle in which the byte enables do not correspond to the low two address bits. Generation of SERR#

The USBC asserts SERR# internally when it detects a PCI address parity error and it is the agent receiving data, if parity is enabled. SERR# is also asserted whenever the SERR enable bit (bit 8 of the PCI Command register) is enabled and a target abort is received, a target abort is signaled, or master abort is signaled. This SERR# internal signal is used to generate an NMI.

**Generation of STOP#**

The USBC only asserts STOP# to retry a transaction when it is the target, never to abort it.

## 5.15.1 USBC Miscellaneous Functions

**Power Switching**

A root hub controls power to the downstream ports. The USBC implements global power control, which means that all port's power status, PortPowerStatus, is controlled by Set/ClearGlobalPower commands. Individual port power switching as well as no power switching modes are also implemented, but individual port power switching signals are not brought out of the USBC. The table below shows the power switching configurations in the descriptor registers.

The output pin PWREN controls global power switching.

**Table 25. Power Switching Mode**

| Mode | No Power Switching | Power Switching Mode | Port Power Control Mask |
|---|---|---|---|
| No Switching | 1 | - | - |
| Global Switching | 0 | 0 | - |
| Individual Port Switching | 0 | 1 | 1 |
| Global Switch in Individual Switching Mode | 0 | 1 | 0 |

**Over-Current Protection**

Over-Current is reported on a global basis. Input pin OVRCUR is read directly through OverCurrentIndicator in *HcRhStatus*. Over-current mode is configured in *HcRhDescriptorA* by NoOverCurrentProtection and OverCurrentProtectionMode. When in individual over-current mode, OVRCUR status is

reported through the PortOverCurrentIndicator in *HcRhPortStatus* for test purposes.

When an over-current condition occurs OverCurrentIndicatorChange is set and power is disabled. If port power is not switched (NoPowerSwitch), the port power status is not affected by an over-current condition.

### Legacy Keyboard and Mouse Support

To support applications and drivers in non-USB aware environments (e.g. DOS) the USBC provides hardware support for the emulation of a PS/2 keyboard and/or mouse by their USB equivalents. When keyboard emulation is enabled, the USBC intercepts I/O accesses to port 60 and port 64. The keyboard controller must use subtractive decode.

### Keyboard/Mouse Input

When a successful transfer of data has occurred from the keyboard, the Transfer Descriptor is moved to the Done Queue by the USBC. At the beginning of the next frame when the interrupt associated with the transfer completion is to be signaled, an interrupt is generated. System software should be designed to set the InterruptRouting bit in *HcControl* to 1, so that the interrupts result in an SMI. On receipt of the SMI the emulation software removes the Transfer Descriptor from the Done Queue, clears the USBC IRQ, and translates the keyboard/mouse data into an equivalent PS/2-compatible sequence for the application software. For each byte of PS/2 compatible data that is to be presented to the applications software, the emulation code writes to the *HceOutput* register. The emulation code then sets the appropriate bits in the *HceStatus* register (normally sets OutputFull for keyboard data and OutputFull plus AuxOutputFull for mouse data.) If keyboard/mouse interrupts are enabled, setting the *HceStatus* register bits causes generation of an IRQ1 for keyboard data and IRQ12 for mouse data. The emulation code then exits and waits for the next emulation interrupt.

When the host CPU exits from SMM, it can service the pending IRQ1/IRQ12. This normally results in a read from I/O port 60h. When I/O port 60h is read, the USBC intercepts the access and returns the current contents of *HceOutput*. The USBC then also clears the OutputFull bit in *HceStatus*, and de-asserts IRQ1/IRQ12.

If the emulation software has multiple characters to send to the application software, it sets the CharacterPending bit in the

*HceControl* register. This causes the USBC to generate an emulation interrupt on the next frame boundary after the application has read from port 60h (*HceOutput*.)

**Keyboard Output**

Keyboard output is indicated by application software writing data to either I/O address 60h or 64h. Upon a write to either address the USBC will capture the data in the *HceInput* register and, except in the case of a Gate A20 sequence, update the *HceStatus* register's InputFull and CmdData bits. When the InputFull bit is set, an emulation interrupt is generated.

Upon receipt of the emulation interrupt, the emulation software will read *HceControl* and *HceStatus* to determine the cause of the emulation interrupt and perform the operation indicated by the data.

**Emulation Interrupts**

Emulation interrupts are caused by reads and writes of the emulation registers. Interrupts generated by the emulation hardware are steered by the USBC to either an SMI or the standard interrupt. Steering is determined by the setting of the InterruptRouting bit in the *HcControl* Register.

Emulation interrupts for data coming from the keyboard/mouse are generated on frame boundaries. At the beginning of each frame, the conditions which define asynchronous emulation interrupt are checked and, if an interrupt condition exists, the emulation interrupt is signaled to the host at the same time as the interrupts coming from normal USB processing. This has the effect of reducing the number of SMIs that are generated for legacy input to no more than 1,000 per second. Although still somewhat large, this number of interrupts is less than the number that could be generated if emulation interrupts were not merged with the normal USBC interrupts.

**Mixed Environment Interrupts**

A mixed environment is one in which a USB device and an PS/2 device are supported simultaneously (e.g., a USB keyboard and a PS/2 mouse.) The mixed environment is supported by allowing the emulation software to control the PS/2 interface. Control of this interface includes capturing I/O accesses to port 60h and 64h and also includes capture of interrupts from the PS/2 keyboard controller. IRQ1 and IRQ12 from the legacy keyboard controller are routed through the USBC. When ExternalIRQEn in *HceControl* is set, IRQ1 and IRQ12 from the legacy keyboard controller are blocked by the USBC and an

emulation interrupt is generated instead. This allows the emulation software to capture data coming from the legacy controller and present it to the application through the emulated interface.

### Gate A20 Sequence

The Gate A20 sequence often occurs in DOS applications, usually to enable A20. To reduce the number of SMIs caused by the Gate A20 sequence, the USBC only generates an SMI if the A20 sequence changes the state of Gate A20.

The Gate A20 sequence is initiated with a write of D1h to port 64h. On detecting this write, the USBC will set the GateA20Sequence bit in *HceControl*. It will capture the data byte in *HceInput* but will not set InputFull bit in *HceStatus*. When GateA20Sequence is set, a write of a value to I/O port 60h that has bit 1 set to a value different than A20State in *HceControl* will cause set InputFull and cause an interrupt. An SMI with InputFull and GateA20Sequence both set indicates that the application is trying to change the setting of Gate A20 on the keyboard controller. However, when GateA20Sequence is set and a write of a value to I/O port 60h that has bit 1 set to the same value as A20State in *HceControl* is detected then no interrupt will occur.

As mentioned above, a write to 64h of any value other than D1h will cause GateA20Sequence to be cleared. If GateA20Sequence is active and a value of FFh is written to port 64h, GateA20Sequence is cleared but InputFull is not set. A write of any value other than D1h or FFh will cause InputFull to be set which will then cause an SMI. A write of FFh to port 64h when GateA20Sequence is not set will cause InputFull to be set.

### Emulation Registers

Four emulation registers are used to provide the legacy support. Each of these registers is located on a 32-bit boundary.

**Table 26. Host Controller Registers**

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

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

**Table 27. Register Side Effects**

| I/O Address | Cycle Type | Register Contents Accessed/Modified | Side Effects |
|---|---|---|---|
| 60h | IN | HceOutput | IN from port 60h will set **OutputFull** in *HceStatus* to 0 |
| 60h | OUT | HceInput | OUT to port 60h will set **InputFull** to 1 and **CmdData** to 0 in *HceStatus*. |
| 64h | IN | HceStatus | IN from port 64h returns current value of *HceStatus* with no other side effect |
| 64h | OUT | HceInput | OUT to port 64h will set **InputFull** and **CmdData** in *HceStatus* to 1. |

**HceInput Register**

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

**HceOutput Register**

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

**HceStatus Register**

The contents of the *HceStatus* Register is returned on an I/O Read of port 64h when emulation is enabled. Reads and writes of port 60h and writes to port 64h can cause changes in this register. Emulation software can directly access this register through its memory address in the USBC register space. Access of this register through its memory address produces no side effects.

## 5.15.2 System Management

**Activity Monitoring**

USB activity is monitored by the System Management block. Four signals are generated by the USBC to allow this monitoring: Bulk activity, Control activity, Isochronous activity, and Interrupt activity. Bulk activity, Control activity, and Isochronous activity will pulse every time a USB transaction of

the specific type occurs. Interrupt activity will pulse only when Interrupt data is transmitted or received and an ACK is returned. This functionality is added to the Interrupt activity pulse to eliminate the possibility of the USB appearing to be active when no USB data is being transmitted.

### USBC Signaled Suspend and Resume

During the Suspend state the USBC's list processing and SOF Token generation are disabled. However, the USBC's remote wakeup logic must monitor USB wakeup activity. The FrameNumber field of *HcFmNumber* does not increment while the USBC is in the Suspend state.

Suspend is entered following a software reset or by command from the Host Controller Driver. While in Suspend, the USBC can force a transition to the Resume state due to a remote wakeup condition. This transition may conflict with the Host Controller Driver initiating a transition to the Reset state. If this situation occurs, the HCD-initiated transition to Reset has priority. The Host Controller Driver must wait 5 ms after transitioning to Suspend before transitioning to the Resume state. Likewise, the USBC must wait 5 ms after entering Suspend before generating a local wakeup event and forcing a transition to Resume. Following a software reset, the Host Controller Driver may cause a transition to the Operational state if the transition occurs no more than 1 ms from the transition into Suspend. If the 1-ms period is violated, it is possible that devices on the bus will go into Suspend.

When in the Resume state, the USBC forces resume signaling on the bus. While in Resume, the USBC is responsible for propagating the USB Resume signal to downstream ports as specified in the USB Specification. The USBC's list processing and SOF Token generation are disabled while in Resume. In addition, the FrameNumber field of *HcFmNumber* does not increment while the USBC is in the Resume state.

Resume is only entered from Suspend. The transition to Resume can be initiated by the Host Controller Driver or by a USB remote wakeup signaled by the Root Hub. The USBC is responsible for resolving state transition conflicts between the hardware wakeup and Host Controller Driver initiated state transitions.

The USB 12 MHz clocks are disabled in the Suspend state. In this state no activity is enabled which means the USB clocks

are unused. Once suspended and all activity requiring a clock completes the USBC's clock generator is suspended.

The following events will prevent the stoppage or re-enable the clock:

- Host Controller Driver forces an exit of the Suspend state.
- A ResumeDetected event interrupt is generated.
- A LS EOP is active.
- A port connect/disconnect event transition requires the clock to be timed or is in progress.
- A Port Reset is active on any port.

### Selective Suspend and Resume

A port can be selectively suspended by issuing a SetPortSuspend command which sets PortSuspendStatus. While suspended the port does not propagate any downstream traffic. The port can be awakened by any of the following methods:

- A ClearPortSuspend command
- An upstream J to K transition.
- A J to SE0 transition which results in a disconnect event (no resume).

When the port is resumed, the port drives the resume signal downstream for 20 ms followed by a LS EOP. The port waits an additional 3 ms during which it propagates all traffic and then sets PortSuspendStatusChange and clears PortSuspendStatus.

A disconnect event at a selectively suspended port does not generate a resume pulse. It sets ConnectStatusChange and clears PortSuspendStatus.

If the USBC is globally suspended, while the port is selectively suspended, the port does not respond to the USBC's transition to global resume. However, a remote wakeup condition at the port will force a resume transition.

### Device Signaled Resume

A remote wakeup generates a ResumeDetected interrupt and a transition to the Resume state. A remote wakeup is defined as follows:

- A K-state (resume) at an enabled port.
- Port connect/disconnect detection and RemoteWakeup-Enable is set.

- A port resume is in progress at a selectively suspended port.

Upstream Resume or connect/disconnect wakeup events are controlled at both the port and global level. The USBC combines a ports ResumeDetected and ConnectStatusChange (if enabled) and does three things: 1) Generates a ResumeDetected interrupt. 2) Forces a transition to the Resume state. 3) Resumes any remaining enabled ports.

**USB Interrupts**

The interrupt output of the USB block is functionally ORed into the PCI input interrupt, PIRQD#. This result is then routed to an ISA interrupt via function 0, offset 56h.

## 5.16 Programmable Interrupt Controller (PIC)

**Introduction**

The Programmable Interrupt Controller (PIC) functions as an overall manager in an Interrupt driven system environment. It accepts request from the peripheral equipment, determines which of the incoming request is of the highest importance (priority), ascertains whether the incoming request has a higher priority value than the level currently being serviced, and issues an interrupt to the CPU based on its determination. Many of the features described here are not used in the normal PC environment. However, these features are useful if the PC is used as an embedded controller, or in industrial applications.

Each peripheral device usually has a special program that is associated with its specific operational requirements; this is referred to as a Interrupt service routine (ISR). The PIC after issuing an interrupt to the processor will return a vector during the interrupt acknowledge cycle INTACK. This vector “points to the vector table which contains the address of the actual ISR.

The PIC is equivalent to the original 8259A. It manages eight levels or request. The AMD-756 controller contains two PICs for a total of 16 interrupts. Some of the channels are programmed by the BIOS for the standard peripherals. The other channels are programmed by the specific device drivers as they are loaded. The PICs can be configured in fixed or rotating priority modes. In a PC they are normally set to fixed priority. If rotating priority is selected the channel being

serviced becomes the lowest priority. This prevents one channel from monopolizing the CPU. The priority mode can be changed dynamically.

**The Interrupt Request Register (IRR) and In-Service Register (ISR)**

The interrupts at the IR input lines are handled by the two registers in cascade, the Interrupt request register and the in-service register. The IRR is used to store all the interrupt levels which are requesting service, and the ISR is used to store all the interrupt levels being serviced.

**Priority Resolver**

This logic block determines the priority of the various bits set in the IRR. The highest priority is selected and strobed into the corresponding bit of the ISR during the INTACK cycle.

**Interrupt Mask Register**

The IMR stores the bits which mask the corresponding interrupt bit in the IRR. The mask bits will not prevent an IRR bit from being set. It does prevent that IRR bit from generating an interrupt to the processor. If an IRR bit is set and the channel mask bit is cleared, an interrupt will occur immediately.

**Read/Write Control Logic**

This block interfaces with the PCI bus control signals. It also contains the operation command word (OCW) and the initialization command word (ICW) registers which store the various control formats.

**The Cascade Buffer/Comparator**

This function block stores and compares the ID's of all the PIC's used in the system. The associated three signals (CAS[2:0]) are outputs when the PIC is used as a master and are inputs when the PIC is used as a target. As a master, the PIC sends the ID of the interrupting target device onto the internal CAS[2:0] lines. The target will then send it's preprogrammed subroutine address onto the selected data bus during the two consecutive INTA pulses. (See section "Cascading the PIC".)

**Interrupt Sequence**

The powerful features of the PIC in a computer are it's programmability and the interrupt routine addressing capability. The latter allows direct or indirect jumping to the specific interrupt routine requested without any polling of the interrupting devices. The normal sequence of events during an interrupt:

1. One or more of the Interrupt Request lines transition setting the appropriate IRR bit(s).
2. The PIC evaluates these request and asserts INT to the CPU, if enabled and the channel is not masked.
3. The CPU responds with two interrupt acknowledge cycles.
4. On the first cycle the state of the IRR bits is frozen for priority resolution. The highest priority ISR bit is set, and the corresponding IRR bit is reset. The master PIC also issues the interrupt code on the CAS[2:0] signals at the end of the first INTACK cycle.
5. On the second INTACK cycle the PIC returns an 8 bit vector (pointer). This vector serves as an index into the Interrupt vector table.
6. In the AEOI mode, the ISR bit is reset at the end of the second INTACK cycle. Otherwise the ISR remains set until a EOI command is issued at the end of the interrupt subroutine.

If no interrupt request is present during the first INTACK cycle, the request was to short (a glitch). The PIC will issue an interrupt level 7.

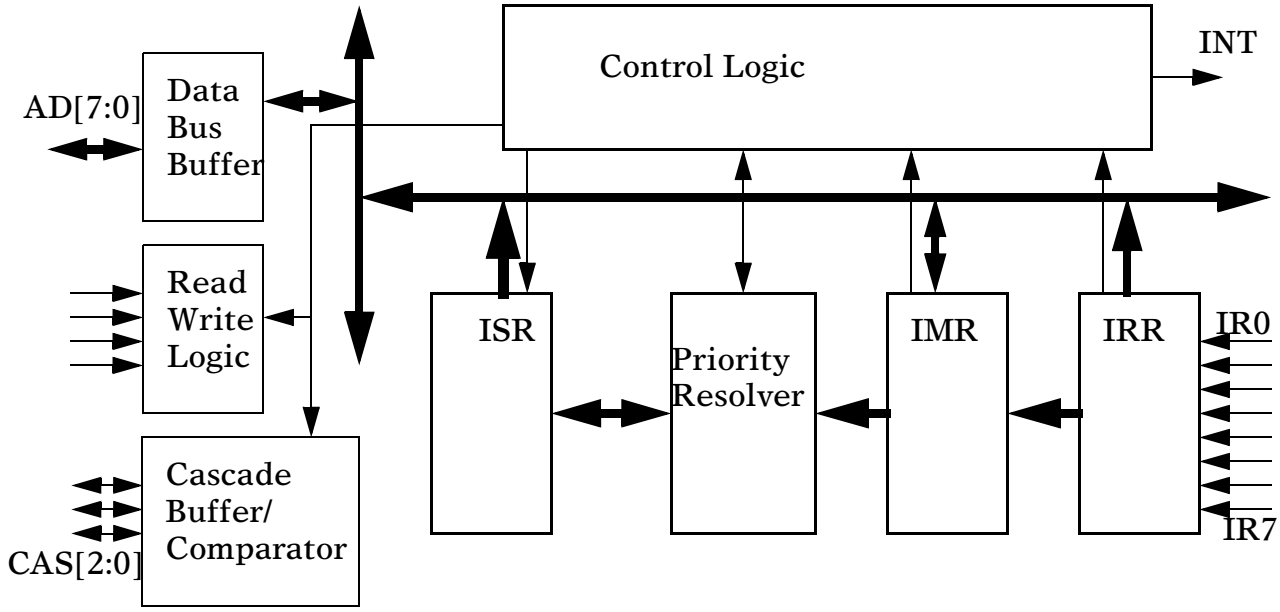


**Figure 32. FPIC Block Diagram**

**Table 28. Interrupt Vector Byte Contents**

| | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| IR7 | T7 | T6 | T5 | T4 | T3 | 1 | 1 | 1 |
| IR6 | T7 | T6 | T5 | T4 | T3 | 1 | 1 | 0 |
| IR5 | T7 | T6 | T5 | T4 | T3 | 1 | 0 | 1 |
| IR4 | T7 | T6 | T5 | T4 | T3 | 1 | 0 | 0 |
| IR3 | T7 | T6 | T5 | T4 | T3 | 0 | 1 | 1 |
| IR2 | T7 | T6 | T5 | T4 | T3 | 0 | 1 | 0 |
| IR1 | T7 | T6 | T5 | T4 | T3 | 0 | 0 | 1 |
| IR0 | T7 | T6 | T5 | T4 | T3 | 0 | 0 | 0 |

### 5.16.1 Programming

The PIC accepts two types of command words.

**Initialization Command Words (ICWs)**

These occur prior to normal operation. Each PIC must be brought to a starting point. Whenever a write to ICW1 occurs the following sequence begins:

- The edge sensitive circuit is reset. This means a low-to-high transition must occur to set the IRR bits.
- The Interrupt mask register is cleared.
- IR7 is assigned priority 7.
- The target mode address is set to 7.
- Special Mask mode is cleared and Status read points to the IRR.
- If bit IC4 in ICW1 is 0. Then all functions in ICW4 are set to 0. Non-buffered mode, no Auto-EOI.

**Operational Command Words (OCWs)**

These command the PIC to operate in a particular mode. They can be written anytime after the PIC has been initialized.

- Fully nested
- Rotating Priority
- Special Masked mode
- Polled mode

See Section 7.4.4 on page 195 for ICW bit definitions.

**ICW1 Vector and Control**

This register contains the three least significant bits of the Interrupt vector based on the interrupt level in service.

LTIM = 1 selects level-triggered mode.
0 = edge-triggered mode.

ADI = 1 interval equals 4
0 interval equals 8

SNGL = 1 means one PIC in the system; 0 means more than one.

IC4 = if set then ICW4 must be read. 0 means ICW4 is not needed

**ICW2 interrupt Vector** The register contains the 5 most significant bits T[7:3] of the interrupt vector.

**ICW3** ICW3 is used when there are Multiple PICs.

a) In the master mode a 1 is set for each target in the system.

b) In the target mode bits 2:0 identify the target

**ICW4** SFNM = 1 selects the Special fully nested mode

BUF =1 selects the buffered mode

M/S =1 means a master; 0 means a target if BUF=1. If BUF = 0 this bit has no meaning.

AEOI =1 selects the automatic End Of Interrupt mode.

UPM =1 selects the 8086 mode of operation.

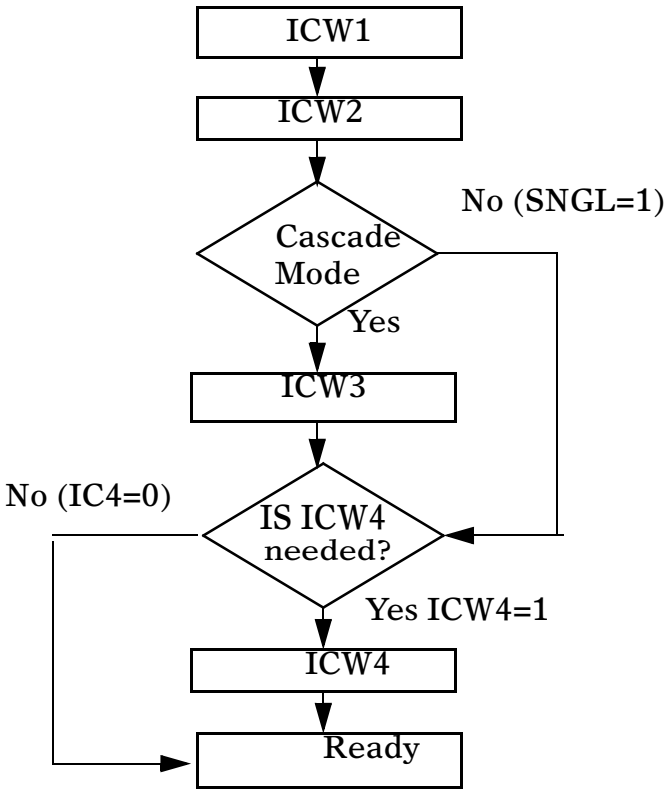


**Figure 33. PIC Initialization Sequence**

**OCW1** Interrupt mask register. Each bit in this register will mask the corresponding IRR bit.

**OCW2** OCW2 contains R,SL,EOI. These three bits control the priority rotation, and end of interrupt modes.

**OCW3** Read register command and mask mode selection

ESMM = 1 enables Special Mask mode

SMM = 1 and ESMM = 1 allows the PIC to enter Special Mask mode. if SMM = 0 the PIC reverts to normal masked mode. If ESMM is 0 this bit has no effect.

**Fully-Nested Mode** This mode is entered after initialization unless another mode is programmed. The interrupt request are ordered in priority from 0 to 7 (0 is the highest). When an interrupt is acknowledged, the highest priority request is determined and its vector placed on the bus. Additionally a bit in the interrupt service register is set. This bit remains set until the processor issues an EOI command before returning from the ISR., or if AEOI is set, it is cleared at the end of the second INTACK cycle. While the IS bit is set, all further interrupts of the same or lower priority are inhibited, while higher levels will generate an interrupt. These interrupts will only be acknowledged if the CPU has re-enabled its internal Interrupt enable bit. In this fashion the programmer can control if he wishes to allow nested interrupts.

After the initialization sequence, IR0 has the highest priority and IR7 has the lowest. Priorities can be changed if rotating priority mode is selected.

**End of interrupt (EOI)** The IS bit can be reset either automatically following the trailing edge of the last INTACK cycle (if AEOI in ICW1 is set), or by a reset EOI command from the CPU prior to returning from the ISR. An EOI command must be issued twice when in cascade mode, once for the master and once for the target.

There are two forms of EOI commands: Specific and non-specific. When the PIC operates in modes which preserve the fully nested structure, it can determine which IS bit to reset on EOI. When a non-specific EOI command is issued, the PIC will automatically reset the highest IS of those that are set., since in the fully nested mode, the highest IS level has to have been the

one serviced. A non-specific EOI can be issued when OCW2 (EOI=1, SL=0, and R=0).

When a mode is used which may disturb the fully nested structure, the PIC may no longer be able to determine the last level acknowledged. In this case, a specific EOI must be issued which includes as part of the command the IS level to reset. A specific EOI can be issued when OCW2 (EOI=1, SL=1,R=0, and L0-2 is the binary level of the IS bit to reset.

Take note that an IS bit that is masked by an IMR bit will not be cleared by a non-specific EOI if the PIC is in the special masked mode.

### Automatic End of Interrupt (AEOI) Mode

If AEOI=1 in ICW4, then the PIC will operate in the AEOI mode. In this mode the PIC automatically performs a non-specific EOI at the end of the second INTACK cycle. Note, from a system perspective this mode should only be used when a nested multilevel interrupt structure is not required with in a single PIC.

AEOI can only be used in a master PIC not in the target.

### Priority Rotation

In some applications there are a number of interrupting devices. It may be desirable to give each device a fair share of the processor's time. When the PIC is in the rotating mode a channel becomes the lowest priority after it is serviced. Thus a device requesting service will have to wait, in the worst case, until each of the 7 other devices get serviced.

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, and EOI=0).

### Specific Priority

The programmer can change priorities by programming the bottom priority and thus fixing the other priorities relative to it. i.e., if IR5 is programmed as the bottom priority, then IR6 will be the highest (IR5, IR4, IR3, IR2, IR1, IR0, IR7, IR6).

The set priority command is issued in OCW2 (R=1, SL=1), and L[2:0] is the binary priority level code of the bottom choice.

In this mode internal status is updated by software control during OCW2 writes. However, these updates are independent

of the EOI command. Priority changes can be executed during an EOI command by using the Rotate on Specific EOI command in OCWR2 (R=1, SL=1, EOI=1) and L[2:0] = IR level to receive the lowest priority.

### Interrupt Mask

Each interrupt request input can be masked individually by the interrupt mask register (IMR) programmed through OCW1. Each bit in the IMR mask the corresponding bit in the IRR if set to 1. Masking one IR channel does not affect the other channels.

### Special Mask Mode

Some applications may require an interrupt service routine to dynamically alter the system priority structure during its execution. For example, the ISR may wish to inhibit lower priority request for a portion of its execution but enable them for another portion. Care must be taken to avoid inhibiting interrupts permanently. For example if an interrupt request is acknowledged and an EOI command did not reset its IS bit the PIC will inhibit all lower priority request with no easy way to re-enable them.

This is where the Special Mask mode comes in. In the Special masked mode, when a mask bit is set 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.

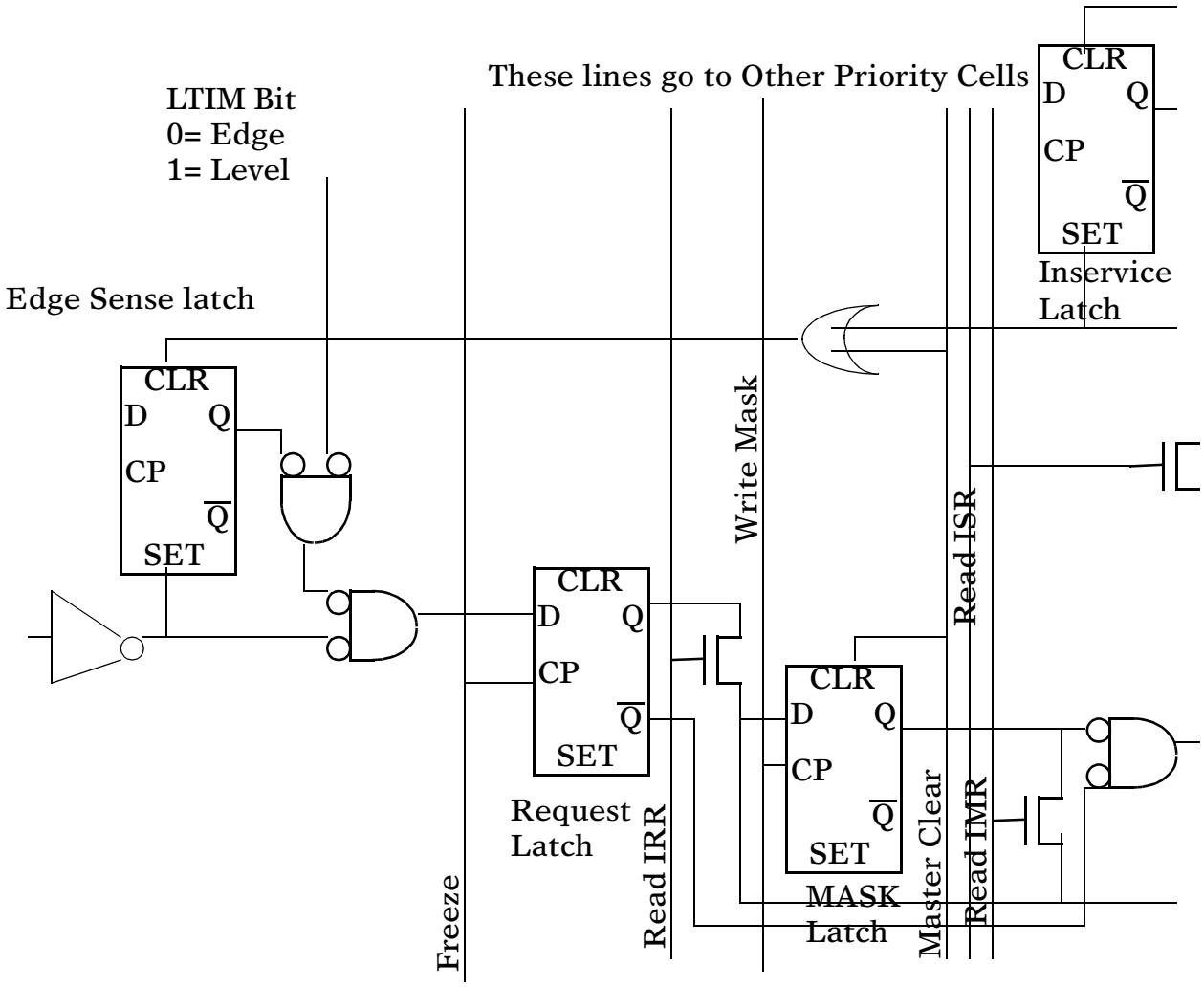


**Figure 34. Priority Cell**

**Poll Command**

In this mode the INT output is issued by setting P=1 in OCW3. The PIC treats the next read as an interrupt acknowledge, set the appropriate IS bit if there is a request pending and reads the priority level. The interrupt level is frozen between the write and the read.

This mode is useful if there is a routine command common to several levels so that the INTACK sequence is not needed. Another application is to use the poll mode to expand the number of priority levels.

**Table 29. Polling Status Format**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| I | - | - | - | - | W2 | W1 | W0 |

I equals 1 if there is an interrupt

W[2:0] binary code of the highest priority requesting service.

### Reading the PIC Status

The input status of several internal registers can be read to give the programmer complete information on the status of the devices. The following registers can be read via OCW3: IRR, ISR, and IMR (OCW1).

To read the IRR a read register command is written to OCW3 (RR=1, RIS=0).

To read the ISR a read register command is written to OCW3 (RR=1, RIS=1).

There is no need to write an OCW3 before every status read operation, as long as the status read corresponds with the previous one; i.e., the PIC remembers whether the IRR or ISR was previously selected by OCW3. This is not true when poll mode is used.

After initialization the PIC points to the IRR.

To read the IMR, no OCW3 write is needed. Just read OCW1.

Polling will override status reads when P=1 and RR=1 in OCW3.

### Edge-Triggered and Level-Triggered Modes

This mode is programmed using bit 3 in ICW1.

If LTIM =0, an interrupt is recognized by a Low-to-High transition on an IR input. The IR input must remain high until the INTACK is received. The IR input can remain high without generating another input.

If LTIM =1, an interrupt is recognized by a High level on an IR input and there is no need for a level change. The IR input must be removed before the EOI command is issued to prevent generating a second interrupt. The priority cell diagram shows a conceptual circuit of the level sensitive and edge sensitive circuitry. Note that the request is a transparent latch.

In both edge and level triggered modes, the IR inputs must remain high until after the falling edge of the first INTACK cycles. If the IR input goes low before this time, a default IR7 vector will be returned. This can be used as a safeguard for detecting interrupts caused by spurious noise glitches on the IR pins. To implement this feature the IR7 routine is used for clean up simply execute a return instruction, thus ignoring the interrupt. If IR7 is used for a device a default IR7 can still be detected by reading the ISR. If high it was a real interrupt. If low it was noise.

**Special Full-Nested Mode**

This mode is used in the case of a large system where cascading is used and the priority has to be preserved in each target. In this case the master is programmed to the fully nested mode (ICW4). This mode is similar to the normal nested mode with the following exceptions:

1. When an Interrupt request from a certain target is in service, this target is not locked out from the master's priority logic, and further interrupt request from higher priority IR's within the target will be recognized by the master and initiate interrupts to the processor. (In normal nested mode, a target is masked out when its request is in service and no higher request from the target can be serviced.)
2. When exiting the interrupt service routine, the software must check whether the interrupt serviced was the only one from that target by sending a non-specific EOI command to the target 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 also. If not, no EOI should be sent.

**Buffered Mode**

This mode of the original 82C59A is not supported.

**Cascade Mode**

The original 82C59A could be interconnected to support up to 8 targets for a total of 64 priority levels. The PIC only supports up to 16 priority levels.

The Master PIC controls the target through the 3 internal cascade signals. Since these signals are not available on external pins no further cascading is possible. Each PIC in the 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 target.

# 5.17 I/O Advanced Programmable Interrupt Controller (IOAPIC)

## Introduction

The AMD-756 peripheral bus controller incorporates an I/O advanced programmable interrupt controller (APIC) in addition to the PIC. The APIC architecture is one of several mechanisms required for improving the multiprocessor performance of AMD Athlon processors. The APIC allows interrupt-related traffic to be offloaded from the memory bus, which makes memory more available for processor use. In addition, the APIC dedicated interrupt bus can provide an advantage in single processor systems, by reducing the interrupt latency associated with the propagation of interrupt-acknowledge cycles over multiple buses.

The APIC controller architecture consists of two components—a local APIC and an IOAPIC. Figure shows a system-level implementation of the APIC components. The local APIC and IOAPIC communicate over the interrupt message bus (IMB).

## Local APIC

The local APIC processes interrupts generated by local I/O devices, by software, and by the programmable timer. In addition, it processes interrupts communicated on the IMB, by the I/O devices connected to the IOAPIC, and by another processor's local APIC (inter-processor interrupts). The local APIC communicates these local and external interrupts to the processor. In addition, the local APIC is responsible for the nesting, queueing, and masking of interrupts. The local APIC can be implemented as a discrete component or integrated with the processor. The AMD Athlon processor integrates the local APIC.

## IOAPIC

The IOAPIC consists of 24 incoming interrupts, a 24-entry redirection table, several programmable registers, and an IMB unit for interfacing with the IMB as shown in Figure 35. Each of the incoming interrupts has a corresponding entry in the redirection table. The entry is programmed with edge/level sensitivity, interrupt vector, priority, the destination processor, and whether the processor is selected statically or dynamically. When an incoming interrupt is asserted, the IOAPIC uses the information programmed in the redirection table entry to format the interrupt message to be communicated on the IMB.

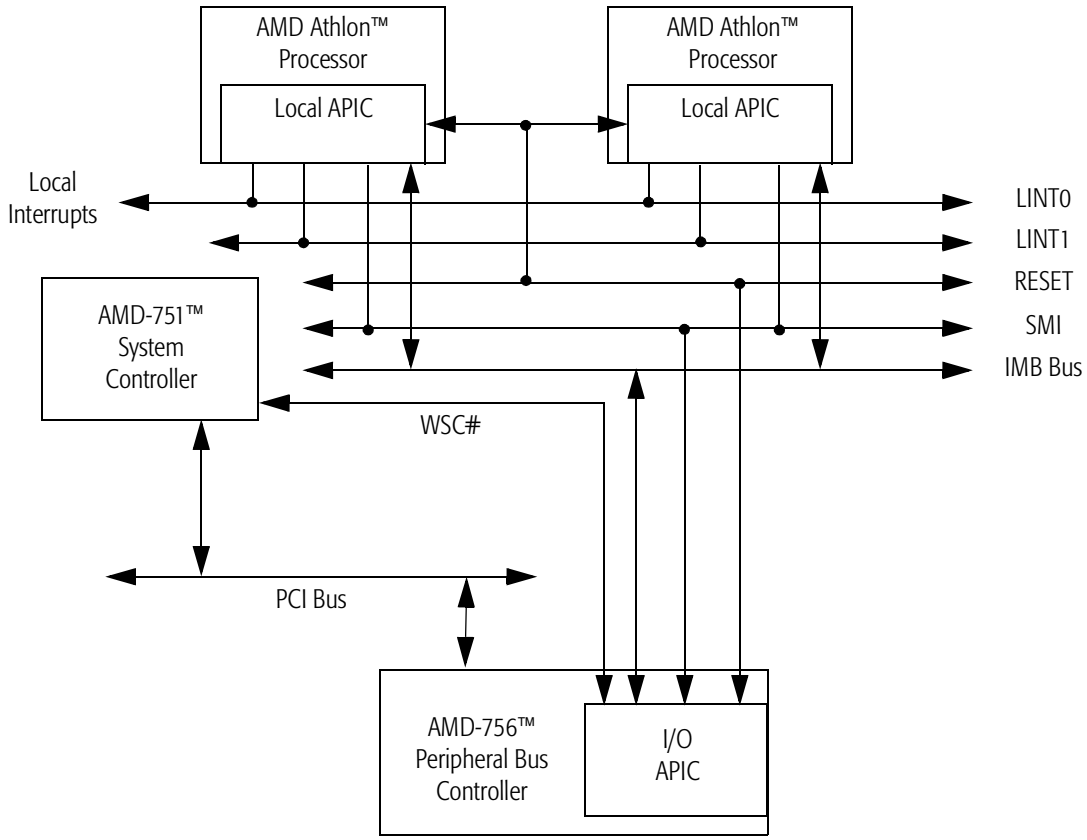


**Figure 35. System-Level Implementation of APIC Components**

**Write Snoop Capable Input**

Before sending an interrupt message, the AMD-756 peripheral bus controller waits for an acknowledge signal (WSC#), which indicates that the APIC can send the interrupt message. The WSC# signal connects between the AMD-756 and the AMD-751 system controller and is used to confirm that, before an interrupt message is sent across the IMB, the most recent PCI bus cycles that write data to system memory have been placed in 'coherent' memory space. In general, interrupt lines (IRQs) can be activated when a device finishes sending data across the PCI bus. However, there is a possibility that the data is not accessible to the processor before the interrupt service routine attempts to access it. To avoid this possibility, the IOAPIC does not send an interrupt message on the IMB until it receives confirmation that all data in the AMD-751 buffers has made it to coherent space.

The AMD-756 peripheral bus controller requests that the AMD-751 system controller issue a Fence command to its

buffers by placing a single PCLK pulse on WSC#. The AMD-751 system controller then marks the data currently in its buffers and waits for this data to reach processor-accessible space. When this data reaches processor-accessible space, the AMD-751 responds by sending a two-clock pulse back to the AMD-756 peripheral bus controller. After this pulse is received, the AMD-756 transmits the interrupt message over the IMB.

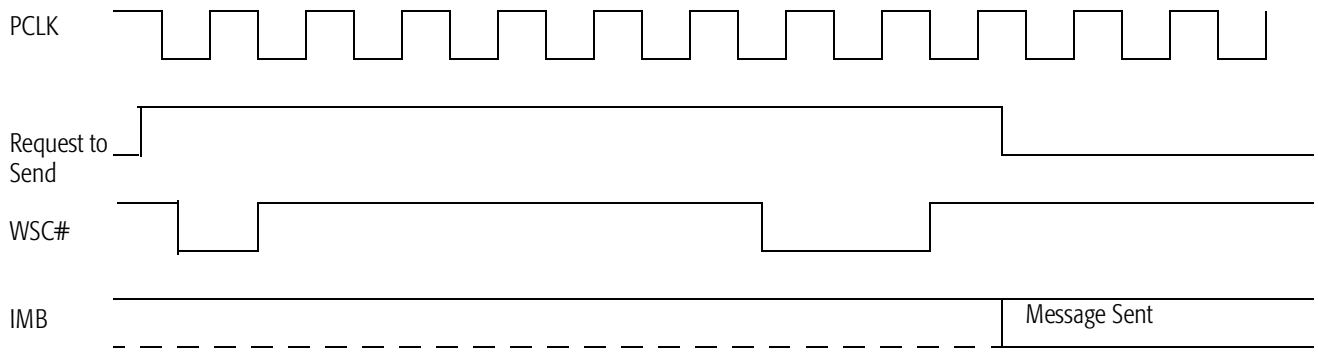


**Figure 36. WSC# Timing**

**Interrupt Message Bus (IMB)**

The local APICs and the IOAPICs communicate via the three-line synchronous interrupt message bus (IMB). Two of these lines (PICD0 and PICD1) are used for data transmission and the third line (PICLK) is a clock. In addition, PICD0 is used for arbitration.

**Arbitration**

A rotating priority arbitration protocol is used to gain access to the bus and send an interrupt message. After reset, each APIC is assigned an arbitration priority of 0 to 15. The arbitration priority is loaded with the value of the arbitration ID after reset. Before sending a message, each APIC presents the type of message it is sending and its current arbitration priority on the APIC bus. The message types are EOI message, which is a high-priority message, and normal message, which can be either a short message or a nonfocused lowest-priority message. Because the EOI message is the highest-priority message, it is granted the bus regardless of the arbitration priority of the sender. If the message is a normal message, the APIC with the highest priority wins the arbitration and sends the message. After the message has been successfully sent, all APICs increment their priority by 1. The APIC whose priority was 15 takes the winners arbitration priority.

# 6 Initialization

All programmable features in the AMD-756 peripheral bus controller are controlled by the PCI configuration registers, which are normally programmed only during system initialization. This chapter summarizes the register functions, default values, access types, and addresses. For more detailed descriptions of the configuration registers, see Chapter 7 on page 169.

Access types are indicated as follows:

| | |
|---|---|
| RW | Read/Write |
| RO | Read Only |
| WO | Write Only |
| RWC | Read, Write 1's to clear individual bits |

# 6.1 Legacy I/O Registers

**Table 30. Slave DMA Controller Registers**

| Port | Register Name | Access |
|---|---|---|
| 00h | Ch 0 Base/Current Address | RW |
| 01h | Ch 0 Base/Current Count | RW |
| 02h | Ch 1 Base/Current Address | RW |
| 03h | Ch 1 Base/Current Count | RW |
| 04h | Ch 2 Base/Current Address | RW |
| 05h | Ch 2 Base/Current Count | RW |
| 06h | Ch 3 Base/Current Address | RW |
| 07h | Ch 3 Base/Current Count | RW |
| 08h | Status/Command | RW |
| 09h | Write Request | WO |
| 0Ah | Write Single Mask | WO |
| 0Bh | Write Mode | WO |
| 0Ch | Clear Byte Pointer F/F | WO |
| 0Dh | Master Clear | WO |
| 0Eh | Clear Mask | WO |
| 0Fh | R/W All Mask Bits | RW |

**Table 31. Master Interrupt Controller Registers**

| Port | Register Name | Access |
|---|---|---|
| 20h | Master Interrupt Control | note 1 |
| 21h | Master Interrupt Mask | note 1 |
| 20h | Master Interrupt Control Shadow | RW |
| 21h | Master Interrupt Mask Shadow | RW |

***Note:***
*1. RW if shadow registers are disabled*

**Table 32. Timer/Counter Registers**

| Port | Register Name | Access |
|---|---|---|
| 40h | Timer/Counter 0 | RW |
| 41h | Timer/Counter 1 | RW |
| 42h | Timer/Counter 2 | RW |
| 43h | Timer/Counter Control | WO |

**Table 33. Keyboard Controller Registers**

| Port | Register Name | Access |
|---|---|---|
| 60h | Keyboard Controller Data | RW |
| 61h | Misc. Functions and Speaker Control | RW |
| 64h | Keyboard Controller Command/Status | RW |

**Table 34. CMOS/RTC/NMI Registers**

| Port | Register Name | Access |
|---|---|---|
| 70h | CMOS Memory Address & NMI Disable | WO |
| 71h | CMOS Memory Data (128 bytes) | RW |
| 72h | CMOS Memory Address | RW |
| 73h | CMOS Memory Data (256 bytes) | RW |
| 74h | CMOS Memory Address | RW |
| 75h | CMOS Memory Data (256 bytes) | RW |

**Table 35. DMA Page Registers**

| Port | Register Name | Access |
|---|---|---|
| 87h | DMA Page—DMA Channel 0 | RW |
| 83h | DMA Page—DMA Channel 1 | RW |
| 81h | DMA Page—DMA Channel 2 | RW |
| 82h | DMA Page—DMA Channel 3 | RW |
| 8Fh | DMA Page—DMA Channel 4 | RW |
| 8Bh | DMA Page—DMA Channel 5 | RW |
| 89h | DMA Page—DMA Channel 6 | RW |
| 8Ah | DMA Page—DMA Channel 7 | RW |

**Table 36. System Control Registers**

| Port | Register Name | Access |
|---|---|---|
| 92h | System Control | RW |

**Table 37. Slave Interrupt Controller Registers**

| Port | Register Name | Access |
|---|---|---|
| A0h | Slave Interrupt Control | Note 1 |
| A1h | Slave Interrupt Mask | Note 1 |
| A0h | Slave Interrupt Control Shadow | RW |
| A1h | Slave Interrupt Mask Shadow | RW |

***Note:***
*1. RW if shadow registers are disabled*

**Table 38. Master DMA Controller Registers**

| Port | Register Name | Access |
|---|---|---|
| C0h | Ch 0 Base/Current Address | RW |
| C2h | Ch 0 Base/Current Count | RW |
| C4h | Ch 1 Base/Current Address | RW |
| C6h | Ch 1 Base/Current Count | RW |
| C8h | Ch 2 Base/Current Address | RW |
| CAh | Ch 2 Base/Current Count | RW |
| CCh | Ch 3 Base/Current Address | RW |
| CEh | Ch 3 Base/Current Count | RW |
| D0h | Status/Command | RW |
| D2h | Write Request | WO |
| D4h | Write Single Mask | WO |
| D6h | Write Mode | WO |
| D8h | Clear Byte Pointer F/F | WO |
| DAh | Master Clear | WO |
| DCh | Clear Mask | WO |
| DEh | R/W All Mask Bits | RW |

**Table 39. Miscellaneous Control**

| Port | Register Name | Access |
|---|---|---|
| 0F1h - 0F0h | CPU Reset/FERR Clear | RW |

**Table 40. Interrupt Controller Level Select**

| Port | Register Name | Access |
|---|---|---|
| 04D1h - 04D0h | Level Sensitive IRQ Select | RW |

**Table 41. PCI Control**

| Port | Register Name | Access |
|---|---|---|
| 0CFBh - 0CF8h | PCI Address | RW |
| 0CFFh - 0CFCh | PCI Data | RW |

**Table 42. Configuration Space PCI-to-ISA Header Registers**

| Offset | PCI Header | Default | Access |
|---|---|---|---|
| 01h–00h | Vendor ID | 1022h | RO |
| 03h–02h | Device ID | 7408h | RO |
| 05h–04h | Command | 000Fh | RW |
| 07h–06h | Status | 0200h | RWC |
| 08h | Revision ID (01h = Rev A) | 01H | RO |
| 09h | Program Interface | 00h | RO |
| 0Ah | Sub Class Code | 01h | RO |
| 0Bh | Base Class Code | 06h | RO |
| 0Ch | Reserved (Cache Line Size) | -- | – |
| 0Dh | Reserved (Latency Timer) | -- | – |
| 0Eh | Header Type | 80h | RO |
| 0Fh | Built-In Self Test (BIST) | 00h | RO |
| 2Dh-2Ch | Bridge Subsystem Vendor ID | 0000h | RO |
| 2Fh-2Eh | Bridge Subsystem ID | 0000h | RO |
| 10h–3Fh | Reserved | -- | – |

**Table 43. ISA Bus Control Registers**

| Offset | Register | Default | Recommended | | Access |
|---|---|---|---|---|---|
| | | | Setting | Result | |
| 40h | ISA Bus Control 1 | 00h | 00h | Normal ISA timing | RW |
| 41H | ISA Bus Control 2 | 00h | 01h | Refresh mode | RW |

**Table 43. ISA Bus Control Registers (continued)**

| Offset | Register | Default | Recommended | | Access |
|---|---|---|---|---|---|
| | | | Setting | Result | |
| 42h | Reserved | | | | |
| 43h | ROM Decode Control | 00h | 00h | ROMCS# F0000h-FFFFFh | RW |
| 44h | Keyboard Controller Control | 00h | 01h | Disable mouse lock | RW |
| 45h | Type F DMA Control | 00h | 00h | Set DMA type F if needed | RW |
| 46h | Miscellaneous Control 1 | 00h | 00h | Disable post memory write | RW |
| 47h | Miscellaneous Control 2 | 00h | C0h | INIT as CPU reset<br>Enable PCI delay transaction | RW |
| 48h | Miscellaneous Control 3 | 01h | 01h | Enable USB, IDE | RW |
| 49h | Miscellaneous Control 4 | 08h | 08h | ISA Bus to be 12 milliamps | — |
| 4Ah | IDE Interrupt Routing | 84h | 84h | Wait for PGNT before grant to ISA master/DMA<br>IDE primary channel IRQ14<br>Secondary channel IRQ 15 | RW |
| 4Bh | IOAPIC Configuration Register | 00-- | 00-- | Accesses to IOAPIC are disabled | — |
| 4Ch | DMA/Master Mem Access Ctrl 1 | 00h | 00h | PCI memory hole bottom address HA23–HA16 = 0 | RW |
| 4Dh | DMA/Master Mem Access Ctrl 2 | 00h | 00h | PCI memory hole top address HA23–HA16 = 0 | RW |
| 4F-4Eh | DMA/Master Mem Access Ctrl 3 | 0300h | F300h | Top of PCI memory for ISA=16M.<br>Forward 00000h-9FFFFh access to PCI | RW |
| 52h-50h | Bridge Subsystem Vendor ID | 0000h | | Value placed is accessed via 2Dh-2Ch | RW |
| 54h-52hh | Bridge Subsystem ID | 0000h | | Value placed is accessed via 2Fh-2Eh | RW |

**Table 44. Distributed DMA**

| Offset | Register | Default | Recommended | | Access |
|---|---|---|---|---|---|
| | | | Setting | Result | |
| 61h–60h | Channel 0 Base Address/Enable | 0000h | 0000h | Disabled | RW |
| 63h–62h | Channel 1Base Address/Enable | 0000h | 0000h | Disabled | RW |
| 65h–64h | Channel 2 Base Address/Enable | 0000h | 0000h | Disabled | RW |
| 67h–66h | Channel 3 Base Address/Enable | 0000h | 0000h | Disabled | RW |
| 69h–68h | Reserved | -- | — | Disabled | — |
| 6Bh–6Ah | Channel 5 Base Address/Enable | 0000h | 0000h | Disabled | RW |
| 6Dh–6Ch | Channel 6 Base Address/Enable | 0000h | 0000h | Disabled | RW |
| 6Fh–6Eh | Channel 7 Base Address/Enable | 0000h | 0000h | Disabled | RW |
| 71h-70h | Device and Subsystem ID RW | 0000h | 0000h | | RO |
| FFh–72h | Reserved | -- | — | | — |

# 6.2 PCI Function 1 Registers—IDE Controller

**Table 45. Configuration Space IDE Header Registers**

| Offset | PCI Header | Default | Access |
|---|---|---|---|
| 01h–00h | Vendor ID | 1022h | RO |
| 03h–02h | Device ID | 7409h | RO |
| 05h–04h | Command | 0080h | RW |
| 07h–06h | Status | 0280h | RW |
| 08h | Revision ID | 02h | RO |
| 09h | Program Interface | 8Ah | RW |
| 0Ah | Sub Class Code | 01h | RO |
| 0Bh | Base Class Code | 01h | RO |
| 0Ch | Reserved (Cache Line Size) | 00h | — |
| 0Dh | Latency Timer | 00h | RW |
| 0Eh | Header Type | 00h | RO |
| 0Fh | Built-In Self Test (BIST) | 00h | RO |
| 13h–10h | Base Address—Primary Data/Command | 0000_01F1h | RW |
| 17h–14h | Base Address—Primary Control/Status | 0000_03F5h | RW |
| 1Bh–18h | Base Address—Secondary Data/Command | 0000_0171h | RW |
| 1Fh–1Ch | Base Address—Secondary Control/Status | 0000_0375h | RW |
| 23h–20h | Base Address—Bus Master Control | 0000_CC01h | RW |
| 2Fh–24h | Reserved (unassigned) | -- | — |
| 33h–30h | Reserved (expansion ROM base address) | -- | — |
| 3Bh–34h | Reserved (unassigned) | -- | — |
| 3Ch | Interrupt Lines | 00h | RW/RO |
| 3Dh | Interrupt Pin | 01h | RO |
| 3Eh | Minimum Grant | 00h | RO |
| 3Fh | Maximum Latency | 00h | RO |

**Table 46. Configuration Space IDE Registers**

| Offset | Register | Default | Recommended | | Access |
|---|---|---|---|---|---|
| | | | Setting | Result | |
| 40h | Chip Enable | 00h | 03h | Enable pri and sec channel | RW |
| 41h | IDE Configuration | 002h | E0h | Enable pri and sec read prefetch buffer<br>Enable pri post write buffer | RW |
| 47h-42h | Reserved (do not program) | -- | -- | | -- |
| 4Bh–48h | Drive Timing Control | A8A8A8A8h | A8A8A8A8h | DIOR# and DIOW# pulse width set to 11 PCI clocks<br>Recovery time set to 9 clocks | RW |
| 4Ch | Address Setup Time | FFh | FFh | Address setup time 4T | RW |
| 4Dh | Reserved (do not program) | -- | -- | | -- |
| 4Eh | Sec Non-1F0h Port Access Timing | FFh | FFh | Sec non-1F0 port access, DIOR# and DIOW# pulse width set to 17 PCI clocks | RW |
| 4Fh | Pri Non-1F0h Port Access Timing | FFh | FFh | Pri non-1F0 port access, DIOR# and DIOW# pulse width set to 17 PCI clocks | RW |
| 53h–50h | Ultra DMA33 Extended Timing Control | 03030303h | 03030303h | Pri and sec Drive 0 and 1Mode enabled by Set Feature command<br>Disabled Ultra DMA33 mode | RW |
| 57h-54h | Reserved | -- | — | | — |
| 5Fh–58h | Reserved | -- | -- | | — |
| 61h–60h | Reserved | -- | -- | | — |
| 67h–62h | Reserved | -- | — | | — |
| 69h–68h | Reserved | -- | -- | | — |
| 6Ah–FFh | Reserved | -- | -- | | — |

**Table 47. IDE Controller I/O Registers**

| Offset | Register Name | Default | Access |
|---|---|---|---|
| 00h | Primary Channel Command | 00h | RW |
| 01h | Reserved | -- | – |
| 02h | Primary Channel Status | 00h | RWC |
| 03h | Reserved | -- | – |
| 07h–04h | Primary Channel PRD Table Address | 00h | RW |
| 08h | Secondary Channel Command | 00h | RW |
| 09h | Reserved | -- | – |
| 0Ah | Secondary Channel Status | 00h | RWC |
| 0Bh | Reserved | -- | – |
| 0Fh–0Ch | Secondary Channel PRD Table Address | 00h | RW |

# 6.3 PCI Function 3 Registers—Power Management

## 6.3.1 Power Management Configuration Space Registers

**Table 48. Configuration Space Power Management Header Registers**

| Offset | PCI Header | Default | Access |
|---|---|---|---|
| 01h–00h | Vendor ID | 1022h | RO |
| 03h–02h | Device ID | 7403h | RO |
| 05h–04h | Command | 0000h | RW |
| 07h–06h | Status | 0280h | RWC |
| 08h | Revision ID | 01h | RO |
| 09h | Program Interface | 00h | RO |
| 0Ah | Sub Class Code | 00h | RO |
| 0Bh | Base Class Code | 00h | RO |
| 0Ch | Cache Line Size | 00h | RO |
| 0Dh | Latency Timer | 16h | RW |
| 0Eh | Header Type | 00h | RO |

**Table 49. Configuration Space Power Management Registers**

| Offset | Register | Default | Recommended | | Access |
|---|---|---|---|---|---|
| | | | Setting | Result | |
| 41h | General Configuration | 40h | 40h | Enable ACPI timer reset<br>ACPI 24-bit timer count<br>32us clock throttling | RW |
| 42h | SCI Interrupt Configuration | 00h | 00h | Disable SCI interrupt | RW |
| 43h | Most Previous Power State | 00h | 00h | | RW |
| 45h–44h | PNP IRQ Select | 0000h | 0000h | Disable pri interrupt channel | RW |
| 47h–46h | PNP DMA and chip select | 0000h | 0000h | Disable sec interrupt channel | RW |
| 48h | Reset Strap Options | xxh | -- | Enable Int RTc, PS2 mouse, Int KBC | RW |
| 4Ah | Serial IRQ Control | 10h | 10h | 6 start clocks | RW |
| 4Ch | PRDY Timer Control | 00h | 00h | Enables counters | RW |
| 4Eh | Square Wave Generation | 00h | 00h | Disable square wave | RW |
| 53h–50h | Power State Pin Control | 0000_0000h | 0000_0000h | Disable power management | RW |

**Table 49. Configuration Space Power Management Registers (continued)**

| Offset | Register | Default | Recommended | | Access |
|---|---|---|---|---|---|
| | | | Setting | Result | |
| 5Bh-58h | System Management I/O Space Pointer | 0000_DD01h | 0000_DD01h | System management I/O base | RW |
| 5Fh-5Ch | Reserved | -- | -- | | -- |
| 63h-60h | System Management Class Code Write | 0000_0000h | 0000_0000h | Value to be returned by register at offset 08h (Classcode) | RW |
| A3h-A0h | Serial Port Trap Address Register | 0278_03F8h | 0278_03F8 | The following trap and mask registers can be used to generate and SMI or SCI, load the associated re-trigger timers, load the system inactivity timers or load the burst timers | RW |
| A5h-A4h | Serial Port Trap Mask Register | 0F0Fh | 0F0Fh | | RW |
| ABh-A8h | Audio Port 1 and 2 Trap Address | 0330_0220h | 0330_0220h | | RW |
| AFh-ACh | Audio Port 3 and 4 Trap Address | 0338_0530h | 0338_0530h | | RW |
| B3h-B0h | Audio Port Trap Mask Register | 0707_010Fh | 0707_010Fh | | RW |
| B7h-B4h | PCMCIA Trap 1 and 2 I/O Address Register | 0000_0000h | 0000_0000h | | RW |
| BBh-B8h | PCMCIA Trap 1 Memory Address Register | 0000_0000h | 0000_0000h | | RW |
| BFh-BCh | PCMCIA Trap 2 Memory Address Register | 0000_0000h | 0000_0000h | | RW |
| C3h-C0h | PCMCIA Trap Mask Register | 0000_0000h | 0000_0000h | | RW |
| C7h-C4h | Programmable I/O Range Monitor 1 through 4 Trap Address Register | 0000_0000h | 0000_0000h | | RW |
| CBh-C8h | Programmable I/O Range Monitor 3 and 4 Trap Address | 0000_0000h | 0000_0000h | | RW |
| CFh-CCH | Programmable I/O Monitor Trap Mask Register | 0000_0000h | 0000_0000h | | RW |
| D3h-D0h | Programmable Memory Range Monitor 1 Trap Address | 0000_0000h | 0000_0000h | | RW |
| D7h-D4h | Programmable Memory Range Monitor 2 Trap Address | 0000_0000h | 0000_0000h | | RW |
| DBh-D8h | Programmable Memory Range Monitor Trap Mask Address | 0000_0000h | 0000_0000h | | RW |

## 6.3.2 Power Management I/O Space Registers

**Table 50. Basic Power Management Control/Status Registers**

| Offset | Register Name | Default | Access |
|---|---|---|---|
| 01h–00h | Power Management 1 Status | 00h | RWC |
| 03h–02h | Power Management 1 Enable | 00h | RW |
| 05h–04h | Power Management 1 Control | 00h | RW |
| 0Bh–08h | Power Management 1 Timer | 00h | RW |

**Table 51. Processor Power Management Registers**

| Offset | Register Name | Default | Access |
|---|---|---|---|
| 13h–10h | Processor Control | 0000h | RW |
| 14h | Processor Level 2 | 00h | RO |
| 15h | Processor Level 3 | 00h | RO |
| 17h-16h | Resume Event Enable | 0000h | RW |
| 19h-18h | Flag Write | 0000h | RW |
| 1Bh-1Ah | Flag Read | 0000h | RO |
| 1Ch | Soft Logic test | 00h | RW |

**Table 52. General Purpose Power Management Registers**

| Offset | Register Name | Default | Access |
|---|---|---|---|
| 23h–22h | ACPI Interrupt Enable | 00h | RW |
| 24h | LPT-USB Event Status | 00h | RW |
| 25h | LPT-USB Event Interrupt Enable | 00h | RW |
| 27h–26h | Power Supply Control | 2200h | RW |

**Table 53. Generic Power Management Registers**

| Offset | Register Name | Default | Access |
|---|---|---|---|
| 29h–28h | Global Status | 0000h | RWC |
| 2Bh–2Ah | Global SMI Enable | 0000h | RW |
| 2Dh–2Ch | Global SMI Control | 0000h | RW |
| 2Fh | Software SMI Command | 00h | RW |
| 94h-40h | Reserved | -- | -- |
| 9Bh–98h | System Inactivity Timer | 0000_0000h | RW |
| A7h-A0h | Reserved | -- | -- |
| ABh-A8h | Hardware Trap Status Register | 0000_0000h | RW |
| AFh-ACh | Hardware Trap Enable | 0000_0000h | RW |

**Table 53. Generic Power Management Registers (continued)**

| Offset | Register Name | Default | Access |
|---|---|---|---|
| B3h-B0h | Hardware Trap Reload Enable Timer | 0000_0000h | RW |
| B7h-B4h | IRQ Reload Enable System Inactivity Timer | 0000_0000h | RW |
| D1h-C0h | General Purpose I/O pins GPIO Select | | RW |
| D7h-D4h | GPIO Pin Interrupt Status | 0000_0000h | RW |
| DBh-D8h | GPIO Pin Interrupt Enable | 0000_0000h | RW |
| DFh-DCh | GPIO Output Clock 0 and 1 | FFFF_FFFFh | RW |
| E1h-E0h | SMBus Global Status | 0000h | RW |
| E3h-E2h | SMBus Global Control | 0000h | RW |
| E5h-E4h | SMBus Host Address | 0000h | RW |
| E7h-E6h | SMBus Host Data | 0000h | RW |
| E8h | SMBus Host Command Field | 00h | RW |
| E9h | SMBus Host Block DataFIFO Access | 00h | RW |
| EBh-EAh | SMBus Host-As-Slave Data | 0000h | RO |
| EDh-ECh | SMBus Host-As-Slave Device Address | 0000h | RO |
| EEh | SMBus Host-As-Slave Host Address | 00h | RW |
| EFh | SMBus Snoop Address | 00h | RW |

# 6.4 PCI Function 4 Registers—USB Controller

**Table 54. Configuration Space USB Header Registers**

| Offset | PCI Header | Default | Access |
|---|---|---|---|
| 01h–00h | Vendor ID | 1022h | RO |
| 03h–02h | Device ID | 740Ch | RO |
| 05h–04h | Command | 0000h | RW |
| 07h–06h | Status | 0280h | RWC |
| 08h | Revision ID (00h = first silicon) | 06h | RO |
| 09h | Program Interface | 10h | RO |
| 0Ah | Sub Class Code | 03h | RO |
| 0Bh | Base Class Code | 0Ch | RO |
| 0Ch | Cache Line Size | 00h | RO |
| 0Dh | Latency Timer | 10h | RW |
| 0Eh | Header Type | 00h | RO |
| 0Fh | Built-In Self Test (BIST) | 00h | RO |
| 13h-10h | Base Address | 0000_0000h | RW |
| 1Fh–14h | Reserved | – | – |
| 23h–20h | I/O Register Base Address | 0000_0001h | RW |
| 3Bh–24h | Reserved | 00h | – |
| 3Ch | Interrupt Line Register | 00h | RW |
| 3Dh | Interrupt Pin Register | 04h | RO |
| 3Eh | Min_Gnt | 00h | RO |
| 3Fh | Max_Gnt | 50h | RO |
| 44h | Operational Mode Enable | 00h | RW |

**Table 55. USB Controller I/O Registers**

| Offset | Register Name | Default | Access |
|---|---|---|---|
| 03h–00h | HCRevision | 0000_0110h | R |
| 07h-04h | HcControl | 0000_0600h | RW |
| 0Bh-08h | HcCommandStatus | 0000_0000h | RW |
| 0Fh-0Ch | HcInterruptStatus | 0000_0000h | RW |
| 13h-10h | HcInterruptEnable | 0000_0000h | RW |
| 17h-14h | HcInterruptDisable | 0000_0000h | RW |
| 1Bh-18h | HcHCCA | 0000_0000h | RW |

**Table 55. USB Controller I/O Registers (continued)**

| Offset | Register Name | Default | Access |
|---|---|---|---|
| 1Fh-1Ch | HcPeriodCurrentED | 0000_0000h | RW |
| 23h-20h | HcControlHeadED | 0000_0000h | RW |
| 27h-24h | HcControlCurrentED | 0000_0000h | RW |
| 2Bh-28h | HcBulkHeadED | 0000_0000h | RW |
| 2Fh-2Ch | HcBulkCurrentED | 0000_0000h | RW |
| 33h-30h | HcDoneHead | 0000_0000h | RW |
| 37h-34h | HcFmInterval | 0000_2EDF | RW |
| 3Bh-38h | HcFrameRemaining | 0000_0000h | RW |
| 3Fh-3Ch | HcFmNumber | 0000_0000h | RW |
| 43h-40h | HcperiodStart | 0000_0000h | RW |
| 47h-44h | HcLSThresold | 0000_0628h | RW |
| 4Bh-48h | HcRhDescriptorA | 0100_0004h | RW |
| 4Fh-4Ch | HcRhDescriptorB | 0000_0000h | RW |
| 53h-50h | HcRhStatus | 0000_0000h | RW |
| 57h-54h | HcRhPortStatus[1] | 0000_0000h | RW |
| 5Bh-58h | HcRhPortStatus[2] | 0000_0000h | RW |
| 5F-5Ch | HcRhPortStatus[3] | 0000_0000h | RW |
| 63-60h | HcRhPortStatus[4] | 0000_0000h | RW |
| 100h | HceControl | 0000_0000h | RW |
| 104h | HceInput | 0000_00XXh | RW |
| 108h | HceOutput | 0000_00XXh | RW |
| 10Ch | HceStatus | 0000_0000 | RW |

## 6.4.1 Pins Latched At The Trailing Edge Of RESET

Each of the following pins is latched at the trailing edge of a reset signal to specify a configuration. To latch a low, these pins are tied to ground through a 10-Kohm resistor. To latch a high, these pins are connected to the appropriate power plane through a 10-Kohm resistor. Each of these latches is accessible in Function 3, Offset 48 and can be modified if needed.

**DADDRF[2:0], DCS1F#**

These pins are latched at the trailing edge of PWRGD reset and read-write accessible via Function 3, Offset 48 bits[9:6]. The output of the pins is also read accessible from the keyboard

controller registers. These bits are mapped to the keyboard controller register bits as shown in Table 56.

**Table 56. Keyboard Controller Register Bit Mapping**

| Pin | Keyboard Controller Register Bit |
|---|---|
| DADDRP[2] | RP[16] |
| DADDRP[1] | RP[15] |
| DADDRP[0] | RP[14] |
| DCS1P# | RP[13] |

**ROM_KBCS#** This is the internal keyboard controller enable is latched at the trailing edge of PWRGD reset. When high, the internal keyboard is selected. When low, the external keyboard is selected. See the keyboard controller section of the pin descriptions to see how this affects the keyboard pin functions.

# 7 Registers

This section describes the AMD-756 peripheral bus controller configuration and I/O registers.

## 7.1 Table Conventions

Possible values for the default state after reset are:

- 0
- 1
- x the value is indeterminate
- d the value is programming dependent, and the value is discussed in the bit article

Possible values for the access type are:

- RW Read/Write
- RO Read Only
- WO Write Only
- RWC Write 1 to the bit to clear it

If a bit's value is 1, it said to be set (or high). Setting a bit means to make it 1. If a bit's value is 0, it is said to be cleared (or low). Clearing a bit means to make it 0.

Reserved bits should be read as 0, unless otherwise noted.

## 7.2 PCI Mechanism #1

Registers in the AMD-756 peripheral bus controller are accessed through PCI configuration mechanism #1, which is described in *PCI Local Bus Specification Revision 2.1*. It employs I/O locations 0CF8h to 0CFBh to specify the target address and locations 0CFCh to 0CFFh for data to the target address. These registers are located in the AMD-751 system controller.

The target address includes the specific PCI bus, device, function number, and register number within a PCI device. The AMD-751 system controller uses the device number to assert one of the AD[23:16] address/data lines. Access to the configuration address space of the AMD-756 peripheral bus controller requires device selection decoding to be done externally via the IDSEL signal, which functions as a chip select signal. The IDSEL signal associated with device number 0 is connected to AD16, IDSEL of device number 1 is connected to AD17, and so forth. The connection of the AMD-756 peripheral bus controller IDSEL is system-specific.

**PCI Configuration Address** **Ports 0CFBh–0CF8h**

| Bit | Name | Description | Default | Access Type |
|---|---|---|---|---|
| 31 | EN | **Configuration Space Enable.**<br>0 = I/O access passed on unchanged<br>1 = Targeted PCI device responds | 0 | RW |
| 30-24 | | **Reserved.** Always reads 0. | 0 | RW |
| 23-16 | BUSNUM | **PCI Bus Number.**<br>Selects a specific system PCI bus. | 0 | RW |
| 15-11 | DEVNUM | **Device Number.**<br>Selects a specific system device. | 0 | RW |
| 10-8 | FUNCNUM | **Function Number.**<br>Selects the number of a specific function space in memory. | 0 | RW |
| 7-2 | REGNUM | **Register Number.**<br>These bits, in conjunction with the PCI byte enable lines C/BE[3:0]#, specify the offset number of a register within the chosen function space. | 0 | RW |
| 1-0 | | **Reserved.** Always reads 0. | 0 | RW |

***Note:***
*PCI Configuration address is a read/write port that responds only to doubleword accesses. Byte or word accesses are passed on unchanged.*

**PCI Configuration Data** **Ports 0CFCh–0CFFh**

| Bit | Name | Description | Default | Access Type |
|---|---|---|---|---|
| 31 | DATA | **Configuration Data.** | x | RW |

***Note:***
*PCI Configuration data is a read/write port that responds only to doubleword accesses. Byte or word accesses are passed on unchanged.*

# 7.3 Register Summaries

The following tables (Table 57–Table 63) list all available registers that reside in the AMD-756 peripheral bus controller. The tables are in order by function number, and the table entries are in order by offset.

**Table 57. PCI-ISA Bridge Configuration (Function 0)**

| Offset | Function | Default | Access |
|---|---|---|---|
| 01h–00h | Vendor ID | 1022h | RO |
| 03h–02h | Device ID | 7408h | RO |
| 05h–04h | Command | 000Fh | RW |
| 07h–06h | Status | 0200h | RWC |
| 08h | Revision ID | 01h | RO |
| 09h | Program Interface | 00h | RO |
| 0Ah | Sub Class Code | 01h | RO |
| 0Bh | Base Class Code | 06h | RO |
| 0Ch | Cache | 00h | RO |
| 0Dh | Latency Timer | 00h | RW |
| 0Eh | Header Type | 80h | RO |
| 0Fh | BIST | 00h | RO |
| 2Bh–10h | Reserved | -- | -- |
| 2Dh–2Ch | Subsystem Vendor ID | 0000h | RO |
| 2Fh–2Eh | Subsystem ID | 0000h | RO |
| 3Fh–30h | Reserved | -- | -- |
| 40h | ISA Bus Control 1 | 00h | RW |
| 41h | ISA Bus Control 2 | 00h | RW |
| 42h | Reserved | | |
| 43h | ROM Decode Control | 00h | RW |
| 44h | Reserved | | |
| 45h | Type F DMA Control | 00h | RW |
| 46h | Miscellaneous Control 1 | 00h | RW |
| 47h | Miscellaneous Control 2 | 00h | RW |
| 48h | Miscellaneous Control 3 | 01h | RW |
| 49h | Miscellaneous Control 4 | 08h | RW |
| 4Ah | IDE Interrupt Routing | 84h | RW |

**Table 57. PCI-ISA Bridge Configuration (Function 0) (continued)**

| Offset | Function | Default | Access |
|---|---|---|---|
| 4Bh | IO APIC Configuration Register | 00h | RW |
| 4Dh-4Ch | ISA DMA Access Control 1 | 0000h | RW |
| 4Fh-4Eh | ISA DMA Access Control 3 | 0300h | RW |
| 53-50h | Device and Subsystem ID Register | 0000h | RW |
| 5Fh–54h | Reserved | -- | -- |
| 61h-60h | Distributed DMA Channel 0 | 0000h | RW |
| 63h-62h | Distributed DMA Channel 1 | 0000h | RW |
| 65h–64h | Distributed DMA Channel 2 | 0000h | RW |
| 67h-66h | Distributed DMA Channel 3 | 0000h | RW |
| 69h-68h | Reserved | -- | -- |
| 6Bh-6Ah | Distributed DMA Channel 5 | 0000h | RW |
| 6Dh-6Ch | Distributed DMA Channel 6 | 0000h | RW |
| 6Fh-6Eh | Distributed DMA Channel 7 | 0000h | RW |
| FFh-70h | Subsystem Vendor ID | 0000h | RW |

**Table 58. IDE Configuration (Function 1)**

| Offset | Function | Default | Access |
|---|---|---|---|
| 01h–00h | Vendor ID | 1022h | RO |
| 03h–02h | Device ID (IDE1 Device) | 7409h | RO |
| 05h–04h | Command | 0000h | RW |
| 07h–06h | Status | 0200h | RWC |
| 08h | Revision ID | 01 | RO |
| 09h | Program Interface | 80h | RW |
| 0Ah | Sub Class Code | 01h | RO |
| 0Bh | Base Class Code | 01h | RO |
| 0Ch | Reserved | -- | -- |
| 0Dh | Latency Timer | 00h | RW |
| 0Eh | Header Type | 00h | RO |
| 0Fh | BIST | 00h | RO |
| 13h–10h | Base Address 0 (BAR0)<br>primary data and command | 0000_01F1h | RW |
| 17h–14h | Base Address 1(BAR1)<br>primary control and status | 0000_03F5h | RW |
| 1Bh–18h | Base Address 2 (BAR2)<br>secondary data and command | 0000_0171h | RW |
| 1Fh-1Ch | Base Address 3 (BAR3)<br>secondary control and status | 0000_0375h | RW |
| 23h-20h | EIDE Controller<br>Master Control Registers Base Address | 0000_CC01h | RW |
| 3Bh–24h | Reserved | -- | -- |
| 3Dh-3Ch | Interrupt Pin and Line | 000Eh | RW |
| 3Fh-3Eh | Minimum Grant and<br>Maximum Latency | 0000h | RO |
| 43h-40h | IDE Configuration Register | 0000_0000h | RO |
| 47h-44h | Reserved | | |
| 4Bh-48h | IDE Drive Timing Control | A8A8_A8A8h | RW |
| 4Fh-4Ch | Cycle Time and Address Setup Time | FFFF_00FFh | RW |
| 53h-50h | Ultra DMA Timing Control | 0303_0303h | RW |
| 5Fh–54h | Reserved | -- | -- |
| FFh–6Ch | Reserved | -- | -- |

**Table 59. Power Management Configuration (Function 3)**

| Offset | Function | Default | Access |
|---|---|---|---|
| 01h–00h | Vendor ID | 1022h | RO |
| 03h–02h | Device ID | 740Bh | RO |
| 05h–04h | Command | 0000h | RW |
| 07h–06h | Status | 0280h | RWC |
| 08h | Revision ID | nn | RO |
| 09h | Program Interface | 00h | RO |
| 0Ah | Sub Class Code | 00h | RO |
| 0Bh | Base Class Code | 00h | RO |
| 0Ch | Cache | 00h | RO |
| 0Dh | Latency Timer | 16h | RW |
| 0Eh | Header Type | 00h | RO |
| 0Fh | BIST | 00h | RO |
| 40h–10h | Reserved | -- | -- |
| 41h | General Configuration | 40h | RW |
| 42h | SCI Interrupt Configuration | 00h | RW |
| 43h | Previous Power State | 00h | RO |
| 45h-44h | PnP IRQ Select | 0000h | RW |
| 47h-46h | PnP DMA and Chip Select | 0000h | RW |
| 49h-48h | Pins Latch On Reset | 00xx_x100_<br>00x0_x100b | RW |
| 4Ah | Serial IRQ Control | 10h | RW |
| 4Bh | Reserved | -- | -- |
| 4Ch | PRDY Timer Control | 00h | RW |
| 4Dh | Reserved | -- | -- |
| 4Eh | Square Wave Generator | 00h | RW |
| 4Fh | Reserved | -- | -- |
| 53h-50h | Power State Pin Control | 0000_0000h | RW |
| 54h | PCI IR Edge/Level Select | 00h | RW |
| 55h | Reserved | -- | -- |
| 57h-56h | PCI IRQ Routing Register | 0000 | RW |
| 5Bh-58h | System Management I/O Base (PM00) | 0000_DD01h | RO |
| 5Fh-5Ch | Test | 0000_0000h | RW |
| 63h-60h | System Management Class Code | 0000_0000h | RW |
| 9Fh–64h | Reserved | -- | -- |

**Table 59. Power Management Configuration (Function 3) (continued)**

| Offset | Function | Default | Access |
|---|---|---|---|
| A3h–A0h | Serial Port Trap Address | 02F8_03F8h | RO |
| A5h–A4h | Serial Port Trap Mask | 0F0Fh | RO |
| A7h–A6h | Reserved | -- | -- |
| ABh–A8h | Audio Port 1 and 2 Trap Address | 0330_0220h | RW |
| AFh–ACh | Audio Port 3 and 4 Trap Address | 0338_0530h | RW |
| B3h–B0h | Audio Trap Mask | 0707_010Fh | RW |
| B7h–B4h | PCMCIA Trap 1 and 2 I/O Address | 0000_0000h | RW |
| BBh–B8h | PCMCIA Trap 1 Address | 0000_0000h | RW |
| BFh–BCh | PCMCIA Trap 2 Address | 0000_0000h | RW |
| C3h–C0h | PCMCIA Trap Task | 0000_0000h | RW |
| C7h–C4h | I/O Range Monitor 1 and 2 Trap Address | 0000_0000h | RW |
| CBh–C8h | I/O Range Monitor 3 and 4 Trap Address | 0000_0000h | RW |
| CFh–CCh | I/O Range Monitor Mask | 0000_0000h | RW |
| D3h–D0h | Memory Range Monitor 1 Trap Address | 0000_0000h | RW |
| D7h–D4h | Memory Range Monitor 2 Trap Address | 0000_0000h | RW |
| DBh–D8h | Memory Range Monitor Trap Mask | 0000_0000h | RW |
| FFh–DCh | Reserved | -- | -- |

**Table 60. USB Configuration (Function 4)**

| Offset | Function | Default | Access |
|---|---|---|---|
| 01h–00h | Vendor ID | 1022h | RO |
| 03h–02h | Device ID | 740Ch | RO |
| 05h–04h | Command | 0000h | RW |
| 07h–06h | Status | 0280h | RWC |
| 08h | Revision ID | nn | RO |
| 09h | Program Interface | 10h | RO |
| 0Ah | Sub Class Code | 03h | RO |
| 0Bh | Base Class Code | 0Ch | RO |
| 0Ch | Cache | 00h | RO |
| 0Dh | Latency Timer | 10h | R |
| 0Eh | Header Type | 00h | RO |
| 0Fh | BIST | 00h | RO |
| 13h–10h | Base Address 0 (USB00) | 0000_0000h | RW |
| 3Bh–14h | Reserved | -- | -- |
| 3Ch | Interrupt Line | 00h | RW |
| 3Dh | Interrupt Pin | 04h | RW |
| 3Eh | Minimum Grant | 00h | RW |
| 3Fh | Maximum Latency | 50h | RW |
| 43h–40h | Reserved | -- | -- |
| 44h | Operational Mode | 00h | RW |
| FFh–45h | Reserved | -- | -- |

**Table 61. Enhanced IDE (EIDE) I/O Registers**

| Offset | Cache Control | Default | Access |
|---|---|---|---|
| 00h | Primary Bus Master Command | 00h | RW |
| 02h | Primary Bus Master Status | 00h | RW |
| 07h-04h | Primary Bus Master PRD Table Address | 0000_0000h | RW |
| 08h | Secondary Bus Master Command | 00h | RW |
| 0Ah | Secondary Bus Master Status | 00h | RW |
| 0Fh–0Ch | Secondary Bus Master PRD Table Address | 0000_0000h | RW |

***Note:***
*The base pointer for the Enhanced IDE (EIDE) I/O registers is Function 1, offset 20h.*

**Table 62. USB Open HCI Memory-Mapped Registers**

| Offset | Cache Control | Default | Access |
|---|---|---|---|
| 03h-00h | HC Revision | 0000_0110h | RO |
| 07h-04h | HC Control | 0000_0600h | RW |
| 0Bh-08h | HC Command/Status | 0000_0000h | RW |
| 0Fh-0Ch | HC Interrupt Status | 0000_0000h | RW |
| 13h-10h | HC Interrupt Enable | 0000_0000h | RW |
| 17h–14h | HC Interrupt Disable | 0000_0000h | RW |
| 1Bh-18h | HC HCCA | 0000_0000h | RW |
| 1Fh–1Ch | HC Period Current ED | 0000_0000h | RO |
| 23h–20h | HC Control Head ED | 0000_0000h | RW |
| 27h–24h | HC Control Current ED | 0000_0000h | RW |
| 2Bh–28h | HC Bulk Head ED | 0000_0000h | RW |
| 2Fh–2Ch | HC Bulk Current ED | 0000_0000h | RW |
| 33h–30h | HC Done Head | 0000_0000h | RW |
| 37h–34h | HC FM Interval | 0000_2EDFh | RW |
| 3Bh–38h | HC Frames Remaining | 0000_0000h | RO |
| 3Fh–3Ch | HC FM Number | 0000_0000h | RW |
| 43h–40h | HC Period Start | 0000_0000h | RW |
| 47h–44h | HC LS Threshold | 0000_0628h | RW |
| 4Bh–48h | HC RH Descriptor A | 0100_0004h | RW,RO |
| 4Fh–4Ch | HC RH Descriptor B | 0000_0000h | RW |
| 53h–50h | HC RH Status | 0000_0000h | RWC,RO |
| 57h–54h | HC RH Port Status 1 | 0000_0000h | RW |
| 5Bh–58h | HC RH Port Status 2 | 0000_0000h | RW |
| 5Fh–5Ch | HC RH Port Status 3 | 0000_0000h | RW |
| 63h–60h | HC RH Port Status 4 | 0000_0000h | RW |
| 100h | HCE Control | 0000_0000h | RO |
| 104h | HCE Input | 0000_00xxh | RW |
| 108h | HCE Output | 0000_00xxh | RW,RO |
| 10Ch | HCE Status | 0000_0000h | RW |

***Note:***
*The base pointer for the USB Open HCI Memory-Mapped registers is Function 4, offset 10h.*

**Table 63. Power Management I/O-Mapped Registers**

| Offset | Function | Default | Access |
|---|---|---|---|
| 01h–00h | Power Management 1 Status | 0000h | RWC |
| 03h–02h | Power Management 1 Enable | 0000h | RW |
| 05h–04h | Power Management 1 Control | 0000h | RW |
| 0Bh–08h | ACPI Power Management Timer | 0000_0000h | RO |
| 13h-10h | CPU Clock Control | 0000_0000h | RW |
| 14h | Processor Level 2 | 00h | RO |
| 15h | Processor Level 3 | 00h | RO |
| 17h-16h | Resume Event Enable | 0000h | RW |
| 19h-18h | Flag Write | 0000h | RW |
| 1Bh-1Ah | Flag Read | 0000h | RW |
| 1Ch | Soft Logic Test | 00h | RW |
| 23h-22h | ACPI Interrupt Enable | 0000h | RW |
| 24h | LPT-USB Event Status | 00h | RW |
| 25h | LPT-USB Event Interrupt Enable | 00h | RW |
| 27h-26h | Power Supply Control | 2200h | RW |
| 29h-28h | Global Status Register | 0000h | RW |
| 2Bh-2Ah | Global SMI Enable | 0000h | RW |
| 2Dh-2Ch | Global SMI Control | 0000h | RW |
| 2Fh | Software SMI Trigger | 00h | RW |
| 9Bh-98h | System Inactivity Timer | 0000_0000h | RW |
| ABh-A8h | Hardware Trap Status | 0000_0000h | RW |
| AFh-ACh | Hardware Trap Enable | 0000_0000h | RW |
| B3h-B0h | Hardware Trap Reload Enable | 0000_0000h | RW |
| B7h-B4h | IRQ Reload Enable | 0000_0000h | RW |
| D1h-C0h | GPIO [17:0] | 00h | RW |
| D7h-D4h | GPIO Pin Interrupt Status | 0000_0000h | RW |
| DBh-D8h | GPIO Pin Interrupt Enable | 0000_0000h | RW |
| DFh-DCh | GPIO Output Clock 0 and 1 | FFFF_FFFFh | RW |
| E1h-E0h | SMBus Global Status | 0000h | RW |
| E3h-E2h | SMBus Global Control | 0000h | RW |
| E5h-E4h | SMBus Host Address | 0000h | RW |
| E7h-E6h | SMBus Host Data | 0000h | RW |
| E8h | SMBus Host Command Field | 00h | RW |
| E9h | SMbus Host block Data FIFO | 00h | RW |

**Table 63. Power Management I/O-Mapped Registers (continued)**

| Offset | Function | Default | Access |
|---|---|---|---|
| EBh-EAh | SMBus Host -As-Slave Data | 0000h | RW |
| EDh-ECh | SMBus Host-As-Slave Device Address | 0000h | RW |
| EEh | SMBus Host-As-Slave Host Address | 10h | RW |
| EFh | SMBus Snoop Address | 10h | RW |

***Note:***
*The base pointer for the Power Management I/O-Mapped registers is Function 3, offset 58 (PM00).*

# 7.4 Legacy I/O Registers

This group of I/O registers includes keyboard and mouse controllers, DMA controllers, interrupt controllers, and timer/counters, as well as a number of miscellaneous ports originally implemented using discrete logic on the original PC/AT. These registers are implemented in an industry-standard manner for backwards compatibility with previous generations of PC hardware.

The legacy I/O registers are listed for reference only. Detailed descriptions of the actions and programming of these registers are given in other industry publications. All of the registers reside in I/O space. They are grouped according to the AMD-756 peripheral bus controller functions. The I/O port address and access type are given for each register.

## 7.4.1 Keyboard Controller Registers

The keyboard controller handles the keyboard and mouse interfaces using port 60h and port 64h. Reads from port 64h return a status byte. Writes to port 64h are command codes. Data is transferred via port 60h.

### Legacy I/O Registers: Keyboard/Mouse Status — Port 64h

| Bit | Name | Description | Reset | Access Type |
|---|---|---|---|---|
| 7 | PE | **Parity Error.** When this bit is set, it indicates a parity error (even parity) on the last byte received from the keyboard/mouse<br>0 = No parity error<br>1 = Parity error on last byte from keyboard/mouse | 0 | RO |
| 6 | GRT | **General Receive/Transmit Timeout.**<br>0 = No error<br>1 = Timeout error occurred. | 0 | RO |
| 5 | MOB | **Mouse Output Buffer Full.**<br>0 = Buffer empty<br>1 = Mouse buffer full | 0 | RO |
| 4 | KS | **Keylock Status.**<br>0 = Locked<br>1= Free (Keylock not set) | 0 | RO |
| 3 | CD | **Command/Data.**<br>0 = Data write<br>1 = Command write | 0 | RO |
| 2 | SF | **System Flag.**<br>0 = Power-on<br>1 = Self test successful | 0 | RO |
| 1 | IB | **Input Buffer Full.**<br>0 = Keyboard input buffer empty<br>1 = Keyboard input buffer full | 0 | RO |
| 0 | KOB | **Keyboard Output Buffer Full.**<br>0 = Keyboard output buffer empty<br>1 = Keyboard output buffer full | 0 | RO |

### Legacy I/O Registers: Keyboard/Mouse Command — Port 64h

| Bit | Name | Description | Reset | Access Type |
|---|---|---|---|---|
| 7-0 | | **Keyboard/Mouse Command.** Value of the specific keyboard/mouse command (See Table 64) | 0 | WO |

**Table 64. Keyboard Controller Command Codes**

| Command Code | Keyboard Command Code Description |
|---|---|
| 20h | Read Control Byte (next byte is Control Byte) |
| 60h | Write Control Byte (next byte is Control Byte) |
| 9xh | Write the value of x (bits 0–3) to input ports P10–P13 |
| A1h | Output Keyboard Controller Version Number |
| A4h | Test if password is installed (returns F1h to indicate 'not installed') |
| A7h | Disable Mouse Interface |
| A8h | Enable Mouse Interface |
| A9h | Mouse Interface Test (results in port 60h)<br>0 = Pass, 1 = Clock stuck low, 2 = Clock stuck high, 3 = Data stuck low, 4 = Data stuck high, FF = General error |
| AAh | KBC Self Test (results in port 60h)<br>55h = OK, FCh = Not OK |
| ABh | Keyboard Interface Test (results in port 60h)<br>0 = OK, 1 = Clock stuck low, 2 = Clock stuck high, 3 = Data stuck low, 4 = Data stuck high, FF = General error |
| ADh | Disable Keyboard Interface |
| AEh | Enable Keyboard Interface |
| AFh | Return Version Number |
| B7h–B0h | Set pin low (each value operates on one pin - B0h:P10, B1h:P1, B2h:P2, B3h:P3, B4h:P22, B5h:P23h, B6:P14, B7h:P15) |
| BFh–B8h | Set pin high (each value operates on one pin - B0h:P10, B1h:P1, B2h:P2, B3h:P3, B4h:P22, B5h:P23h, B6:P14, B7h:P15) |
| C0h | Read Input Port (read ports P10 – P17 input data to the output buffer) |
| C1h | Poll Input Port Low (read input data on input ports P13–P11 repeatedly and put results in bits 7–5 of status register) |
| C2h | Poll Input Port High (read input data on input ports P17–P15 repeatedly and put results in bits 7–5 of status register) |
| C8h | Unblock P22–P23 (use before command D1h to change the active mode) |
| C9h | Reblock P22–P23 (protection mechanism for D1h command) |
| CAh | Read Mode (output KBC mode info to port 60 output buffer)<br>bit 0 = 0 for ISA, bit 0 = 1 for PS/2 |
| D0h | Read Port (copy P27–P20 output values to port 60h) |
| D1h | Write Port (data byte following is written to keyboard output port as if it came from the keyboard) |
| D2h | Write Keyboard Output Buffer and clear status bit 5 (write following byte to keyboard) |
| D3h | Write Mouse Output Buffer and set status bit 5 (write the following byte to the mouse, and put the value in mouse input buffer so it appears to have come from the mouse) |
| D4h | Write Mouse (write the following byte to the mouse) |

**Table 64. Keyboard Controller Command Codes (continued)**

| Command Code | Keyboard Command Code Description |
|---|---|
| E0h | Read Test Inputs (T0–T1 read to bits 0–1 of respective byte, bit0 = keyboard clock, bit 1 = mouse clock) |
| Exh | Set input ports P23–P21 per command bits 3–1 |
| Fxh | Pulse input ports P23–P20 low for 6 µsec per command bits 3–0 |

***Note:***

*1. Codes not listed are undefined or their functions are eliminated by direct control of the keyboard controller logic.*

*2. The keyboard controller is compatible with industry-standard 82C42 keyboard controllers except that, because of its integration into a larger chip, many of the I/O port pins are not available for external use as general-purpose I/O pins, even when P13–P16 are set during power-up as strapping options. Consequently, many of the commands in this table work but perform no useful function, such as commands that set P12–P17 high or low.*

*3. Setting P10–11, P22–23, P26–27, and T0–1 high or low serve no useful purpose because these bits are used to implement the keyboard and mouse ports and are directly controlled by keyboard controller logic.*

The KBC control register is accessible by writing commands 20h/60h to the command port (64h). The control byte is written by first sending a value of 60h to the command port, then sending the control byte value to port 60h. The control register can be read by sending a command of 20h to port 64h, waiting for an output-buffer-full status reading on bit 5 or bit 0 of 64h, then reading the control byte value from port 60h.

**Legacy I/O Registers: KBC Control** **Port 60h and 64h**

| Bit | Name | Description | Reset | Access Type |
|---|---|---|---|---|
| 7 | | **Reserved.** Always reads 0. | 0 | RW |
| 6 | PCC | **PC Compatibility.** This bit controls conversion of the keyboard scan codes to PC Format.<br>0 = Disable scan conversion<br>1 = Convert scan codes | 1 | RW |
| 5 | MD | **Mouse Disable.**<br>0 = Enable mouse<br>1 = Disable mouse | 0 | RW |
| 4 | KD | **Keyboard Disable.**<br>0 = Enable keyboard<br>1 = Disable keyboard | 0 | RW |
| 3 | KLD | **Keyboard Lock Disable.**<br>0 = Keyboard lock enabled<br>1 = Keyboard lock disabled | 0 | RW |
| 2 | SFLG | **System Flag.** This bit can be read back as [Status Register] port 64h, bit 2. | 0 | RW |

**Legacy I/O Registers: KBC Control** **Port 60h and 64h**

| Bit | Name | Description | Reset | Access Type |
|---|---|---|---|---|
| 1 | MIE | **Mouse Interrupt Enable.**<br>0 = Disable mouse interrupts<br>1 = Enable mouse interrupts (on IRQ12 when mouse output buffer is full) | 0 | RW |
| 0 | KBE | **Keyboard Interrupt Enable.**<br>0 = Disable keyboard interrupts<br>1 = Enable keyboard interrupts (on IRQ1 when keyboard output buffer is written) | 1 | RW |

## Traditional Keyboard Controllers

Traditional (non-integrated) keyboard controllers have an input port and an output port with specific pins dedicated to certain functions and other pins available for general purpose I/O. Specific commands are provided to set these pins high and low. All outputs are open-collector to allow the pins to function as inputs. The output value for that pin is set high (non-driving), and the desired input value is read on the input port. These ports are defined as shown in Table 65.

**Table 65. Traditional Port Pin Definition**

| Bit | Input Port | LoCode | HiCode |
|---|---|---|---|
| 0 | P10 – Keyboard Data In | B0 | B8 |
| 1 | P11 – Mouse Data In | B1 | B9 |
| 2 | P12 – Turbo Pin (PS/2 mode only) | B2 | BA |
| 3 | P13 – user defined | B3 | BB |
| 4 | P14 – user defined | B6 | BE |
| 5 | P15 – user defined | B7 | BF |
| 6 | P16 – user defined | – | – |
| 7 | P17 – undefined | – | – |
| **Bit** | **Output Port** | **LoCode** | **HiCode** |
| 0 | P20 – SYSRST (1 = execute reset) | – | – |
| 1 | P21 – GATEA20 (1 = A20 enabled) | – | – |
| 2 | P22 – Mouse data out | B4 | BC |
| 3 | P23 – Mouse clock out | B5 | BD |
| 4 | P24 – Keyboard OBF Interrupt (IRQ1) | – | – |
| 5 | P25 – Mouse OBF Interrupt (IRQ12) | – | – |
| 6 | P26 – Keyboard clock out | – | – |

**Table 65. Traditional Port Pin Definition**

| Bit | Input Port | LoCode | HiCode |
|---|---|---|---|
| 7 | P27 – Keyboard data out | – | – |
| **Bit** | **Test Port** | **LoCode** | **HiCode** |
| 0 | T0 – Keyboard Clock In | – | – |
| 1 | T1 – Mouse Clock In | – | – |

**Legacy I/O Registers: Keyboard Controller Input Buffer** **Port 60h**

| Bit | Name | Description | Reset | Access Type |
|---|---|---|---|---|
| 7-0 | | **Input Buffer.** This write-only register should only be written when port 64h, bit 1 is 0. A value of 1 indicates that the input buffer is full. | 0 | WO |

**Legacy I/O Registers: Keyboard Controller Output Buffer (RO)** **Port 60h**

| Bit | Name | Description | Reset | Access Type |
|---|---|---|---|---|
| 7-0 | | **Output Buffer.** this read-only register should only be read when port 64h, bit 0 is 1. A value of 0 indicates that the output buffer is empty. | 0 | |

## 7.4.2 DMA Controller I/O Registers

**Master DMA Controller Registers**

Channels 0–3 of the master DMA controller control system DMA channels 0–3. There are 16 master DMA controller registers, as shown in Table 66.

**Table 66. Master DMA Controller Ports 00h–0Fh**

| I/O Address Bits 15–0 | Register Name | Access Type |
|---|---|---|
| 0000_0000_000x_0000 | Ch 0 Base/Current Address | RW |
| 0000_0000_000x_0001 | Ch 0 Base/Current Count | RW |
| 0000_0000_000x_0010 | Ch 1 Base/Current Address | RW |
| 0000_0000_000x_0011 | Ch 1 Base/Current Count | RW |
| 0000_0000_000x_0100 | Ch 2 Base/Current Address | RW |
| 0000_0000_000x_0101 | Ch 2 Base/Current Count | RW |
| 0000_0000_000x_0110 | Ch 3 Base/Current Address | RW |
| 0000_0000_000x_0111 | Ch 3 Base/Current Count | RW |
| 0000_0000_000x_1000 | Status/Command | RW |

**Table 66. Master DMA Controller Ports 00h–0Fh**

| I/O Address Bits 15–0 | Register Name | Access Type |
|---|---|---|
| 0000_0000_000x_1001 | Write Request | WO |
| 0000_0000_000x_1010 | Write Single Mask | WO |
| 0000_0000_000x_1011 | Write Mode | WO |
| 0000_0000_000x_1100 | Clear Byte Pointer F/F | WO |
| 0000_0000_000x_1101 | Master Clear | WO |
| 0000_0000_000x_1110 | Clear Mask | WO |
| 0000_0000_000x_1111 | R/W All Mask Bits | RW |

***Note:***
*Not all address bits are decoded.*

## Slave DMA Controller Registers

Channels 0–3 of the slave DMA controller control system DMA channels 0–3. There are 16 slave DMA controller registers, as shown in Table 67.

**Table 67. Slave DMA Controller Ports C0h–DFh**

| I/O Address Bits 15–0 | Register Name | Access Type |
|---|---|---|
| 0000_0000_1100_000x | Ch 0 Base/Current Address | RW |
| 0000_0000_1100_001x | Ch 0 Base/Current Count | RW |
| 0000_0000_1100_010x | Ch 1 Base/Current Address | RW |
| 0000_0000_1100_011x | Ch 1 Base/Current Count | RW |
| 0000_0000_1100_100x | Ch 2 Base/Current Address | RW |
| 0000_0000_1100_101x | Ch 2 Base/Current Count | RW |
| 0000_0000_1100_110x | Ch 3 Base/Current Address | RW |
| 0000_0000_1100_111x | Ch 3 Base/Current Count | RW |
| 0000_0000_1101_000x | Status/Command | RW |
| 0000_0000_1101_001x | Write Request | WO |
| 0000_0000_1101_010x | Write Single Mask | WO |
| 0000_0000_1101_011x | Write Mode | WO |
| 0000_0000_1101_100x | Clear Byte Pointer F/F | WO |
| 0000_0000_1101_101x | Master Clear | WO |
| 0000_0000_1101_110x | Clear Mask | WO |
| 0000_0000_1101_111x | R/W All Mask Bits | RW |

***Note:***
*Not all address bits are decoded.*

**DMA Page Registers**

There are eight DMA page registers, one for each DMA channel. These registers provide bits 16–23 of the 24-bit address for each DMA channel. Bits 0–15 are stored in registers in the master and slave DMA controllers. The DMA Page Registers are located at the I/O port addresses shown in Table 68.

**Table 68. DMA Page Ports 80h–8Fh**

| I/O Address Bits 15–0 | Register Name | Access Type |
|---|---|---|
| 0000_0000_1000_0111 | Ch 0 DMA Page (M–0) | RW |
| 0000_0000_1000_0011 | Ch 1 DMA Page (M–1) | RW |
| 0000_0000_1000_0001 | Ch 2 DMA Page (M–2) | RW |
| 0000_0000_1000_0010 | Ch 3 DMA Page (M–3) | RW |
| 0000_0000_1000_1111 | Ch 4 DMA Page (M–4) | RW |
| 0000_0000_1000_1011 | Ch 5 DMA Page (M–5) | RW |
| 0000_0000_1000_1001 | Ch 6 DMA Page (M–6) | RW |
| 0000_0000_1000_1010 | Ch 7 DMA Page (M–7) | RW |

## 7.4.3 Interrupt Controller Registers

**Master Interrupt Controller Registers**

The Master Interrupt Controller controls system interrupt channels 0–7. The two registers are shown in Table 69.

**Table 69. Master Interrupt Controller Ports 20h–21h**

| I/O Address Bits 15–0 | Register Name | |
|---|---|---|
| 0000_0000_001x_xxx0 | Master Interrupt Control | RW |
| 0000_0000_001x_xxx1 | Master Interrupt Mask | RW |

***Note:***
*Not all address bits are decoded.*

**Slave Interrupt Controller Registers**

The Slave Interrupt Controller controls system interrupt channels 8–15. The two registers are shown in Table 70.

**Table 70. Slave Interrupt Controller Ports A0h–A1h**

| I/O Address Bits 15–0 | Register Name | |
|---|---|---|
| 0000_0000_101x_xxx0 | Slave Interrupt Control | RW |
| 0000_0000_101x_xxx1 | Slave Interrupt Mask | RW |

***Note:***
*Not all address bits are decoded.*

## 7.4.4 Interrupt Controller Shadow Registers

The following shadow registers are enabled by setting bit 4 of offset 47h. If the shadow registers are enabled, they are read back at the indicated I/O ports instead of the standard interrupt controller registers. Writes to the standard ports are directed to the standard interrupt controller registers.

**Interrupt Controller Shadow Registers: Master Interrupt Control Shadow** **Port 20h**

| Bit | Name | Description | Reset | Access Type |
|---|---|---|---|---|
| 7-5 | | **Reserved.** Always reads 0. | 0 | RO |
| 4 | OCW3-5 | **Special Mask Mode (OCW3, bit 5).**<br>0 = Normal Masked Mode<br>1 = Special Masked Mode | x | RO |
| 3 | OCW2-7 | **Priority (OCW2, bit 7).**<br>0 = Fixed<br>1 = Rotate | x | RO |
| 2 | ICW4-4 | **Special Fully Nested Mode (ICW4, bit 4).** Setting this bit turns on the special fully nested mode.<br>0 = Off<br>1 = On | x | RO |
| 1 | ICW4-1 | **End of Interrupt (ICW4, bit 1).**<br>0 = Normal EOI<br>1 = Automatic EOI | x | RO |
| 0 | ICW1-3 | **Trigger Type (ICW1, bit 3).**<br>0 = Edge<br>1 = Level | x | RO |

***Note:***
1. *OCW = Operational Command Word*
2. *ICW = Initialization Command Word*

**Interrupt Controller Shadow Registers: Master Interrupt Mask Shadow** **Port 21h**

| Bit | Name | Description | 0-state | 1-state |
|---|---|---|---|---|
| 7-5 | | **Reserved.** Always reads 0. | 0 | RO |
| 4-0 | | **T7–T3 of the Interrupt Vector Address.** | x | RO |

### Interrupt Controller Shadow Registers: Slave Interrupt Control Shadow — Port A0h

| Bit | Name | Description | Reset | Access Type |
|---|---|---|---|---|
| 7-5 | | **Reserved.** Always reads 0. | 0 | RO |
| 4 | OCW3-5 | **Special Mask Mode (OCW3, bit 5).**<br>0 = Normal Masked Mode<br>1 = Special Masked Mode | x | RO |
| 3 | OCW2-7 | **Priority (OCW2, bit 7).**<br>0 = Fixed<br>1 = Rotate | x | RO |
| 2 | ICW4-4 | **Special Fully Nested Mode (ICW4, bit 4).** Setting this bit turns on the special fully nested mode.<br>0 = Off<br>1 = On | x | RO |
| 1 | ICW4-1 | **End of Interrupt (ICW4, bit 1).**<br>0 = NormalEOI<br>1 = Automatic EOI | x | RO |
| 0 | ICW1-3 | **Trigger Type (ICW1, bit 3).**<br>0 = Edge<br>1 = Level | x | RO |

***Note:***
1. *OCW = Operational Command Word*
2. *ICW = Initialization Command Word*

### Interrupt Controller Shadow Registers: Slave Interrupt Mask Shadow — Port A1h

| Bit | Name | Description | Reset | Access Type |
|---|---|---|---|---|
| 7-5 | | **Reserved.** Always reads 0. | 0 | RO |
| 4-0 | | **T7–T3 of the Interrupt Vector Address.** | x | RO |

### 7.4.5 Timer/Counter Registers

There are four timer/counter registers, as shown in Table 71.

**Table 71. Timer/Counter Ports 40h–43h**

| I/O Address Bits 15–0 | Register Name | |
|---|---|---|
| 0000 0000 010x xx00 | Timer/Counter 0 Count | RW |
| 0000 0000 010x xx01 | Timer/Counter 1 Count | RW |
| 0000 0000 010x xx10 | Timer/Counter 2 Count | RW |
| 0000 0000 010x xx11 | Timer/Counter Command Mode | WO |

***Note:***
*Not all address bits are decoded.*

### 7.4.6 CMOS/RTC Registers

The system real-time clock (RTC) is part of the CMOS block. The RTC control registers are located at specific offsets in the CMOS data area (00h–0Dh and 7Dh–7Fh). Detailed descriptions of CMOS/RTC operation and programming can be obtained from several industry publications. For reference, the definition of the RTC register locations and bits are summarized in Table 72.

**Table 72. CMOS Register Summary**

| Offset | Description | | Binary Range | BCD Range |
|---|---|---|---|---|
| 00h | Seconds | | 00h–3Bh | 00h–59h |
| 01h | Seconds Alarm | | 00h–3Bh | 00h–59h |
| 02h | Minutes | | 00h–3Bh | 00h–59h |
| 03h | Minutes Alarm | | 00h–3Bh | 00h–59h |
| o4h | Time: Hours | am 12 hour clock | 01h–1Ch | 01h–12h |
| | | pm 12 hour clock | 81h–8Ch | 81h–92h |
| | | 24 hour clock | 00h–17h | 00h–23h |
| 05h | Alarm: Hours | am 12 hour clock | 01h–1Ch | 01h–12h |
| | | pm 12 hour clock | 81h–8Ch | 81h–92h |
| | | 24 hour clock | 00h–17h | 00h–23h |
| 06h | Day of the Week (Sunday = 1: Saturday = 7) | | 01h–07h | 01h–07h |
| 07h | Day of the Month | | 01h–1Fh | 01h–31h |

**Table 72. CMOS Register Summary (continued)**

| Offset | Description | Binary Range | BCD Range |
|---|---|---|---|
| 08h | Month | 01h–0Ch | 01h–12h |
| 09h | Year | 00h–63h | 00h–99h |
| 0Ah | Bit 7 | Update in progress | |
| | Bits 6–4 | Divide (For 1Hz clock generation)<br>001 = Divide by 4096K<br>010 = Divide by 32K<br>011 = Divide by 1024<br>100 = Divide by 32<br>101 = RTCX_IN drives 1 Hz direct<br>110 = RTC held in reset<br>111 = RTC held in reset | |
| | Bits 3–0 | Rate select for periodic interrupt | |
| 0Bh | Bit 7 | Inhibit update transfers | |
| | Bit 6 | Periodic interrupt enable | |
| | Bit 5 | Alarm interrupt enable | |
| | Bit 4 | Update ended interrupt enable | |
| | Bit 3 | No function | |
| | Bit 2 | Data mode (0 = BCD, 1 = binary) | |
| | Bit 1 | Hours format (0 = 12, 1 = 24) | |
| | Bit 0 | Daylight saving enable | |
| 0Ch | Bit 7 | Interrupt request flag | |
| | Bit 6 | Periodic interrupt flag | |
| | Bit 5 | Alarm interrupt flag | |
| | Bit 4 | Update ended flag | |
| | Bits 3–0 | Unused (always reads 0) | |
| 0Dh | Bit 7 | VRT (= 1 if VBAT voltage is OK) | |
| | Bits 6–0 | Unused (always reads 0) | |
| 0Eh–7Ch | **Software-defined storage registers** (111 bytes) | | |
| 7Dh | Date alarm | 01h–0Fh | 01h–31h |
| 7Eh | Month alarm | 01h–0Ch | 01h–12h |
| 7Fh | Century field | 13h–14h | 19h–20h |
| 80h–FFh | **Software-defined storage registers** (128 bytes) | | |

There are three pairs of registers for accessing the on-chip CMOS RAM. Ports 70h–71h and 72h–73h access the RAM when the internal RTC is enabled, and ports 74h–75h access the RAM when the internal RTC is disabled.

Ports 70h–71h are compatible with relevant PC industry standards and access the lower 128 bytes of the 256-byte on-chip CMOS RAM. Ports 72h–73h access the full extended 256-byte space. These ports can be used only if the internal RTC is enabled by setting Function 0, offset 5Ah, bit 2. If this bit is cleared, accesses to ports 70h–71h and 72h–73h will be directed to an external RTC.

Ports 74h–75h access the full on-chip extended 256-byte space when the on-chip RTC is disabled. These ports can be accessed only if Function 0, offset 5Bh, bit 1 is set to enable the internal RTC SRAM and if offset 48h, bit 3 is set to enable access to port 74h–75h.

**CMOS/RTC Registers: CMOS Address** **Port 70h**

| Bit | Name | Description | Reset | Access Type |
|---|---|---|---|---|
| 7 | NMID | **NMI Disable.** When enabled, NMI is asserted on encountering IOCHCK# on the ISA bus or SERR# on the PCI bus.<br>0 = Enabled<br>1 = Disabled | 1 | WO |
| 6-0 | | **CMOS Address (128 bytes).** | x | WO |

**CMOS/RTC Registers: CMOS Data** **Port 71h**

| Bit | Name | Description | Reset | Access Type |
|---|---|---|---|---|
| 7-0 | | **CMOS Address (128 bytes).** | x | RW |

**CMOS/RTC Registers: CMOS Address** **Port 72h**

| Bit | Name | Description | Reset | Access Type |
|---|---|---|---|---|
| 7-0 | | **CMOS Address (256 bytes).** | x | RW |

### CMOS/RTC Registers: CMOS Data — Port 73h

| Bit | Name | Description | Reset | Access Type |
|---|---|---|---|---|
| 7-0 | | **CMOS Data (256 bytes).** | x | RW |

### CMOS/RTC Registers: CMOS Address — Port 74h

| Bit | Name | Description | Reset | Access Type |
|---|---|---|---|---|
| 7-0 | | **CMOS Address (256 bytes).** This port is intended to be used to access the internal CMOS RAM when an external real-time clock is selected. | x | RW |

### CMOS/RTC Registers: CMOS Data — Port 75h

| Bit | Name | Description | Reset | Access Type |
|---|---|---|---|---|
| 7 | | **CMOS Data (256 bytes).** This port is intended to be used to access the internal CMOS RAM when an external real-time clock is selected. | x | RW |

## 7.4.7 Miscellaneous I/O Functions

**Miscellaneous I/O Register 1** **Port 61h**

| Bit | Name | Description | Reset | Access Type |
|---|---|---|---|---|
| 7 | SERR | **SERR# Latch.** This bit is set high when PCI-bus signal SERR# is asserted and remains set until cleared by bit 2 of this register.<br>0 = No SERR#<br>1 = PCI bus signal SERR# asserted | 0 | RO |
| 6 | IOCHA | **IOCHCK# Active.** This bit is set when the ISA bus IOCHCK# signal is asserted. Once set, this bit may be cleared by setting bit 3 of this register. Bit 3 should be cleared to enable recording of the next IOCHCK#. IOCHCK# generates an NMI to the CPU if NMI is enabled.<br>0 = No IOCHCK#<br>1 = ISA bus signal IOCHCK# asserted | 0 | RO |
| 5 | T/C2O | **Timer/Counter 2 Output.** This bit is an unsynchronized reflection of the output of Timer/Counter 2. | 0 | RO |
| 4 | RD | **Refresh Detected.** This bit toggles on every rising edge of the ISA bus REFRESH# signal. | 0 | RO |
| 3 | IOCHD | **IOCHCK# Disable.** Setting this bit clears bit 6 of this register and disables IOCHCK# assertions.<br>0 = IOCHCK# enabled<br>1 = IOCHCK# disabled | 0 | RW |
| 2 | CLRSERR | **CLRSERR** Setting this bit clears bit 6 of this register. When low bit 7 of this register can be set.<br>0 = Allow bit 7 to function<br>1 = Clears bit 7 | 0 | RW |
| 1 | SE | **Speaker Enable.** Setting this bit enables the Timer/Counter 2 output to drive the SPKR pin.<br>0 = Disabled 1 = Enabled | 0 | RW |
| 0 | T/C2D | **Timer/Counter 2 Enable.**<br>0 = Disabled 1 = Enabled | 0 | RW |

**Miscellaneous I/O Register 2** **Port 92h**

| Bit | Name | Description | Reset | Access Type |
|---|---|---|---|---|
| 7-2 | | **Reserved.** Always reads 0. | 0 | |
| 1 | A20ALE | **A20 Address Line Enable.** Writing 0 to this bit forces the A20 line to 0 disabling A20.<br>0 = Disabled 1 = Enabled | 0 | RW |
| 0 | HSR | **High Speed Reset.** Setting this bit pulses reset to switch from protected mode to real mode.<br>0 = Disabled 1 = Pulse Reset | 0 | RW |

# 7.5 Function 0 Registers (PCI-ISA Bridge)

### Function 0 Registers (PCI-ISA Bridge): PCI Configuration Space Header — Offset 01h–00h

| Bit | Name | Description | Reset | Access Type |
|---|---|---|---|---|
| 15-0 | | **Vendor ID.** Always reads 1022h. | 1022h | RO |

### Function 0 Registers (PCI-ISA Bridge): Device ID (RO) — Offset 03h–02h

| Bit | Name | Description | Reset | Access Type |
|---|---|---|---|---|
| 15-0 | | **Device ID.** Always reads 7408h. | 7408h | RO |

### Function 0 Registers (PCI-ISA Bridge): Command — Offset 05h–04h

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15-4 | | 0 | **Reserved.** Always reads 0. | |
| 3 | SCE | 1 | **Special Cycle Monitoring.** 0 = ignores PCI shut down special cycles. 1 = responds to PCI shutdown cycles by generating a pulse over either CPURST# or INIT#. . | RW |
| 2 | BM | 1 | **Bus Master.** Always reads 1. 1 = Enables PCI-ISA bridge to be a master on the PCI Bus. | RO |
| 1 | MS | 1 | **Memory Space.** Always reads 1. 1 = Enable access to the ISA bus memory space. | RO |
| 0 | IOS | 1 | **I/O Space.** Always reads 1. 1 = Enable access to the ISA I/O space including the ISA bus and all the ISA registers mapped in AMD-756. | RO |

### Function 0 Registers (PCI-ISA Bridge): Status — Offset 07h–06h

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15 | DPE | 0 | **Detected Parity Error.** Always reads 0. 0 = No error 1 = Error | RO |
| 14 | SSE | 0 | **Signaled System Error.** Always reads 0. 0 = No error 1 = Error | RO |
| 13 | SMA | 0 | **Signaled Master Abort.** 0 = No abort 1 = AMD-756 controller generates master abort. | RWC |
| 12 | RTA | 0 | **Received Target Abort.** 0 = No abort 1 = AMD-756 controller receives target abort. | RWC |
| 11 | STA | 0 | **Signaled Target Abort.** Always reads 0. 0 = No abort 1 = AMD-756 controller receives target abort. | RO |
| 10-9 | DT | 01 | **DEVSEL# Timing.** Fixed at 01 = medium timing. | RO |
| 8 | DPD | 0 | **Data Parity Detected.** Always reads 0. | RO |
| 7 | FBTB | 0 | **Fast Back-to-Back Enable.** Always reads 0. | RO |
| 6 | UDF | 0 | **User-Definable Features.** Always reads 0. | RO |
| 5 | C66 | 0 | **66-MHz Capable.** Always reads 0. | RO |

### Function 0 Registers (PCI-ISA Bridge): Status — Offset 07h–06h

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 4-0 | | | **Reserved.** Always reads 0. | |

### Function 0 Registers (PCI-ISA Bridge): Revision ID — Offset 08h

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | | 01h | **Revision ID.** 01h = revision A. | RO |

### Function 0 Registers (PCI-ISA Bridge): Programming Interface — Offset 09h

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | | 00h | **Programming Interface.** Always reads 00h. | RO |

### Function 0 Registers (PCI-ISA Bridge): Sub-Class Code — Offset 0Ah

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | | 01h | **Sub-class Code.** Always reads 01h. | RO |

### Function 0 Registers (PCI-ISA Bridge): Base Class Code — Offset 0Bh

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | | 06h | **Base Class Code.** Always reads 06h. | RO |

### Function 0 Registers (PCI-ISA Bridge): Cache — Offset 0Ch

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | | 00h | **Cache.** Always reads 00h. | RO |

### Function 0 Registers (PCI-ISA Bridge): Latency — Offset 0Dh

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | | 00h | **Latency.** Always reads 00h. | RO |

### Function 0 Registers (PCI-ISA Bridge): Header Type — Offset 0Eh

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | | 80h | **Header Type.** Always reads 80h. | RO |

**Function 0 Registers (PCI-ISA Bridge): BIST** **Offset 0Fh**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | | 00h | **BIST.** Always reads 0. | RO |

**Function 0 Registers (PCI-ISA Bridge): Subsystem ID and Subsystem Vendor ID** **Offset 2Fh–2Ch**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-16 | | 0000h | **Subsystem ID.** Can be configured via offset 50h. | RO |
| 15-0 | | 0000h | **Subsystem Vendor ID.** Can be configured via offset 50h. | RO |

## 7.5.1 ISA Bus Control

**Function 0 Registers (PCI-ISA Bridge): ISA Bus Control 1** **Offset 40h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-4 | | 0 | **Reserved.** | |
| 3 | IORT | 0 | **I/O Recovery Time.**<br>0 = There is a minimum of 5.5 BCLK cycles between the trailing edge of an ISA I/O command signal (IOR# or IOW#) and the leading edge of the next ISA I/O command signal.<br>1 = There is a minimum of 13.5 BCLK cycles between adjacent I/O cycles.<br>This bit does not affect ISA bus memory cycles. | RW |
| 2 | | 0 | **Reserved**. | |
| 1 | RWS | 0 | **ROM Wait States.**<br>0 = The ISA command signal (MEMR# or MEMW#) is 2.0 BCLK cycles (wait states) in duration.<br>1 = The ISA command signal is 1.0 BCLK cycles (wait state) in duration. | RW |
| 0 | ROMW | 0 | **ROM Write.**<br>0 = The ROMCS# is disabled and does not become active during writes to BIOS address space. MEMW# is active during writes to address space near 1 megabyte but not near 4 gigabytes.<br>1 = The ROMCS# and MEMW# pins are enabled and become active during writes to the BIOS address space. | RW |

**Function 0 Registers (PCI-ISA Bridge): ISA Bus Control 2** **Offset 41h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7 | MLEN | 0 | **Mouse Lock Enable.** 0= This function is disabled. 1 = this function is enabled. | |
| 6 | | 0 | **Reserved.** Always reads 0. | |
| 5 | P92FR | 0 | **Port 92 Fast Reset.**<br>0 = Disables writes to PORT92[0].<br>1 = Enables writes to PORT92[0]. | RW |
| 4 | | 0 | **Reserved.** Always reads 0. | |

### Function 0 Registers (PCI-ISA Bridge): ISA Bus Control 2 — Offset 41h

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 3 | DDMAC | 0 | **Double DMA Clock.**<br>0 = DMA clock is BCLK (ISA clock) divided by two.<br>1 = DMA clock is the same frequency as BCLK.<br>*This bit does not affect type F DMA mode.* | RW |
| 2–1 | | 0 | **Reserved.** Always reads 0 | |
| 0 | RTM | 0 | **ISA Refresh Cycle.**<br>0 = ISA bus refresh cycles are disabled.<br>1 = ISA bus refresh cycles are enabled and occur at a frequency determined by timer 1 of the PIT. | RW |

### Function 0 Registers (PCI-ISA Bridge): ROM Decode Control — Offset 43h

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7 | RD7 | 0 | FFFE_0000h–FFFE_FFFFh 0 = Exclude 1 = Include | RW |
| 6 | RD6 | 0 | FFF8_0000h–FFFD_FFFFh 0 = Exclude 1 = Include | RW |
| 5 | RD5 | 0 | 000E_8000h–000E_FFFFh 0 = Exclude 1 = Include | RW |
| 4 | RD4 | 0 | 000E_0000h–000E_7FFFh 0 = Exclude 1 = Include | RW |
| 3 | RD3 | 0 | 000D_8000h–000D_FFFFh 0 = Exclude 1 = Include | RW |
| 2 | RD2 | 0 | 000D_0000h–000D_7FFFh 0 = Exclude 1 = Include | RW |
| 1 | RD1 | 0 | 000C_8000h–000C_FFFFh 0 = Exclude 1 = Include | RW |
| 0 | RD0 | 0 | 000C_0000h–000C_7FFFh 0 = Exclude 1 = Include | RW |

***Note:***
*Setting each bit includes the associated address range in the ROMCS# decode. The ROMCS# pin is shared with the external keyboard chip select function and is used to enable accesses to the system BIOS. ROMCS# always decodes accesses to 000F_0000h–000F_FFFFh and FFFF_0000h–FFFF_FFFFh and can also be enabled to decode the address ranges specified by this register.*

### Function 0 Registers (PCI-ISA Bridge): Type F DMA Control — Offset 45h

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7 | | | | |
| 6 | CH7 | 0 | **DMA Type F on Channel 7.** 0 = Disabled 1 = Enable Type F DMA cycles. | RW |
| 5 | CH6 | 0 | **DMA Type F on Channel 6.** 0 = Disabled 1 = Enable Type F DMA cycles. | RW |
| 4 | CH5 | 0 | **DMA Type F on Channel 5.** 0 = Disabled 1 = Enable Type F DMA cycles. | RW |
| 3 | CH3 | 0 | **DMA Type F on Channel 3.** 0 = Disabled 1 = Enable Type F DMA cycles. | RW |
| 2 | CH2 | 0 | **DMA Type F on Channel 2.** 0 = Disabled 1 = Enable Type F DMA cycles. | RW |
| 1 | CH1 | 0 | **DMA Type F on Channel 1.** 0 = Disabled 1 = Enable Type F DMA cycles. | RW |
| 0 | CH0 | 0 | **DMA Type F on Channel 0.** 0 = Disabled 1 = Enable Type F DMA cycles. | RW |

***Note:***
*Setting each enables type F DMA cycles on the corresponding DMA channel, when low AT-compatible cycles are enabled.*

### Function 0 Registers (PCI-ISA Bridge): Miscellaneous Control 1 — Offset 46h

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-1 | | | Reserved. | |
| 0 | PMWE | 0 | **Post Memory Write Enable.**<br>0 = (Disabled) The PCI targets state machine waits until the cycle is complete before passing ready back to the PCI bus.<br>1 = (Enabled) The one-DWORD write buffer from PCI target writes to the ISA bus is enabled. This buffer is only used for memory writes. | RW |

### Function 0 Registers (PCI-ISA Bridge): Miscellaneous Control 2 — Offset 47h

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7 | RS | 0 | **CPU Reset Source.**<br>0 = The CPURST pin is used for processor resets.<br>1 = The INIT pin is used for processor resets.<br>*The reset generated by function 0, offset 47[PCIRST] clears this bit before a CPURST or INIT pulse is generated. This causes reset always to use the CPURST pin.* | RW |
| 6 | DTE | 0 | **PCI Delay Transaction Enable.**<br>0 = (Disabled) PCI accesses that target the ISA bus are terminated with the disconnect command after the ISA bus cycle is complete. This often results in a PCI cycle that consumes more than the maximum 16-PCLK limit and violates the PCI specification revision 2.1.<br>1 = (Enabled) PCI accesses that target the ISA bus and internal ISA registers utilize the delayed-transaction protocol described in section 3.3.3.3 of the PCI specification revision 2.1. The table below indicates how the AMD-756 peripheral bus controller behaves based on the type of cycle. | RW |

| Cycle Type | Behavior of the AMD-756 peripheral bus controller |
|---|---|
| Memory Write | Write is posted. Within a few PCLK cycles, the bus controller terminates the write with the PCI-defined disconnect command (including the transfer of data) and completes the PCI cycle. The ISA-bus cycle continues and subsequently completes. |
| Memory read, I/O read, and I/O write | The bus controller immediately terminates the cycle with the PCI-defined retry command while executing the specified ISA-bus cycle. The master is obliged to keep retrying the cycle, which is terminated with retry each time by the bus controller. After the ISA cycle is compete, the next master-initiated retry is terminated with the disconnect command. This transfers the data (in the case of a read) and completes the PCI cycle. |
| Memory write followed by a read or I/O cycle | If a memory write is posted and is immediately followed by a memory read, I/O read, or I/O write, retry commands are issued to the master, sometimes for twice as long. The posted write completes first followed by the read or I/O cycle. |

### Function 0 Registers (PCI-ISA Bridge): Miscellaneous Control 2 (continued) Offset 47h

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 5 | PE | 0 | **EISA 4D0/4D1 Port Enable.**<br>0 = Disables the bit controls found in EI4D0 and EI4D1.<br>1 = Enables the bit controls found in EI4D0 and EI4D1.<br>EI4D0 and EI4D1 can be used to specify level-triggered interrupts for individual ISA IRQ lines. PE does not affect the ability to read or write to EI4D0 and EI4D1—only whether the outputs of these registers control the IRQ lines. | RW |
| 4 | ICSRE | 0 | **Interrupt Controller Shadow Register Enable.**<br>Obsolete bit, should always be left in its default state. | RW |
| 3–1 | | | **Reserved.** | |
| 0 | PCIRST | 0 | **Software PCI Reset.**<br>0 = No reset<br>1 = Causes a a 32-PCLK-cycle reset pulse that resets the AMD-756 peripheral bus controller and the rest of the platform (PCI bus, ISA bus, and CPU) by asserting the PCIRST#, RSTDRV, and CPURST pins. | WO |

### Function 0 Registers (PCI-ISA Bridge): Miscellaneous Control 3 Offset 48h

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7 | SELIRQ8 | 0 | **Select IRQ8.**<br>1= The RTCIRQ# is selected on the IGNNE# pin.<br>0= The IGNNE# function is provided on the IGNNE# pin. | RW |
| 6-4 | | 0 | **Reserved.** Always reads 0. | |
| 3 | RTC74D | 0 | **Extra RTC Port 74/75 Enable.**<br>0 = Accesses to I/O ports 74h and 75h generate normal ISA bus cycles.<br>1 = (And if the external real time clock is selected by function 3, offset 48, bit [INTRTC],) the internal real time clock SRAM can be accessed via I/O ports 74h for the index or address select and 75h for the data.<br>This bit has no effect if the internal real time clock is enabled. | RW |
| 2 | IUSBCD | 0 | **Integrated USB Controller Enable.**<br>0 = The USB controller is enabled.<br>1 = The USB controller is disabled. All USB configuration, memory, and I/O registers are inaccessible, and PCI mastering ability is removed. | RW |
| 1 | IDEN | 0 | **IDE Controller Enable.**<br>0 = The IDE controller is enabled.<br>1 = The IDE controller is disabled. All IDE configuration and I/O registers are inaccessible, and PCI mastering ability is removed. | RW |
| 0 | TMP | 1 | **Top of PCI Memory Decode.** Refer to function 0, offset 4Eh for more information<br>0 = Enabled 1 = Disabled | RW |

### Function 0 Registers (PCI-ISA Bridge): Miscellaneous Control 4 — Offset 49h

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7 | | | **Reserved.** | |
| 6 | RVLINT | 0 | **Reveal Hidden Interrupt Sources.**<br>0 = PICD[1]#, PICD[0]#, WSC#, and PICCLK all operate as the system Interrupt Message Bus.<br>1= PICD[1]# is PITIRQ, PICD[0]# is KBCIRQ, WSC# is SCIIRQ, and PICCLK is USBIRQ. | RW |
| 5-4 | | | **Reserved.** | |
| 3 | ISA12MA | 1 | **Select 12-mA ISA Bus Signals.**<br>0 = ISA bus signals can source and sink 24 mA.<br>1 = ISA bus signals can source and sink 12 mA. | RW |
| 2 | CMLKB8 | 0 | **CMOS RAM B8h–BFh Locked.**<br>0 = These CMOS locations are RW.<br>1 = Writes to these CMOS locations are ignored, and reads always return FFh.<br>Note: Once this bit is set, it can only be cleared by a PWRGD reset. | See Note |
| 1 | CMLK38 | 0 | **CMOS RAM 38h–3Fh Locked.**<br>0 = These CMOS locations are RW.<br>1 = Writes to these CMOS locations are ignored, and reads always return FFh.<br>Note: Once this bit is set, it can only be cleared by a PWRGD reset. | See Note |
| 0 | PRISCH | 0 | **PCI Master Access Priority.**<br>0 = The access priority is ISA bus, USB controller, IDE controller.<br>1 = The access priority is ISA bus, IDE controller, USB controller. | RW |

### Function 0 Registers (PCI-ISA Bridge): IDE Interrupt Routing — Offset 4Ah

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7 | PGNT1ST | 1 | **Wait for PGNT Before Grant to ISA Master/DMA.**<br>0 = (Disabled) No waiting occurs.<br>1 = (Enabled) Wait until the PCI bus is granted to the AMD-756 peripheral bus controller before DMA acknowledge is passed back to the ISA bus for the DMA or master data transfer. | RW |
| 6-4 | | 000 | **Reserved.** | |
| 3-2 | SIDEIRQ | 01 | **Secondary IDE IRQ Routing.** These bits determine the IRQ for the secondary IDE controller. Default is IRQ15.<br>00 = IRQ14 10=IRQ10<br>01 = IRQ15 11=IRQ11 | RW |
| 1-0 | PIDEIRQ | 00 | **Primary IDE IRQ Routing.** These bits determine the IRQ for the primary IDE controller. Default is IRQ14.<br>00 = IRQ14 10=IRQ10<br>01 = IRQ15 11=IRQ11 | RW |

### Function 0 Registers (PCI-ISA Bridge): IDE Interrupt Routing — Offset 4Ah

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| ***Note:*** *When Function 1 offset 08 bit 8 is high, then IRQ14 is assumed to be the IDE primary port interrupt pin, so PIDEIRQ should be set to its default value.* *When Function 1 offset 08 bit 10 is high, then IRQ15 is assumed to be the IDE secondary port interrupt pin, so PIDEIRQ should be set to its default value.* | | | | |

**Function 0 Registers (PCI-ISA Bridge): IOAPIC** **Offset 4Bh**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-5 | | | **Reserved.** | |
| 4-3 | APICCS[1:0] | 00 | **APIC Clock Select.** Selects the source and frequency of the PICCLK as shown in Table 73 | RW |
| 2 | SCI2IOA | 00 | **SCI to IOAPIC Redirection Register 22.**<br>1 = The SCI output of power management is routed through IRQ22 of the IOAPIC. Function 3 offset 42 [SCISEL] is disabled.<br>0 = IRQ22 of the IOAPIC is driven with the output of the GPIO16 input path. Function 3 offset 42 [SCISEL] operates normally. | RW |
| 1 | SMI2IOA | 0 | **SMI to IOAPIC Redirection Register 23.**<br>1 = The SMI output of power management is routed through IRQ23 of the IOAPIC and the SMI# pin does not become active.<br>0 = IRQ23 of the IOAPIC is driven with the output of the GPIO17 input path and the SMI# pin is controlled with the internal SMI logic. | RW |
| 0 | APICEN | 0 | **IOAPIC Enable.**<br>0 = Accesses to the IOAPIC register space are ignored and the IMB is not used.<br>1 = The IOAPIC is enabled and the IMB is used to transmit all interrupts. | RW |

## Table 73. PICCLK Frequency and Source

| APICCS[1:0] | Frequency | Source |
|---|---|---|
| 00b | PCLK divided by 4 (8.3 MHz. max) | AMD-756 drives the PICCLK |
| 01b | PCLK divided by 2 (16.7 Mhz max) | AMD-756 drives the PICCLK |
| 10b | PCLK (33.3 Mhz max) | AMD-756 drives the PICCLK |
| 11b | Unknown. In this mode, a clock source of 33 MHz or less is required. | PICCLK is driven by an external component. |

**Function 0 Registers (PCI-ISA Bridge): ISA DMA/Master Memory Access Control 1** **Offset 4Ch**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | PMHBA | 0 | **PCI Memory Hole Bottom Address.** These bits correspond to HA[23:16]. | RW |

***Note:***
*Access to the memory defined in the PCI memory hole is not forwarded to PCI. This function is disabled if the top address is less than or equal to the bottom address.*

### Function 0 Registers (PCI-ISA Bridge): ISA DMA/Master Memory Access Control 2 — Offset 4Dh

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | PMHTA | 0 | **PCI Memory Hole Top Address.** These bits correspond to HA[23:16] | RW |

***Note:***

*1. Access to the memory defined in the PCI memory hole is not forwarded to PCI. This function is disabled if the top address is less than or equal to the bottom address.*

*2. ISA Master and DMA access to the memory define by offset 4Ch(above) and offset 4Dh (below) will not cause PCI cycles.*

### Function 0 Registers (PCI-ISA Bridge): ISA DMA/Master Memory Access Control 3 — Offset 4Fh–4Eh

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15-12 | | 0000 | **Top of PCI Memory for ISA DMA/Master Accesses*.** Determines the highest memory address available for ISA DMA/Master memory accesses.<br>0000 =1 Mbyte<br>0001 =2 Mbytes<br>.<br>.<br>1111 =16 Mbytes | RW |
| 11 | | 0 | **Forward E0000h–EFFFFh.** 0 = Do not forward. 1 = Forward | RW |
| 10 | | 0 | **Forward A0000h–BFFFFh.** 0 = Do not forward. 1 = Forward. | RW |
| 9 | | 1 | **Forward 80000h–9FFFFh.** 0 = Do not forward. 1 = Forward. | RW |
| 8 | | 1 | **Forward 00000h–7FFFFh.** 0 = Do not forward. 1 = Forward. | RW |
| 7 | | 0 | **Forward DC000h–DFFFFh.** 0 = Do not forward. 1 = Forward. | RW |
| 6 | | 0 | **Forward D8000h–DBFFFh.** 0 = Do not forward. 1 = Forward. | RW |
| 5 | | 0 | **Forward D4000h–D7FFFh.** 0 = Do not forward. 1 = Forward. | RW |
| 4 | | 0 | **Forward D0000h–D3FFFh.** 0 = Do not forward. 1 = Forward. | RW |
| 3 | | 0 | **Forward CC000h–CFFFFh.** 0 = Do not forward. 1 = Forward. | RW |
| 2 | | 0 | **Forward C8000h–CBFFFh.** 0 = Do not forward. 1 = Forward. | RW |
| 1 | | 0 | **Forward C4000h–C7FFFh.** 0 = Do not forward. 1 = Forward. | RW |
| 0 | | 0 | **Forward C0000h–C3FFFh.** 0 = Do not forward. 1 = Forward. | RW |

***Note:***

*ISA DMA/masters that access addresses higher than the top of PCI are not directed to the PCI bus.*

### Function 0 Registers (PCI-ISA Bridge): Subsystem ID and Subsystem Vendor ID — Offset 53h–50h

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-16 | | 0 | **Subsystem ID.** | RW |
| 15-0 | | 0 | **Subsystem Vendor ID.** | RW |

### 7.5.2 Distributed DMA Control

**Function 0 Registers (PCI-ISA Bridge): Distributed DMA Ch n Base/Enable** **See Table 74.**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15-4 | DDBADD | 0 | **Channel n Base Address Bits.** This field specifies address bits [15:4] of the 16-byte block of I/O mapped registers in the DMA slave. Accesses to legacy DMA registers are redirected to registers in this block if CE is enabled. | RW |
| 3 | CE | 0 | **Channel n Enable.**<br>0 = Disabled 1 = Enabled | RW |
| 2-0 | | 0 | **Reserved.** Always reads 0. | |

**Table 74. DMA Channel Offset Mapping**

| Offset | Distributed DMA Channel |
|---|---|
| 61h–60h | Channel 0 |
| 63h–62h | Channel 1 |
| 65h–64h | Channel 2 |
| 67h–66h | Channel 3 |
| 6Bh–6Ah | Channel 5 |
| 6Dh–6Ch | Channel 6 |
| 6Fh–6Eh | Channel 7 |

## 7.6 Function 1 Registers (Enhanced IDE Controller)

EIDE Controller registers for the IDE controller are located in Function 1 of the AMD-756 peripheral bus controller PCI configuration space and are accessed through PCI configuration mechanism #1 via address 0CF8h/0CFCh.

The AMD-756 peripheral bus controller enhanced IDE controller interface is fully compatible with the SFF 8038i v.1.0 specification. There are two sets of software-accessible registers, the PCI configuration registers and the bus master IDE I/O registers.

## 7.6.1 Function 1 PCI Configuration Space Header

**Function 1 Registers (EIDE Controller): Vendor ID** **Offset 01h–00h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15-0 | | 00010000 00100010b | **Vendor ID.** This is a read-only register containing the value 1022h. | RO |

**Function 1 Registers (EIDE Controller): Device ID** **Offset 03h–02h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15-0 | | 01110100 00001001b | **Device ID.** This is a read-only register containing the value 7409 h. | RO |

**Function 1 Registers (EIDE Controller): Command** **Offset 05h–04h,**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15-3 | | 0 | **Reserved.** Always reads the default values. | |
| 2 | BM | 0 | **Bus Master Enable.** Scatter -gather operation can be issued only when this bit is enabled.<br>0 = Disabled 1 = Enabled | RW |
| 1 | MS | 0 | **Memory Space.** This bit is fixed in the low state.<br>0 = Access to memory mapped space of the IDE controller is disabled. | RO |
| 0 | IOS | 0 | **I/O Space.**<br>0 = (Disabled) The device does not respond to any I/O addresses for either compatible or native mode.<br>1 = (Enabled) The device responds to I/O addresses for either compatible or native modes. | RW |

### Function 1 Registers (EIDE Controller): Status — Offset 07h–06h

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15 | DPE | 0 | **Detected Parity Error.** This bit is fixed at 0. | RO |
| 14 | SSE | 0 | **Signalled System Error.** Always reads 0. | RO |
| 13 | RMA | 0 | **Signalled IDE Master Abort.** This bit is set by the hardware when bus master IDE cycle is terminated with a Master abort. | RWC |
| 12 | RTA | 0 | **Received Target Abort.** This bit set high by the hardware when a target abort command is received from the PCI bus during a master IDE initiated master cycle. | RWC |
| 11 | STA | 0 | **Signalled Target Abort.** Always reads 0. | RO |
| 10-9 | DT | 01 | **DEVSEL# Timing.** These bits are fixed at 01b, which is medium timing. | RO |
| 8-0 | | 0 | **Reserved.** Always reads 0. | |

### Function 1 Registers (EIDE Controller): Revision ID — Offset 08h

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | | 0000010b | **Revision ID.** This is a read-only register containing the revision code for the IDE controller, 02h = revision C. | RO |

### Function 1 Registers (EIDE Controller): Programming Interface — Offset 09h

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7 | MIDEC | 1 | **Master IDE Capability.** Always reads 1. | RO |
| 6-4 | | 000 | **Reserved.** Always reads 0. | |
| 3 | SPI | 1 | **Secondary Programmable Indicator.** Always reads 1. Supports both modes. Mode is selected by writing bit 2. | RO |
| 2 | SCOM | 0 | **Secondary Channel Operating Mode.**<br>0 = Compatibility Mode Function 1, Offset 18 and Function 1, Offset 1C are ignored and not visible. Address decode is based on 170h-177h, 376h. Function 1, Offset 3C, bits[7:0] are read-only and fixed at 00h. Function 1, Offset 3C, bits [15:8] are read-only and fixed at 00h. IRQ15 can be used by the IDE controller.<br>1= Native PCI Mode. Function 1, Offset 18 and Function 1, Offset 1C are visible and available for address decode. Function 1, Offset 3C, bits[7:0] are read-write. Function 1, Offset 3C, bits [15:8] are read-only and fixed at 01h. IRQ15 becomes NMSIRQ used exclusively by the secondary IDE port. | RW |
| 1 | PPI | 1 | **Primary Programmable Indicator.** Always reads 1. Supports both modes. Mode is selected by writing bit 0. | RO |
| 0 | PCOM | 0 | **Primary Channel Operating Mode.**<br>0 = Compatibility Mode Function 1, Offset 10 and Function 1, Offset 14 are ignored and not visible. Address decode is based on 1F0h-1F7h, 3F6h. Function 1, Offset 3C, bits[7:0] are read-only and fixed at 00h. Function 1, Offset 3C, bits [15:8] are read-only and fixed at 00h. IRQ14 can be used by the IDE controller.<br>1= Native PCI Mode. Function 1, Offset 10 and Function 1, Offset 14 are visible and available for address decode. Function 1, Offset 3C, bits[7:0] are read-write. Function 1, Offset 3C, bits [15:8] are read-only and fixed at 01h. IRQ14 becomes NMPIRQ, used exclusively by the primary IDE port. | RW |

**Table 75. Compatibility Mode vs. Native PCI Mode**

| Mode | IDE Channel | Command Block Registers | Control Block Registers | IRQ |
|---|---|---|---|---|
| Compatibility Mode | Primary | Fixed at I/O offset 1F7h–1F0h | Fixed at I/O offset 3F6h | 14 |
| | Secondary | Fixed at I/O offset 177h–170h | Fixed at I/O offset 376h | 15 |
| Native PCI Mode | Primary | Determined by offset 10h | Determined by offset 14h | NMPIRQ |
| | Secondary | Determined by offset 18h | Determined by offset 1Ch | NMSIRQ |

***Note:***
*Command register blocks are 8 bytes of I/O space, while control registers are 4 bytes of I/O space (only byte 2 is used).*

**Function 1 Registers (EIDE Controller): Sub Class Code** **Offset 0Ah**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | | 0000 0001b | **Sub Class Code.** Contains the value 01h, indicating an IDE device. | RO |

**Function 1 Registers (EIDE Controller): Base Class Code** **Offset 0Bh**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | | 0000 0001b | **Base Class Code.** Contains the value 01h, indicating a mass storage device. | RO |

**Function 1 Registers (EIDE Controller):** **Offset 0Ch**

| Bit | Name | Default | Description | 0-state |
|---|---|---|---|---|
| 7-0 | | 0000 0000b | **Reserved.** Always reads 0. | |

**Function 1 Registers (EIDE Controller): Latency Timer** **Offset 0Dh**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7–0 | | 0000 0000b | **Latency Timer.** This register defines the minimum amount of time, in PCI clock cycles, that the bus master can retain ownership of the bus. | RW |

**Function 1 Registers (EIDE Controller): Header Type** **Offset 0Eh**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | | 0000 0000b | **Header Type.** This register contains the value 00h. | RO |

### Function 1 Registers (EIDE Controller): BIST — Offset 0Fh

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | | 0000 0000b | **BIST.** This register contains the value 00h. | RO |

### Function 1 Registers (EIDE Controller): Primary Data/Command Base Address — Offset 13h–10h

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-3 | | 0000 0000 0000 0000 0000 0001 1111 0b | **Primary Data/Command Port Address.** These bits specify an 8-byte I/O address space that maps to the ATA-compliant register set for the primary port. | RW |
| 2-0 | | 001b | **Value fixed at 001b.** | RO |

***Note:***
*When Function 1, offset 8, bit 8 is low, the primary port is in compatibility mode and this register is ignored and not visible.*

### Function 1 Registers (EIDE Controller): Primary Control/Status Base Address — Offset 17h–14h

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-2 | | 0000 0000 0000 0000 0000 0011 1111 01b | **Primary Control/Status Port Address.** These bits specify an 4-byte I/O address space. Only the third byte is active. For example, 3F6h is the active byte for the default base address of 3F4h. | RW |
| 1-0 | | 01b | **Value fixed at 01b.** | RO |

***Note:***
*When Function 1, offset 8, bit 8 is low, the primary port is in compatibility mode and this register is ignored and not visible.*

### Function 1 Registers (EIDE Controller): Secondary Data/Command Base Address — Offset 1Bh–18h

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-3 | | 0000 0000 0000 0000 0000 0001 0111 0b | **Secondary Data/Command Port Address.** These bits specify an 8-byte I/O address space for the Command block of the device connected to the secondary IDE port. | RW |
| 2–0 | | 001b | **Value fixed at 001b.** | RO |
| ***Note:*** *When Function 1, offset 8, bit 10 is low, the secondary port is in compatibility mode and this register is ignored and not visible.* | | | | |

### Function 1 Registers (EIDE Controller): Secondary Control/Status Base Address — Offset 1Fh–1Ch

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-2 | | 0000 0000 0000 0000 0000 0011 0111 01b | **Secondary Control/Status Port Address.** These bits specify an 4-byte I/O address space. Only the third byte is active. For example, 376h is the active byte for the default base address of 374h. | RW |
| 1-0 | | 01b | **Value fixed at 01b.** | RO |
| ***Note:*** *When Function 1, offset 8, bit10 is low, the secondary port is in compatibility mode and this register is ignored and not visible.* | | | | |

### Function 1 Registers (EIDE Controller): Bus Master Control Registers Base Address (RW) — Offset 23h–20h

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-4 | | 0000 0000 0000 0000 1100 1100 0000b | **Bus Master Control Registers Base Address.** These bits specify a 16-byte I/O address space compliant with the SFF8038i rev. 1.0 specification. | |
| 3-0 | | 0001b | **Value fixed at 0001b.** | |

**Function 1 Registers (EIDE Controller): Interrupt Line** **Offset 3Ch**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | | 0000 0000b | **Interrupt Line.** If Function 1, Offset 8, bit 8 or Bit 10 is set, this is a read-write register. If function 1, offset 8, bit 8 and bit 10 are cleared, then this is a read only register and reads 00h.This field is used by the OS for interrupt routing information. | RW or RO |

**Function 1 Registers (EIDE Controller): Interrupt Pin** **Offset 3Dh**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | | 0000 0000b | **Interrupt Pin.** These bits read 01h when Function 1, Offset 8, bit 8 or Bit 10 is set. These bits read 00h when Function 1,Offset 8, bit 8 and bit 10 are both cleared. | RO |

**Function 1 Registers (EIDE Controller): Minimum Grant** **Offset 3Eh**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | | 0000 0000b | **Minimum Grant.** This is a read-only register containing the value 00h. | RO |

**Function 1 Registers (EIDE Controller): Maximum Latency** **Offset 3Fh**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | | 0000 0000b | **Maximum Latency.** This is a read-only register containing the value 00h. | RO |

## 7.6.2 IDE Configuration Registers

**Function 1 Registers (EIDE Controller): Channel Enable** **Offset 40h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-2 | | 0000 10b | **Reserved.** Always reads 02h | |
| 1 | PCE | 0 | **Primary Channel Enable.** 0 = Disabled 1 = The primary port of the IDE controller is enabled when this bit is high. | RW |
| 0 | SCE | 0 | **Secondary Channel Enable.** 0 = Disabled 1 = The primary port of the IDE controller is enabled when this bit is high. | RW |

**Function 1 Registers (EIDE Controller): IDE Configuration** **Offset 41h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7 | PRPB | 0 | **Primary IDE Read Prefetch Buffer.** 0 = Disabled 1 = Enabled | RW |
| 6 | PPWB | 0 | **Primary IDE Post Write Buffer.** 0 = Disabled 1 = Enabled. Only 32 bit writes to the data port are allowed when this bit is high**.** | RW |
| 5 | SRPB | 0 | **Secondary IDE Read Prefetch Buffer.** 0 = Disabled 1 = Enabled | RW |
| 4 | SPWB | 0 | **Secondary IDE Post Write Buffer.** 0 = Disabled 1 = Enabled. Only 32 bit writes to the data port are allowed when this bit is high**.** | RW |
| 3-0 | | 0000b | **Reserved.** These bits must remain at their default value. | RW |

### Function 1 Registers (EIDE Controller): Drive Timing Control — Offset 4Bh–48h

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-28 | PD0APW | 1010 | **Primary Drive 0 Active Pulse Width.** | RW |
| 27-24 | PD0RT | 1000 | **Primary Drive 0 Recovery Time.** | RW |
| 2320 | PD1APW | 1010 | **Primary Drive 1 Active Pulse Width.** | RW |
| 19-16 | PD1RT | 1000 | **Primary Drive 1 Recovery Time.** | RW |
| 15-12 | SD0APW | 1010 | **Secondary Drive 0 Active Pulse Width.** | RW |
| 11-8 | SD0RT | 1000 | **Secondary Drive 0 Recovery Time.** | RW |
| 7-4 | SD1APW | 1010 | **Secondary Drive 1 Active Pulse Width.** | RW |
| 3-0 | SD1RT | 1000 | **Secondary Drive 1 Recovery Time.** | RW |

***Note:***
*Each field of this register defines the active pulse width and recovery time for a particular IDE DIOR# or DIOW# signal. The actual value for each field is the encoded value plus one and indicates the number of PCI clocks. The default state (A8h) results in a recovery time of 270 ns and an active pulse of 330 ns for a 30-ns PCI clock. This corresponds to ATA PIO Mode 0.*

### Function 1 Registers (EIDE Controller): Address Setup Time — Offset 4Ch

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-6 | PD0AST | 11 | **Primary Drive 0 Address Setup Time.** | RW |
| 5-4 | PD1Ast | 11 | **Primary Drive 1 Address Setup Time.** | RW |
| 3-2 | SD0AST | 11 | **Secondary Drive 0 Address Setup Time.** | RW |
| 1-0 | SD1AST | 11 | **Secondary Drive 1 Address Setup Time.** | RW |

***Note:***
*The coding for the startup timings is as follows:*
*00 = 1 PCLK 10 = 3 PCLK*
*01 = 2 PCLK 11 = 4 PCLK*

### Function 1 Registers (EIDE Controller): Secondary Non-1F0 Port Access Timing — Offset 4Eh

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-4 | SAPW | 1111 | **DIOR#/DIOW# Active Pulse Width.** | RW |
| 3-0 | SRT | 1111 | **DIOR#/DIOW# Recovery Time.** | RW |

***Note:***
*The actual value in the field is the encoded value plus one and indicates the number of PCI clocks.*

### Function 1 Registers (EIDE Controller): Primary Non-1F0 Port Access Timing — Offset 4Fh

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-4 | PAPW | 1111 | **DIOR#/DIOW# Active Pulse Width.** | RW |
| 3-0 | PRT | 1111 | **DIOR#/DIOW# Recovery Time.** | RW |

***Note:***
*The actual value in the field is the encoded value plus one and indicates the number of PCI clocks.*

**Ultra DMA-33 Extended Timing Control** **Function 1 Offset 53h-50h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| **Bit** | **Name** | **Default** | **Description** | **0-state** |
| 31 | P0MEM[7] | 0 | **Primary Drive 0 UDMA Mode Enable Method[7].** If this bit is set, UDMA mode is enabled by setting the UDMA Mode Enable bit in this register. If this bit is cleared, UDMA mode is enabled by detecting the Set Feature ATA command. | |
| 30 | P0EN[6] | 0 | **Primary Drive 0 UDMA Mode Enable [6].** If this bit is set, UDMA mode is enabled. | |
| 29-27 | | 0 | **Primary Drive 0 UDMA [5:3].** These bits are fixed at their default values. | |
| 26-24 | P0CT[2:0] | 0 | **Primary Drive 0 UDMA Cycle Time [2:0].** These bits are defined in Table 76. | |
| 23 | P1MEM[7] | 0 | **Primary Drive 1 UDMA Mode Enable MEthod.** If this bit is set, UDMA mode is enabled by setting the UDMA Mode Enable bit in this register. If this bit is cleared, UDMA mode is enabled by detecting the Set Feature ATA command. | |
| 22 | P1EN[6] | 0 | **Primary Drive 1 UDMA Mode Enable [6].** If this bit is set, UDMA mode is enabled. | |
| 21-19 | | 0 | **Primary Drive 1 UDMA [5:3].** These bits are fixed at their default values. | |
| 18-16 | P1CT[2:0] | 0 | **Primary Drive 1 UDMA Cycle Time [2:0].** These bits are defined in Table 76. | |
| 15 | S0MEM[7] | 0 | **Secondary Drive 0 UDMA Mode Enable Method [7].** If this bit is set, UDMA mode is enabled by setting the UDMA Mode Enable bit in this register. If this bit is cleared, UDMA mode is enabled by detecting the Set Feature ATA command. | |
| 14 | S0EN[6] | 0 | **Secondary Drive 0 UDMA Mode Enable [6].** If this bit is set, UDMA mode is enabled. | |
| 13-11 | | 0 | **Secondary Drive 0 UDMA [5:3].** These bits are fixed at their default values. | |
| 10-8 | S0CT[2:0] | 0 | **Secondary Drive 0 UDMA Cycle Time [2:0].** These bits are defined in Table 76. | |
| 7 | S1MEM[7] | 0 | **Secondary Drive 1 UDMA Mode Enable Method [7].** If this bit is set, UDMA mode is enabled by setting the UDMA Mode Enable bit in this register. If this bit is cleared, UDMA mode is enabled by detecting the Set Feature ATA command. | |
| 6 | S1EN[6] | 0 | **Secondary Drive 1 UDMA Mode Enable[6].** If this bit is set, UDMA mode is enabled. | |
| 5-3 | | 0 | **Secondary Drive 1 UDMA [5:3].** These bits are fixed at their default values. | |
| 2-0 | S1CT[2:0] | 0 | **Secondary Drive 1 UDMA Cycle Time[2:0].** These bits are defined in Table 76. | |

***Note:***
*Each byte of these registers defines Ultra DMA-33 operation for the indicated drive. The bit definitions are consistent within each byte.*

**Table 76. UDMA Extended Timing**

| Cycle Time [2:0] | Ultra DMA Mode | Cycle Time |
|---|---|---|
| 000b | UDMA mode 2 | 60 ns |
| 001b | UDMA mode 1 | 90 ns |
| 010b | UDMA mode 0 | 120 ns |
| 011b | Slow UDMA mode 0 | 120 ns |
| 100b | UDMA mode 3 | 45 ns |
| 101b | UDMA mode 4 | 30 ns |

IDE Registers comply with the SFF 8038I v. 1.0 standard. The base address of these registers is determined by configuration

register Function 1, offset 20h–23h (see page 217). The command block primary channel is 1F0h–1F7h, while the secondary channel is 170H–177h. Refer to the specification for further details.

**I/O Register Primary Command** **Offset 00h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-4 | | 0000 | **Reserved .** Always reads 0. | |
| 3 | RWC | 0 | **Read or Write Control.** This bit sets the direction of the bus master transfer. This bit must not be changed when the start bit 0 is set.<br>0 = Bus master reads. 1 = Bus master writes. | RW |
| 2-1 | | 00 | **Reserved .** Always reads 0. | |
| 0 | SSBM | 0 | **Start/Stop Bus Master .** Write Only. Reads always return 0<br>0 = Master operation is stopped. Master operation cannot be stopped and resumed. If this bit is cleared while a transfer is in progress the transfer is aborted. This bit should be cleared when the data transfer is completed.<br>1 = Bus master operation is enabled. The controller then transfers data between the IDE device and memory. | W |

**I/O Register Primary Channel Status** **Offset 02h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7 | SMPLXO | 0 | **Simplex Only .**<br>0 = The primary and secondary channels can be operated at the same time.<br>1 = The primary and secondary channels cannot be operated at the same time. | RO |
| 6 | DMA1 | 0 | **Drive 1 DMA Capable.**<br>0 = No DMA capability<br>1 = DMA capable, set by device-dependent code (BIOS or device driver) to indicate that drive 1 is capable of DMA transfers, and has been initialized**.** | RW |
| 5 | DMA2 | 0 | **Drive 0 DMA Capable.**<br>0 = No DMA capability<br>1 = DMA capable, set by device-dependent code (BIOS or device driver) to indicate that drive 0 is capable of DMA transfers, and has been initialized**.** | RW |
| 4-3 | | 00 | **Reserved .** Always reads 0. | |
| 2 | Interrupt | 0 | **Interrupt.**<br>This bit is set by the rising edge of the IDE interrupt line. Then all data transferred from the drive is in system memory. | RWC |
| 1 | ERR | 0 | **Error.** 0 = No error 1 = Error | RWC |
| 0 | BMIDEA | 0 | **Bus Master IDE Active.**<br>0 = Bus master IDE is stopped.<br>1 = Bus master IDE is active.<br>This bit is set when the start bit is set in the command register. This bit is cleared when the last transfer is done. It is also cleared when the start bit in the command register is cleared. | RO |

**Table 77. Primary Channel Status Register Status Bit Meanings**

| Bit 2 | Bit 0 | Description |
|---|---|---|
| 0 | 0 | This combination signals an error. If the error bit 1 is not set, the PRD is to small. If the error bit 1 is set then there was a data transfer problem. |
| 0 | 1 | DMA transfer in progress |
| 1 | 0 | Normal completion. Physical memory equal to IDE transfer size |
| 1 | 1 | Normal completion. Physical memory greater than IDE transfer size |

**I/O Register Primary Channel PRD Table Address** **Offset 07h–04h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-2 | | 0 | **Primary Channel PRD Table Address.**<br>The Primary Channel Physical Region Descriptor (PRD) Table Address is an I/O register.<br>The descriptor table must be doubleword aligned and must not cross a 64K byte boundary in memory. | RW |
| 1-0 | | 00 | **Reserved.** Always reads 0. | |

**I/O Register Secondary Channel Command** **Offset 08h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-4 | | 0000 | **Reserved.** Always reads 0. | |
| 3 | RWC | 0 | **Read or Write Control.**<br>This bit sets the direction of the bus master transfer. This bit must not be changed when the start bit 0 is set.<br>0 = Bus master read. 1 = Bus master write. | RW |
| 2-1 | | 0 | **Reserved.** Always reads 0. | |
| 0 | SSBM | 0 | **Start/Stop Bus Master.** Reads always return 0.<br>0 = Bus master operation stopped. Master operation cannot be stopped and resumed. If this bit is cleared while a transfer is in progress the transfer is aborted. This bit should be cleared when the data transfer is completed.<br>1 = Bus master operation enabled. The controller then transfers data between the IDE device and memory. | |

**I/O Register Secondary Channel Status** **Offset 0Ah**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7 | SMPLXO | 0 | **Simplex Only.** This bit indicates whether the primary and secondary channels can be operated at the same time.<br>0 = Channels operate at the same time.<br>1 = Channels do not operate at the same time. | RO |

**I/O Register Secondary Channel Status** **Offset 0Ah**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 6 | DMA1 | 0 | **Drive 1 DMA Capable.**<br>0 = No DMA capability for drive 1.<br>1 = Drive 1 is capable of DMA transfers, and has been initialized. This bit is set by device-dependent code (BIOS or device driver). | RW |
| 5 | DMA2 | 0 | **Drive 0 DMA Capable.**<br>0 = No DMA capability for drive 0.<br>1 = Drive 0 is capable of DMA transfers, and has been initialized. This bit is set by device-dependent code (BIOS or device driver). | RW |
| 4-3 | | 0 | **Reserved .** Always reads 0. | |
| 2 | Interrupt | 0 | **Interrupt.**<br>This bit is set by the rising edge of the IDE interrupt line.<br>0 = No interrupt<br>1 = All data transferred from the drive is in system memory. | RWC |
| 1 | ERR | 0 | **Error.** 0 = No error 1 = Error detected. | RWC |
| 0 | BMIDEA | 0 | **Bus Master IDE Active.**<br>0 = Bus master stopped, indicates the last transfer is done, or the start bit in the command register is cleared.<br>1 = Bus master active, indicates the start bit is set in the command register. | RO |

**I/O Register Secondary Channel PRD Table Address** **Offset 0Fh–0Ch**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-0 | | 0 | **Secondary Channel PRD Table Address.**<br>The Secondary Channel Physical Region Descriptor (PRD) Table Address is an I/O register.<br>The descriptor table must be doubleword aligned and must not cross a 64K byte boundary in memory. | RW |

# 7.7 Function 3 Registers (Power Management)

This section describes the ACPI (Advanced Configuration and Power Interface) power management system of the AMD-756 peripheral bus controller. This system supports both ACPI and legacy power management functions and is compatible with the APM v. 1.2 and ACPI v. 0.9 specifications.

## 7.7.1 Function 3 PCI Configuration Space Header

**Power Management: Vendor ID** **Function 3 Offset 01h–00h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15-0 | | 0001000000100010 | **The Vendor ID register contains the value 1022h.** | RO |

**Power Management: Device ID** **Function 3 Offset 03h–02h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15-0 | | 0111010000001011 | **The Device ID register contains the value 740Bh.** | RO |

**Power Management: Command** **Function 3 Offset 05h–04h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15-0 | | 0 | **Reserved.** Always reads 0. | |

**Power Management: Status** **Function 3 Offset 07h–06h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15 | DPE | 0 | **Detected Parity Error.** Always reads 0. | RO |
| 14 | SSE | 0 | **Signalled System Error.** Always reads 0. | RO |
| 13 | SMA | 0 | **Signalled Master Abort.** Always reads 0. | RO |
| 12 | RTA | 0 | **Received Target Abort.** Always reads 0. | RO |
| 11 | STA | 0 | **Signalled Target Abort.** Always reads 0. | RO |
| 10-9 | DEVSEL# | 01 | **DEVSEL# Timing.** Always reads 01b (medium. timing) | RO |
| 8 | | 0 | **Data Parity Detected.** Always reads 0. | RO |
| 7 | FBB | 1 | **Fast Back-to-Back capable .** Always reads 1. | RO |
| 6 | UDEF | 0 | **User definable features.** Always reads 0**.** | RO |
| 5 | 66CAP | 0 | **66 MHz capable .** Always reads 0**.** | RO |
| 4-0 | | 0000 | **Reserved.** Always reads 0. | |

**Power Management: Revision ID** **Function 3 Offset 08h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | | 00000001 | **Silicon Revision Code.**<br>This register indicates the silicon revision code, for instance 01h = revision A, C0 and D2. 03h = D2 and greater. The register's value varies with the revision to which the chip belongs. | RO |

**Power Management: Programming Interface** **Function 3 Offset 09h**

| Bit | Name | Default | Description | 0-state |
|---|---|---|---|---|
| 7-0 | | 00000000 | **Programming Interface.**<br>This register determines the programming interface used. The value returned by this register can be changed by writing the desired value to PCI Configuration Function 3, offset 61h. | RO |

**Power Management: Sub Class Code** **Function 3 Offset 0Ah**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | | 00000000 | **Sub Class Code.**<br>The value returned by this register can be changed by writing the desired value to PCI Configuration Function 3, offset 62h. | RO |

**Power Management: Base Class Code** **Function 3 Offset 0Bh**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | | 00000000 | **Base Class Code.**<br>The value returned by this register can be changed by writing the desired value to PCI Configuration Function 3, offset 63h. | RO |

**Power Management: Cache** **Function 3 Offset 0Ch**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | | 00000000 | **Cache.** Always reads 0. | RW |

**Power Management: Latency Timer** **Function 3 Offset 0Dh**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | | 00010110 | **Latency Timer.**<br>This register contains the default 16h. | RW |

**Power Management: Header Type** **Function 3 Offset 0Eh**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | | 00000000 | **Header Type.**<br>This register contains the value 00h. | RO |

## 7.7.2 Power Management Configuration Registers

**Power Management: General Configuration** **Function 3 Offset 41h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7 | PMIOEN | 0 | **I/O Enable for ACPI I/O Base .**<br>0 = Disable access to ACPI I/O block<br>1 = Allow access to ACPI I/O register block | RW |
| 6 | ATR | 1 | **ACPI Timer Reset.**<br>0 = (Disabled) the timer is allowed to count.<br>1 = (Enabled) CPI is asynchronously cleared at all times**.** | RW |
| 5-4 | | 00 | **Reserved.** Always reads 0. | |
| 3 | ATS | 0 | **ACPI Timer Size Select.** 0 = 24-bit timer 1 = 32-bit timer | RW |

**Power Management: General Configuration** **Function 3 Offset 41h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 2 | | 0 | **Reserved.** Must remain 0 for proper operation. | |
| 1 | STPGNT | 0 | **PCI Stop Grant Cycle Specification.**<br>0 = Stop grant cycle detected when address-phase AD[4] = 1 during a PCI special cycle<br>1 = Stop grant cycle detected when data-phase AD[31:0] = 0012_0002h during a PCI special cycle | RW |
| 0 | | 0 | **Reserved.** Always reads 0. | |

**Power Management: SCI Interrupt Configuration** **Function 3 Offset 42h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-4 | | 0 | **Reserved.** Always reads 0. | |
| 3-0 | SCISEL | 0 | **SCI Interrupt Assignment.** The value of these bits specifies the legacy PIC IRQ number that is used for ACPI defined SCI interrupts. When Function 0, offset 41, SCI2IO bit is high, the IRQ to the PIC selected by this register is forced low. | RW |

| Bits | SCI Interrupt | Bits | SCI Interrupt |
|---|---|---|---|
| 0000 | disabled (default) | 1000 | IRQ7 |
| 0001 | IRQ1 | 1001 | IRQ8 |
| 0010 | Reserved | 1010 | IRQ10 |
| 0011 | IRQ3 | 1011 | IRQ11 |
| 0100 | IRQ4 | 1100 | IRQ12 |
| 0101 | IRQ5 | 1101 | Reserved |
| 0110 | IRQ6 | 1110 | IRQ14 |
| 0111 | IRQ7 | 1111 | IRQ15 |

**Power Management: Most Previous Power State** **Function 3 Offset 43h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-4 | | 0 | **Reserved .** Always reads 0. | |
| 3 | SACPFD | | **SOFF Command Status for AC Power Fail Detect.**<br>1= A command was sent to PM04 to place the system in SOFF.<br>0 = No SOFF command sent.<br>This bit is powered by VDD_RTC power plane . This is cleared by writing a 1 to it. Writing a 0 has no effect. | RWC |

**Power Management: Most Previous Power State** **Function 3 Offset 43h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 2-0 | PPSTATE | 0 | **Previous Power State .**<br><br>Bits / Effect:<br>000 MOFF Mechanical Off<br>001 SOFF soft off<br>010 FON full on<br>011 Reserved<br>100 Reserved<br>101 C2 stop grant<br>110 C3 CPU clock stopped<br>111 POS power on suspend<br><br>The default for the LSB depends on the state of the WRON# pin at the trailing edge of RST_SOFT. | RO |

**Power Management: PNP IRQ Select** **Function 3 Offset 45h–44h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15-12 | | 0 | **Reserved.** Always read 0. | |
| 11-8 | IRQ2SEL | 0 | This field assigns an IRQ to PNPIRQ2, which is ORed with the selected IRQ after the IRQ enters the part and before the IRQ goes on to the 8259-based PIC See Table 78 for IRQ assignment coding.<br>If registers PM00 + C9h-C7h do not select the PNPIRQ[2:0] functions, this field has no effect. | RW |
| 7-4 | IRQ1SEL | 0 | This field assigns an IRQ to PNPIRQ1, which is ORed with the selected IRQ after the IRQ enters the part and before the IRQ goes on to the 8259-based PIC. See Table 78 for IRQ assignment coding.<br>If registers PM00 + C9h-C7h do not select the PNPIRQ[2:0] functions, this field has no effect. | RW |
| 3-0 | IRQ0SEL | 0 | This field assigns an IRQ to PNPIRQ0, which is ORed with the selected IRQ after the IRQ enters the part and before the IRQ goes on to the 8259-based PIC. See Table 78 for IRQ assignment coding.<br>If registers PM00 + C9h-C7h do not select the PNPIRQ[2:0] functions, this field has no effect. | RW |

**Table 78. IRQ Mapping**

| IRQSEL | Interrupt selected | IRQSEL | Interrupt selected |
|---|---|---|---|
| 0000 | Reserved | 1000 | Reserved |
| 0001 | IRQ1 | 1001 | IRQ9 |
| 0010 | Reserved | 1010 | IRQ10 |
| 0011 | IRQ3 | 1011 | IRQ11 |
| 0100 | IRQ4 | 1100 | IRQ12 |
| 0101 | IRQ5 | 1101 | Reserved |
| 0110 | IRQ6 | 1110 | IRQ14 |
| 0111 | IRQ7 | 1111 | IRQ15 |

**Power Management: PNP DMA and Chip Select** **Function 3 Offset 47h–46h**

<table>
<tr><th>Bit</th><th>Name</th><th>Default</th><th>Description</th><th>Access Type</th></tr>
<tr><td>15-13</td><td></td><td>0</td><td>Reserved. Always read 0.</td><td></td></tr>
<tr><td>12</td><td>CS1UBM</td><td>0</td><td>PNPCS1# Upper Mask Bit. If this bit is set, bits 10–8 of the I/O address are masked from PNPCS0 decode for I/O cycles. ISA bus devices, such as super I/O chips, require an external decoding of SA[15:11] = 0h. This bit can be used to extend the mask to bits 10–0, so that PNPCS1# can be used for external decoding functions. In addition, this bit affects Trap Status Bits PMA0[PRM4_TMR_STS] and the System Inactivity Timer Status Bits PMA8[PRM4_TRP_STS]. If this bit is cleared, masking for PNPCS1# is available for only the eight LSBs, as specified by Function 3, Offset CC[MASKIO4]</td><td></td></tr>
<tr><td>12</td><td>CS0UBM</td><td>0</td><td>PNPCS0# Upper Mask Bit. If this bit is set, bits 10–8 of the I/O address are masked from PNPCS0 decode for I/O cycles. ISA bus devices, such as super I/O chips, require an external decoding of SA[15:11] = 0h. This bit can be used to extend the mask to bits 10–0, so that PNPCS0# can be used for external decoding functions. In addition, this bit affects Trap Status Bits PMA0[PRM3_TMR_STS] and the System Inactivity Timer Status Bits PMA8[PRM3_TRP_STS]. If this bit is cleared, masking for PNPCS0# is available for only the eight LSBs, as specified by Function 3, Offset CC[MASKIO3].</td><td></td></tr>
<tr><td>10-9</td><td>IRQ12_SEL</td><td>0</td><td>Pin definition select for IRQ12.<br>These bits specify the function for the IRQ12 as follows:<br><table><tr><th>Bits</th><th>Effect</th></tr><tr><td>00</td><td>IRQ12</td></tr><tr><td>01</td><td>Pin disabled to allow IRQ12 to be controlled by the mouse interrupt.</td></tr><tr><td>10</td><td>SMBALERT# input to the system management logic</td></tr><tr><td>11</td><td>USBOC1#, USB over current 1, to the USB controller</td></tr></table></td><td>RW</td></tr>
<tr><td>8-7</td><td></td><td>0</td><td>Reserved. Always read 0.</td><td></td></tr>
</table>

**Power Management: PNP DMA and Chip Select** **Function 3 Offset 47h–46h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 6-4 | DMASEL | 0 | **PNPDRQ and PNPDAK# DMA Channel Selection.** This field selects the DMA channel for the PNP DMA pins, PNPDRQ and PNPDAK#. Channel 4 is reserved. I/O Mapped Power Management Register, offset CAh is required to select the PNPDRQ function in order for that pin to be connected to the internal DMA controller. I/O Mapped Power Management Register, offset CBh is required to select the PNPDAK# function in order for that pin to be controlled by the internal DMA controller. | RW |
| 3 | CS1M | 0 | **PNPCS1# memory space selection.**<br>0 = PNPCS1# is not selected.<br>1 = PNPCS1# is selected by accesses to the memory addresses specified by programmable memory range monitor 2 specified by Function 3, offset D4h and D8h and PCI accesses to this range are claimed by AMD-756 peripheral bus controller and converted into ISA bus cycles.<br>If the PNPCS1# function is not selected by I/O mapped Power Management, offset C6h, then this bit has no effect. | RW |
| 2 | CS1IS | 0 | **PNPCS1# I/O space selection.**<br>0 = PNPCS1# is not selected.<br>1 = PNPCS1# is selected by accesses to the I/O addresses specified by programmable I/O range monitor 4 specified by Function 3, offset C8h and CCh and PCI accesses to this range are claimed by AMD-756 peripheral bus controller and converted into ISA bus cycles.<br>If the PNPCS1# function is not selected by I/O mapped Power Management, offset C6h, then this bit has no effect. | RW |
| 1 | CS0M | 0 | **PNPCS0# memory space selection.**<br>0 = PNPCS0# is not selected.<br>1 = PNPCS0# is selected by accesses to the memory addresses specified by programmable memory range monitor 1 specified by Function 3, offset D0h and D8h and PCI accesses to this range are claimed by AMD-756 peripheral bus controller and converted into ISA bus cycles.<br>If the PNPCS0# function is not selected by I/O mapped Power Management, offset C5h, then this bit has no effect. | RW |
| 0 | CS0IS | 0 | **PNPCS0# I/O space selection.**<br>0 = PNPCS0# is not selected.<br>1 = PNPCS0# is selected by accesses to the I/O addresses specified by programmable I/O range monitor 3 specified by Function 3, offset C8h and CCh and PCI accesses to this range are claimed by AMD-756 peripheral bus controller and converted into ISA bus cycles.<br>If the PNPCS0# function is not selected by I/O mapped Power Management, offset C5h, then this bit has no effect. | RW |

**Power Management: Pins Latched on the Trailing Edge of Reset** **Function 3 Offset 49h–48h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15-14 | | 00 | **Reserved.** Always reads 0. | |
| 13 | ENIDE | ? | **Enable IDE pull-up/down resistors.** If this bit is set, the internal pullups/pulldowns for IDE bus signals are enabled, which includes pullups on DDATAP[15:0], DATAS[15:0], and pulldowns on DDRQP and DDRQS. If this bit is cleared, the internal pullups/pulldowns for IDE bus signals are disabled. The default for this bit is specified by the state of the SPKR# input signal during reset. | |
| 12 | ENPCI | ? | **Enable PCI Pullup Resistors.** If this bit is set, the internal pullups for PCI bus signals are enabled, which includes pullups on DEVSEL#, FRAME#, IRDY#, PIRQ[A,B,C,D]#, SERR#,STOP#, and TRDY#. If this bit is cleared, the internal pullups for PCI bus signals are disabled. The default for this bit is specified by the state of the SPKR# input signal during reset. | |
| 11 | ENISA | | **Enable ISA Pullup Pulldown Resistors.** If this bit is set, the internal pullups/pulldowns for ISA bus signals are enabled, which includes pullups on IOCHK#, IOR#, IOW#, IRWQ[15,14,12:9, 7:3], LA[23:17], MEMR#, MEMW#, SA[16:0], SBHE#, SD[15:0], SMEMR#, SMEMW#, and pulldowns on DRQ [7:5, 3:0]. If this bit is cleared, the internal pullups/pulldowns for ISA bus signals are disabled. The default for this bit is specified by the state of the SPKR# input signal during reset. | RW |
| 10 | RTCEN | 1 | **Real Time Clock Enable.**<br>This bit determines whether an external or internal realtime clock is selected.<br>0 = (Disabled) An external real time clock is selected.<br>1 = (Enabled) The internal real time clock is enabled.<br>***Note:***<br>*When the internal real time clock is not selected, target accesses to the real time clock are ignored by the internal logic and passed to the ISA bus. Also, the internally-generated IRQ8# cannot become active.*<br>*This bit is reset by RST_SOFT and the value of this bit is retained while in the SOFF state.* | RW |
| 9-6 | RP[16:13] | 0000 | **RP[16:13].**<br>These bits are also accessible through the keyboard controller. They are selected by pull ups or pull downs on DADDRP[2:0] and DCS1P# at the trailing edge of PWRGD reset. These four bits go to the keyboard controller to be accessed through bits [6:3] of the read-input-port command, which is generated by an I/O write to 64h of C0h followed by an I/O read of 60h).<br>The default for these bits is specified by pull up or pull down resistors on pins the specified. They are latched during the trailing edge of reset (PWRGD for all of them except PWRON). | RW |
| 5 | NMLRST | | **Normal Reset Enabled.**<br>1 = A pull up on IOCHRDY during the rising edge of PWRGD, selects the normal 1.8 millisecond pulse reset pulse.<br>0 = A pull down on IOCHRDY during the rising edge of PWRGD, selects fast reset. Used only for simulations and production test. Not intended for use in target systems. | RW |
| 4 | | 0 | **Reserved.** | |
| 3 | KBDE | | **Keyboard Disable.**<br>0 =Disable the internal keyboard controller, the external keyboard controller is selected and the KA20G, KBRC#, EKIRQ1, and EKIRQ12 pin functions are selected.<br>1 = Enable the internal keyboard controller, a pull up on ROM_KBCS# selects the internal keyboard controller on the trailing edge of PWRGD reset; thus, the KBCK, KBDT, MSCK, and MSDT pin functions are selected. | RW |

**Power Management: Pins Latched on the Trailing Edge of Reset** **Function 3 Offset 49h–48h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 2 | PS2E | 1 | **PS/2 Enable.**<br>0 = Disable Internal PS/2 mouse<br>1 = Enable internal PS/2 mouse. | RW |
| 1-0 | | 0 | **Reserved.** (Always reads 0) | |

**Power Management: Serial IRQ Control** **Function 3 Offset 4Ah**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7 | | 0 | **Reserved** | RW |
| 6 | CONTMD | 0 | **CONTMD.**<br>This bit selects the serial IRQ logic to be in quiet or continuous mode. In quiet mode, start frames are initiated by external slave devices. In continuous mode, the start frame is initiated immediately following each stop frame.<br>0 = Quiet mode 1 = Continuous mode | RW |
| 5-2 | FRAMES | 0000 | **Frames.**<br>These bits select the number of 3-clock IRQ frames that the logic will generate during a serial IRQ cycle before issuing the stop frame. | RW |
| 1-0 | StartCLKS | 0 | **Start Clocks.**<br>This bit specifies the number of clocks wide the start pulse over SERIRQ is during the start frame of a serial IRQ cycle (including the slave cycle if in quiet mode). | RW |

| Bits | Number of Frames | Bits | Number of Frames |
|---|---|---|---|
| 0000 | 17 | 1000 | 25 |
| 0001 | 18 | 1001 | 26 |
| 0010 | 19 | 1010 | 27 |
| 0011 | 20 | 1011 | 28 |
| 0100 | 21 | 1100 | 29 |
| 0101 | 22 | 1101 | 30 |
| 0110 | 23 | 1110 | 31 |
| 0111 | 24 | 1111 | 32 |

| Bits | Number of clocks |
|---|---|
| 00 | 4 (default) |
| 01 | 6 |
| 10 | 8 |
| 11 | reserved |

**Power Management: PRDY Timer Control** **Function 3 Offset 4Ch**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-5 | | 0 | **Reserved.** Always reads 0. | |
| 4 | ACPID | 0 | **ACPID.**<br>ACPI power management timer disable. ACPI power management timer specified by I/O Mapped Power Management, offset 08h is disabled while PRDY is active.<br>0 = Enabled 1 = Disabled | RW |
| 3 | SITD | 0 | **SITD.**<br>System inactivity timer disable. System inactivity timer specified by I/O mapped Power Management, offset 98h is disabled from counting while PRDY is active.<br>0 = Enabled 1 = Disabled | RW |
| 2 | | 0 | **Reserved.** Must remain 0 for proper operation. | RW |
| 1 | RTCD | 0 | **RTCD.**<br>Real time clock disable. Real time clock's counters that are clocked off of the 32 KHz clock are disabled from counting while PRDY is active.<br>0 = Enabled 1 = Disabled | RW |
| 0 | PITD | x | **PITD.**<br>Programmable interval timer disable. The clock to all three timers of the internal 8254-compatible PIT are disabled when PRDY is active.<br>0 = Enabled 1 = Disabled | RW |

Each of the bits in the PRDY timer control register controls the ability of the PRDY input signal to disable the AMD-756 controller counters. When PRDY becomes active then the counters that correspond to the set bits in this register stop counting until PRDY becomes inactive. The counters are disabled without glitches on clocks that may cause unpredictable behavior. If the PRDY function of the KEYLOCK pin is not selected by I/P Mapped Power Management, offset CDh, then this register has no effect.

**Power Management: Square Wave Generation** **Function 3 Offset 4Eh**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-4 | | 0 | **Reserved .** Always reads 0. | |
| 3-0 | | 0 | **SQWAVE.**<br>Square wave frequency control. When PM00 +D0h selects the square wave output function, this field is used to specify the frequency of the square wave output on the INTIRQ8# pin. The square wave output is generated by dividing down the 32 KHz. clock. Here is how this field is decoded: | RW |

| Bits | Frequency (Hz) | Bits | Frequency (Hz) |
|---|---|---|---|
| 0000 | Output low | 1000 | 256 |
| 0001 | 32768 | 1001 | 128 |
| 0010 | 16384 | 1010 | 64 |
| 0011 | 8192 | 1011 | 32 |
| 0100 | 4096 | 1100 | 16 |
| 0101 | 2048 | 1101 | 8 |
| 0110 | 1024 | 1110 | 4 |
| 0111 | 512 | 1111 | 2 |

**Power Management: Power State Pin Control** **Function 3 Offset 53h - 50h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31 | APICEN | 0 | **APIC Interrupt Message Bus PICCLK Enable During POS State.** If this bit is set, the operation of the PICCLK during the POS state is allowed to continue. If this bit is cleared, PICCLK is driven Low after the stop-grant cycle while going into the POS suspend state. The PICCLK starts clocking again as soon as the resume event is detected. | |
| 30-29 | | 0 | **Reserved.** These bits must remain 0 for proper operation. | RW |
| 28 | TTHEN | 0 | **TTH Enable.** Thermal throttling enable**.**<br>0 = Disabled<br>1 = (Enabled) (and if I/O Mapped Power Management, offset C2h selects the THERM# function for the pin, and the output of the GPIO2 input path is high), then thermal throttling (duty cycle specified by the TTH_RATIO field) is enabled.<br>Thermal throttling has absolute priority over normal throttling (see I/O mapped Power Management, offset 10h); but it will not be invoked if the system is in C2, C3, POS, or SOFF. | RW |

**Power Management: Power State Pin Control (continued)** **Function 3 Offset 53h - 50h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 27-25 | TTH_RATIO | 000 | **Thermal Throttle Ratio.**<br>Thermal throttling duty cycle. These bits specify the duty cycle of the STPCLK# signal to the CPU when the system is in thermal throttling mode (initiated by the THERM# pin when enabled by TTH_EN). The field is decoded as follows:<br>**Ratio Bits** / **Description**<br>000 — Reserved<br>001 — 12.5% STPCLK# active<br>010 — 25.0% STPCLK# active<br>011 — 37.5% STPCLK# active<br>100 — 50.0% STPCLK# active<br>101 — 62.5% STPCLK# active<br>110 — 75.0% STPCLK# active<br>111 — 87.5% STPCLK# active | RW |
| 24 | PITRSML | 0 | **PIT Enable.** 1 = PIT will not generate IRQ0 to the PIC while in POS. This is necessary for timer tick events from resuming the system while in POS.<br>0 = PIT will generate timer tick interrupts to the the PIC while in POS.. | RW |
| 23 | CRST_POSEN | 0 | **CRST_POS Enable.**<br>0 = (Disabled) CPURST# is not asserted during the transition.<br>1 = (Enabled) The assertion of CPURST# during the transition from POS to FON is enabled.<br>This bit must not be set unless the corresponding SUSP_POSE (see below) bit is set (i.e., CPU resets are only allowed if SUSPEND# gets asserted). | RW |
| 22 | SUSP_POSEN | 0 | **SUSP_POS Enable.**<br>0 = (Disabled) SUSPEND# is always high.<br>1 = (Enabled) SUSPEND# is asserted during the POS state.<br>This bit has no effect if the I/O Mapped Power Management, offset C4h does not select the SUSPEND# function. | RW |
| 21 | SLPP_POSEN | 0 | **SLPP_POS Enable.**<br>0 = (Disabled) CPUSLEEP# is always high.<br>1= (Enabled) CPUSLEEP# assertion to the CPU during the POS state is enabled.<br>This bit has no effect if the I/O Mapped Power Management, offset C5h does not select the CPUSLEEP# function. | RW |
| 20 | PSTP_POSEN | 0 | **PSTP_POS Enable.**<br>0 = (Disabled) PCISTOP# is always high.<br>1 = (Enabled) PCISTOP# assertion to the external PLL during the POS state is enabled.<br>This bit has no effect if the I/O Mapped Power Management, offset C7h does not select the PCISTOP# function. | RW |

**Power Management: Power State Pin Control (continued)** **Function 3 Offset 53h - 50h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 19 | CSTP_POSEN | 0 | **CSTP_POS Enable.**<br>0 = (Disabled) CPUSTOP# is always high.<br>1 = (Enabled) CPUSTOP# assertion to the external PLL during the POS state is enabled.<br>This bit has no effect if the I/O Mapped Power Management, offset C6h register does not select the CPUSTOP# function. | RW |
| 18 | POSEN | 0 | **POS Enable.**<br>0 = (Disabled) STPCLK# will not be asserted during the POS state.<br>1 = (Enabled) STPCLK# assertion during the POS state is enabled.<br>This bit must be set high for any other bit in this byte register to function (i.e., if STPCLK# is not asserted for POS, then none of the other power management control signals can be asserted for POS). | RW |
| 17 | DCST_POSEN | 0 | **DCST_POS Enable.**<br>0 = (Disabled) DCSTOP# is always high.<br>1 = (Enabled) DCSTOP# assertion to the DRAM controller during the POS state is enabled.<br>This bit has no effect if the I/O Mapped Power Management, offset C9h register does not select the DCSTOP# function. | RW |
| 16 | ZZ_POSEN | 0 | **ZZ_POS Enable.**<br>0 = (Disabled) CACHE_ZZ is always low.<br>1 = (Enabled) CACHE_ZZ assertion to the L2 cache during the POS state is enabled.<br>This bit has no effect if the I/O Mapped Power Management, offset C8h register does not select the CACHE_ZZ function. | RW |
| 15 | CRST_C3EN | 0 | **CRST_C3 Enable.**<br>0 = (Disabled) CPURST# is not asserted during the transition.<br>1 = (Enabled) The assertion of CPURST# during the transition from C3 to FON is enabled.<br>It is not legal to set this bit unless the corresponding SUSP_C3E bit is set (i.e., CPU resets are only allowed if SUSPEND# gets asserted). | RW |
| 14 | SUSP_C3EN | 0 | **SUSP_C3 Enable.**<br>0 = (Disabled) SUSPEND# is always high.<br>1 = (Enabled) SUSPEND# assertion during the C3 state is enabled.<br>This bit has no effect if the I/O Mapped Power Management, offset C4h does not select the SUSPEND# function. | RW |
| 13 | SLPP_C3EN | 0 | **SLPP_C3 Enable.**<br>0 = (Disabled) CPUSLEEP# is always high.<br>1 = (Enabled) CPUSLEEP# assertion to the CPU during the C3 state. is enabled.<br>This bit has no effect if the I/O Mapped Power Management, offset C5h does not select the CPUSLEEP# function. | RW |
| 12 | PSTP_C3EN | 0 | **PSTP_C3 Enable.**<br>0 = (Disabled) PCISTOP# is always high.<br>1 = (Enabled) PCISTOP# assertion to the external PLL during the C3 state is enabled.<br>This bit has no effect if the I/O Mapped Power Management, offset C7h does not select the PCISTOP# function. | RW |

**Power Management: Power State Pin Control (continued)** **Function 3 Offset 53h - 50h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 11 | CSTP_C3EN | 0 | **CSTP_C3 Enable.**<br>0 = (Disabled) CPUSTOP# is always high.<br>1 = (Enabled) CPUSTOP# assertion to the external PLL during the C3 state is enabled.<br>This bit has no effect if the I/O Mapped Power Management, offset C6h register does not select the CPUSTOP# function. | RW |
| 10 | C3EN | 0 | **C3 Enable.**<br>0 = (Disabled) STPCLK# will not be asserted during the C3 state.<br>1 = (Enabled) STPCLK# assertion during the C3 state is enabled.<br>This bit must be set high for any other bit in this byte register to function (i.e., if STPCLK# is not asserted for C3, then none of the other power management control signals can be asserted for C3). | RW |
| 9 | DCST_C3EN | 0 | **DCST_C3 Enable.**<br>0 = (Disabled) DCSTOP# is always high.<br>1 = (Enabled) DCSTOP# assertion to the DRAM controller during the C3 state. is enabled.<br>This bit has no effect if the I/O Mapped Power Management, offset C9h register does not select the DCSTOP# function. | RW |
| 8 | ZZ_C3EN | 0 | **ZZ_C3 Enable.**<br>0 = (Disabled) CACHE_ZZ is always low.<br>1 = (Enabled) CACHE_ZZ assertion to the L2 cache during the C3 state is enabled.<br>This bit has no effect if the PM00 +C8h register does not select the CACHE_ZZ function. | RW |
| 7 | CRST_C2EN | 0 | **CRST_C2 Enable.**<br>0 = (Disabled) CPURST# is not asserted during the transition.<br>1 = (Enabled) the assertion of CPURST# during the transition from C2 to FON is enabled.<br>It is not legal to set this bit unless the corresponding SUSP_C2E bit is set (i.e., CPU resets are only allowed if SUSPEND# gets asserted). | RW |
| 6 | SUSP_C2EN | 0 | **SUSP_C2 Enable.**<br>0 = (Disabled) SUSPEND# is always high.<br>1 = (Enabled) SUSPEND# assertion during the C2 state is enabled.<br>This bit has no effect if the I/O Mapped Power Management, offset C4h does not select the SUSPEND# function. | RW |
| 5 | SLPP_C2EN | 0 | **SLPP_C2 Enable.**<br>0 = (Disabled) CPUSLEEP# is always high.<br>1 = (Enabled) CPUSLEEP# assertion to the CPU during the C2 state is enabled.<br>This bit has no effect if the I/O Mapped Power Management, offset C5h does not select the CPUSLEEP# function. | RW |
| 4 | PSTP_C2EN | 0 | **PSTP_C2 Enable.**<br>0 = (Disabled) PCISTOP# is always high.<br>1 = (Enabled) PCISTOP# assertion to the external PLL during the C2 state is enabled.<br>This bit has no effect if the I/O Mapped Power Management, offset C7h does not select the PCISTOP# function. | RW |

**Power Management: Power State Pin Control (continued)** **Function 3 Offset 53h - 50h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 3 | CSTP_C2EN | 0 | **CSTP_C2 Enable.** 0 = (Disabled) CPUSTOP# is always high.<br>1 = (Enabled) CPUSTOP# assertion to the external PLL during the C2 state is enabled.<br>This bit has no effect if the I/O Mapped Power Management, offset C6h register does not select the CPUSTOP# function. | RW |
| 2 | C2EN | 0 | **C2 Enable.**<br>0 = (Disabled) STPCLK# will not be asserted during the C2 state.<br>1 = (Enabled) STPCLK# assertion during the C2 state is enabled.<br>This bit must be set high for any other bit in this byte register to function (i.e., if STPCLK# is not asserted for C2, then none of the other power management control signals can be asserted for C2). | RW |
| 1 | DCST_C2EN | 0 | **DCST_C2 Enable.**<br>0 = (Disabled) DCSTOP# is always high.<br>1 = (Enabled) DCSTOP# assertion to the DRAM controller during the C2 state is enabled.<br>This bit has no effect if the I/O Mapped Power Management, offset C9h register does not select the DCSTOP# function. | RW |
| 0 | ZZ_C2EN | 1 | **ZZ_C2 Enable.**<br>0 = (Disabled) CACHE_ZZ is always low.<br>1 = (Enabled) CACHE_ZZ assertion to the L2 cache during the C2 state is enabled.<br>This bit has no effect if the I/O Mapped Power Management, offset C8h register does not select the CACHE_ZZ function. | RW |

The first three bytes of this register specify the output pins to be controlled to enter and exit the C2, C3, and Power On Suspend (POS) states; each byte provides the enables for each of these three low-power states. C2 is defined by the enables in bits[7:0], C3 is defined by the enables in bits [15:8], and POS is defined by the enables in bits[23:16].

The fourth byte provides automatic thermal throttling based on CPU over-temperature detection.

**Power Management: PCI Edge or Level Select** **Function 3 Offset 54h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-4 | | 0 | **Reserved.** Always reads 0. | RW |
| 3 | EDGEPID | 0 | **Edge Triggered Interrupt Select for PCI Interrupt D.** This bit controls the polarity of the PCI interrupt pin PIRQ[D]#.<br>0 = PIRQ[D] is active Low and level triggered, which is the normal PCI-compliant mode.<br>1= PIRQ[D] active High and edge triggered, which is not compliant with PCI, but is typical of ISA interrupts. | RW |
| 2 | EDGEPIC | | **Edge Triggered Interrupt Select for PCI Interrupt C.** This bit controls the polarity of the PCI interrupt pin PIRQ[C]#.<br>0 = PIRQ[C] is active Low and level triggered, which is the normal PCI-compliant mode.<br>1= PIRQ[C] active High and edge triggered, which is not compliant with PCI, but is typical of ISA interrupts. | RW |
| 1 | EDGEPIB | 0 | **Edge Triggered Interrupt Select for PCI Interrupt B.** This bit controls the polarity of the PCI interrupt pin PIRQ[B]#.<br>0 = PIRQ[B] is active Low and level triggered, which is the normal PCI-compliant mode.<br>1= PIRQ[B] is active High and edge triggered, which is not compliant with PCI, but is typical of ISA interrupts. | RW |
| 0 | EDGEPA | 0 | **Edge Triggered Interrupt Select for PCI Interrupt A.** This bit controls the polarity of the PCI interrupt pin PIRQ[A]#.<br>0 = PIRQ[A] is active Low and level triggered, which is the normal PCI-compliant mode.<br>1= PIRQ[A] active High and edge triggered, which is not compliant with PCI, but is typical of ISA interrupts. | RW |

**Power Management: PCI IRQ Routing** **Function 3 Offset 57h–56h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15-12 | PIRQDSEL | 0 | **PIRQD# Select.** These bits map the PCIIRQD# pin to the internal ISA-bus compatible interrupt controller. This field is decoded as shown in Table 78. | RW |
| 11-8 | PIIRQCSEL | 0 | **PIRQC# Select.** These bits map the PCIIRQC# pin to the internal ISA-bus compatible interrupt controller. This field is decoded as shown in Table 78. | RW |
| 7-4 | PIRQBSEL | 0 | **PIRQB# Select.** These bits map the PCIIRQB# pin to the internal ISA-bus compatible interrupt controller. This field is decoded as shown in Table 78. | RW |
| 3-0 | PIRQASEL | 0 | **PIRQA# Select.** These bits map the PCIIRQA# pin to the internal ISA-bus compatible interrupt controller. This field is decoded as shown in Table 78. | RW |

**Power Management: System Management I/O Space Pointer** **Function 3 Offset 5Bh–58h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-16 | | 0 | **Reserved.** Must remain 0 for proper operation. | RW |
| 15-8 | | 11011101 | **PMBASE.**<br>These bits specify the PCI address bits[15:8] of the 256-byte block of I/O-mapped registers used for system management (address space I/O Mapped Power Management). Access to this address space is enabled by Function 3 Offset 41 bit 7 [PMIOEN]. | RW |
| 7-0 | | 00000001 | **PM Base LSB.** Always reads 01h | RO |

**Power Management: System Management Class Code** **Function 3 Offset 63h–60h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-8 | CCWRITE | 0 | **CCWRITE.** Writes to this register are latched and stored so that they can be read from the CLASSCODE field of Function 3, offsets 0Bh-09h. | RW |
| 7-0 | | 0 | **Reserved.** Must remain 0 for proper operation. | RW |

**Power Management: Serial Port Trap Address** **Function 3 Offset A3h–A0h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-16 | ADDRCB | 00000010 01111000 | **ADDRCB.** Address for the COMB trap event (default 0278h). Function 3, offset A0 along with function 3, offset A4 define the address for COMA and COMB trap events. These events can be used to generate an SMIs or SCIs, load the associated re-trigger timers (I/O Mapped Power Management, offset 58h and offset 5Ch), load the system inactivity timer, or load the burst timers. | RW |
| 15-0 | ADDRCA | 00000011 11111000 | **ADDRCA.** Address for the COMA trap event. (default 03F8h). | RW |

**Power Management: Serial Port Trap Mask** **Function 3 Offset A7h–A4h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15–8 | MASKCB | 00001111 | **MASKCB.** Mask for the COMB trap event (default 0Fh). | RW |
| 7-0 | MASKCA | 00001111 | **MASKCA.** Mask for the COMA trap event. (default 0Fh). | RW |

**Power Management: Audio Port 2 and 1 Trap Address** **Function 3 Offset ABh–A8h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-16 | ADDRAUD2 | 00000011 00110000 | **ADDRAUD2 .** Address for the audio trap event #2 (default 0330h). | RW |
| 15–0 | ADDRAUD1 | 00000010 00100000 | **ADDRAUD1.** Address for the audio trap event #1 (default 0220h). Function 3, offset A8, function 3, offset AC, and function 3, offset B0 combine to define the audio trap event. This event can be used to generate an SMIs or SCIs, load the associated re-trigger timer (I/O Power Management, offset 60h), load the system inactivity timer, or load the burst timers. | RW |

**Power Management: Audio Port 4 and 3 Trap Address** **Function 3 Offset AFh–ACh**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-16 | ADDRAUD4 | 00000011 10001000 | **ADDRAUD4.** Address for the audio trap event #2 (default 0388h). | RW |
| 15-0 | ADDRAUD3 | 00000101 00110000 | **ADDRAUD3.** Address for the audio trap event #1 (default 0530h). | RW |

**Power Management: Audio Port Trap Mask** **Function 3 Offset B3h–B0h**

| Bit | Name | Default | Description | 0-state |
|---|---|---|---|---|
| 31-24 | MASKAUD4 | 00000111 | **MASKAUD4.** Mask for the audio trap event #4 (default 07h). | RW |
| 23-16 | MASKAUD3 | 00000111 | **MASKAUD3.** Mask for the audio trap event #3 (default 07h). | RW |
| 15-8 | MASKAUD2 | 00000001 | **MASKAUD2.** Mask for the audio trap event #2 (default 01h). | RW |
| 7-0 | MASKAUD1 | 00001111 | **MASKAUD1.** Mask for the audio trap event #1 (default 0Fh). | RW |

**Power Management: PCMCIA 1 and 2 I/O Trap Address** **Function 3 Offset B7h–B4h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-16 | ADDRPIO2 | 00000000 00000000 | **ADDRPIO2.**<br>Address for the PCMCIA trap event #2 (default 0000h).<br>Function 3, offset B4h, function 3, offset B8h, function 3, offset BCh, and function 3, offset C0h combine to define the address for the PCMCIA1 and PCMCIA2 trap events. These events can be used to generate SMIs or SCIs, load the associated re-trigger timers (I/O Mapped Power Management, offset 6Ch and offset 70h), load the system inactivity timer, or load the burst timers. | RW |
| 15–0 | ADDRPIO1 | 00000000 00000000 | **ADDRPIO1.**<br>Address for the PCMCIA trap event #1 (default 0000h). | RW |

**Power Management: PCMCIA Trap 1 Memory Address** **Function 3 Offset BBh–B8h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-10 | ADDRPME1 | 00000000 00000000 000000 | **ADDRPME1.**<br>Memory address for the PCMCIA1 trap event. | RW |
| 9-0 | | 0 | **Reserved.** These bits must remain 0 for proper operation. | RW |

**Power Management: PCMCIA Trap 2 Memory Address** **Function 3 Offset BFh–BCh**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-10 | ADDRPME2 | 00000000 00000000 00 | **ADDRPME2.**<br>Memory address for the PCMCIA2 trap event (default 0000h). | RW |
| 9-0 | | 0 | **Reserved.** These bits must remain 0 for proper operation. | RW |

**Power Management: PCMCIA Trap Mask** **Function 3 Offset C3h–C0h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-24 | MASKPME2 | 00000000 | **MASKPME2.** Mask for the PCMCIA trap event #2 (default 00h). | RW |
| 23-16 | MASKPME1 | 00000000 | **MASKPME1.** Mask for the PCMCIA trap event #1(default 00h). | RW |
| 15-8 | MASKPIO2 | 00000000 | **MASKPIO2.** Mask for the PCMCIA trap event #2 (default 00h). | RW |
| 7-0 | MASKPIO1 | 00000000 | **MASKPIO1.** Mask for the PCMCIA trap event #1 (default 00h). | RW |

**Power Management: I /O Range Monitor 2 and 1 I/O Trap Address** **Function 3 Offset C7h–C4h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-16 | ADDRIO2 | 00000000 00000000 | **ADDRIO2.** Address for the I/O trap event #2 (default 0000h).<br>Function 3, offset C4h, function 3, offset C8h, function 3, and offset CCh, combine to define the address for the programmable I/O range monitor trap events. These events can be used to generate SMIs or SCIs, load the associated re-trigger timers (I/O Mapped Power Management, offset 78h, offset 7Ch, offset 80h, and offset 84h), load the system inactivity timer, or load the burst timers. | RW |
| 15-0 | ADDRIO1 | 00000000 00000000 | **ADDRIO1.** Address for the I/O trap event #1 (default 0000h). | RW |

**Power Management: I/O Range Monitor 4 and 3 I/O Trap Address** **Function 3 Offset CBh–C8h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-16 | ADDRIO4 | 00000000 00000000 | **ADDRIO4.** Address for the I/O trap event #4 (default 0000h). | RW |
| 15-0 | ADDRIO3 | 00000000 00000000 | **ADDRIO3.** Address for the I/O trap event #3 (default 0000h). | RW |

**Power Management: I/O Range Monitor Trap Mask** **Function 3 Offset CFh–CCh**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-24 | MASKIO4 | 00000000 | **MASKIO4.** Mask for the I/O trap event #4 (default 00h). | RW |
| 23-16 | MASKIO3 | 00000000 | **MASKIO3.** Mask for the I/O trap event #3(default 00h). | RW |
| 15-8 | MASKIO2 | 00000000 | **MASKIO2.** Mask for the I/O trap event #2 (default 00h). | RW |
| 7-0 | MASKIO1 | 00000000 | **MASKIO1.** Mask for the I/O trap event #1 (default 00h) | RW |

**Power Management: Memory Range Monitor Trap 1 Address** **Function 3 Offset D3h–D0h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-8 | ADDRMEM1 | 00000000 00000000 00000000 | **ADDRMEM1.** Memory address for the PMEMRM1 trap event (default 000000h). | RW |
| 7-0 | | 00000000 | **Reserved.** These bits must remain 0 for proper operation. | RW |

**Power Management: Memory Range Monitor Trap 2 Address** **Function 3 Offset D3h–D0h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-8 | ADDRMEM2 | 00000000 00000000 00000000 | **ADDRMEM2.**<br>Memory address for the PMEMRM2 trap event (default 0000h).<br>Function 3, offset D0h, function 3, offset D4h, and function 3, offset D8h combine to define the address for the programmable memory range monitor 1 and 2 trap events (PMEMRM[1,2]). These events can be used to generate SMIs or SCIs, load the associated re-trigger timers (I/O Mapped Power Management, offset 88h and offset 8Ch), load the system inactivity timer, or load the burst timers. These trap events occur when the following equations are true:<br>PMEMRM1: AD[31:8] + MASKMEM1 == ADDRMEM1 + MASKMEM1;<br>PMEMRM2: AD[31:8] + MASKMEM2 == ADDRMEM2 + MASKMEM2;<br>Where AD is the address phase of a PCI bus memory cycle, it is not necessary for the cycle to be targeted at the AMD-756. The mask bits for the memory addresses can cover bits AD[23:8]. | RW |
| 7-0 | | 00000000 | **Reserved.** These bits must remain 0 for proper operation. | RW |

**Power Management: Memory Range Monitor Trap Mask** **Function 3 Offset DBh–D8h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-16 | MASKMEM2 | 00000000 00000000 | **MASKMEM2.** Mask for the memory range trap event #2 (default 0000h). | RW |
| 15-0 | MASKMEM1 | 00000000 00000000 | **MASKMEM1.** Mask for the memory range trap event #1 (default 0000h). | RW |

## 7.7.3 Power Management I/O Space Registers

The following power management I/O mapped registers are accessed through function 3 offset 58 base pointer register. Throughout this section the I/O Mapped Power Management base register will be referred to as PM00.

**Power Management: Status** **I/O Mapped Offset 01h–00h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15 | WS | 0 | **Wakeup Status.** This bit is set when the system is in the POS state and an enabled resume event occurs as defined in PM00 +16h. Upon setting this bit, the system automatically transitions from the POS state to the normal working state (from C3 to FON for the processor). | RWC |
| 14-12 | | 000 | **Reserved.** Always reads 0. | |
| 11 | PBOS | 0 | **Power Button Override Status.** This bit is set when the PWRBTN# input pin is continuously asserted for more than 4 seconds. The setting of this bit resets the PB_STS bit and transitions the system into the soft off state. This bit is reset by RST_SOFT and the value of this bit is retained while in the SOFF state. | RWC |
| 10 | RTCS | 0 | **RTC Status.** This bit is set by hardware when the real time clock generates an alarm interrupt. If the external real time clock is enabled, then this bit is set when EXTIRQ8# is asserted. EXTIRQ8# is muxed with the SLPBTN# pin; the EXTIRQ8# function must be selected for the pin to cause the bit to become high. This bit is reset by RST_SOFT and the value of this bit is retained while in the SOFF state. | RWC |
| 9 | SLP | 0 | **Sleep Button Status.** When high, indicates that the sleep button SLPBTN# has been asserted. The debounce circuitry causes a 12-to-16 millisecond delay from the time the input signal stabilizes until this bit changes. If the GPIO debounce circuitry selected by PM00 +C3h is enabled, then the signal will be debounced twice before this bit is set. This bit is reset by RST_S is enabled (PM00 +C3h), then there is an OFT and the value of this bit is retained while in the SOFF state. | RWC |
| 8 | PBS | 0 | **Power Button Status.** This bit is set when the PWRBTN# signal is asserted. The debounce circuitry causes a 12- to 16-millisecond delay from the time the input signal stabilizes until this bit changes. If PWRBTN# is held low for more than four seconds, then this bit is cleared and PBOR_STS is set, and the system transitions into the soft off state. This bit is reset by RST_SOFT and the value of this bit is retained while in the SOFF state. | RWC |
| 7-6 | | 0 | **Reserved.** Always reads 0. | |
| 5 | GS | 0 | **Global Status.** This bit is set by hardware when PM00 +2C bit[BIOS_RLS] is set (typically by an SMI routine to release control of the SCI/SMI lock). If enabled by PM00 +02 bit[GBL_EN], this can be used to generate an SCI/SMI interrupt. | RWC |
| 4 | BMS | 0 | **Bus Master Status.** This bit is set by hardware when either FRAME# or BMREQ# becomes active, or any internal PCI master requests the PCI bus, based on the state of the PM00 +CCh selection register-while in the C3 power state active. | RWC |
| 3-1 | | 000 | **Reserved .** Always reads 0. | |
| 0 | TMS | 0 | **Timer Carry Status.** This bit is set when the 23rd (or 31st) bit of the 24-bit (or 32-bit) ACPI power management timer PM00 +08h changes. | RWC |

The bits in this register are set only by hardware and can be reset by software by writing a one to the desired bit position.

**Power Management: Enable** **I/O Mapped Offset 03h–02h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15-11 | | 0 | **Reserved.** Always reads 0. | |
| 10 | RTC_EN | 0 | **RTC Enable.**<br>This bit can be set to trigger either an SCI or an SMI, depending on the setting of register PM00 +04h, [SCI_EN] bit 0 when register PM00, [RTS_STS] bit 10 is set high. | RW |
| 9 | SLPBTN_EN | 0 | **Sleep Button SCI/SMI Enable.**<br>This bit can be set to generate either an SCI or an SMI, depending on the state of register PM00 +0h4, bit[SCI_EN] when register PM00, [SLPBTN_STS] is set high. | RW |
| 8 | PB_EN | 0 | **Power Button SCI/SMI Enable.**<br>This bit can be set to trigger either an SCI or an SMI, depending on the state of register PM00 +0h4, [SCI_EN] bit 0 when PM00[PWRBTN_STS] is set high. | RW |
| 7-6 | | 0 | **Reserved.** Always reads 0. | |
| 5 | GBL_EN | 0 | **Global SCI/SMI Enable.**<br>This bit can be set to generate either an SCI or an SMI, depending on the state of PM00 +04h[SCI_EN] bit 0 when PM00[GBL_STS] is set high | RW |
| 4-1 | | 0 | **Reserved.** Always reads 0. | |
| 0 | TMR_EN | 0 | **ACPI Timer Enable.**<br>This bit can be set to generate either an SCI or an SMI, depending on the state of PM00 +04 [SCI_EN] bit 0 when PM00[TMR_STS] bit 0 is set high. | RW |

The bits in this register correspond to the bits in the Power Management Status Register at Function 3, offset 01h–00h.

**Power Management: Control** **I/O Mapped Offset 05h–04h**

<table>
<tr><th>Bit</th><th>Name</th><th>Default</th><th>Description</th><th>Access Type</th></tr>
<tr><td>15-14</td><td></td><td>00</td><td>Reserved. Always reads 0.</td><td></td></tr>
<tr><td>13</td><td>SE</td><td>0</td><td>Sleep Enable.<br>Reads from this bit always return zero. Writing a one to this bit causes the system to sequence into the sleep state defined by the Sleep_Type field, bits 12–10.</td><td>WO</td></tr>
<tr><td>12-10</td><td></td><td>000</td><td>Sleep Type.
<table>
<tr><th>Bits</th><th>Sleep State Setting</th></tr>
<tr><td>000</td><td>Soft Off (also called Suspend to Disk).<br>The VDD3 power plane is turned off while the VDD-SOFT and VDD-RTC (VBAT) planes remain on.</td></tr>
<tr><td>001</td><td>Reserved</td></tr>
<tr><td>010</td><td>Reserved</td></tr>
<tr><td>011</td><td>Reserved</td></tr>
<tr><td>100</td><td>Power On Suspend.<br>All power planes remain on but the processor is put into the C3 state. System context is maintained.<br>System transitions into specified POS state potentially including all C3 clock controls and assertion of the SUSPEND# signal.</td></tr>
<tr><td>101</td><td>Normal mode.<br>Software can place this field into this state before entering one of the clock controlled states (C2 or C3) to store the system state for resume. This does not cause any action by the hardware.</td></tr>
<tr><td>110</td><td>Reserved</td></tr>
<tr><td>111</td><td>Reserved</td></tr>
</table></td><td>RW</td></tr>
<tr><td>9-3</td><td></td><td>0</td><td>Reserved . Always reads 0.</td><td></td></tr>
<tr><td>2</td><td>GBL_RLS</td><td>0</td><td>Global Release.<br>This bit is set by ACPI software to indicate the release of the SCI/SMI lock. When this bit is set, hardware automatically sets the PM28[BIOS_STS] bit. GBL_RLS is cleared by hardware when the BIOS_STS bit is cleared by software.<br>0 = Disabled 1 = Enabled</td><td>RW</td></tr>
<tr><td>1</td><td>BMR</td><td>0</td><td>Bus Master Reload.<br>This bit is used to enable the occurrence of a bus master request to transition the processor from the C3 state to the FON state.<br>0 = Disabled 1 = Enabled</td><td>RW</td></tr>
<tr><td>0</td><td>SCI_EN</td><td>0</td><td>SCI Enable.<br>This bit determines whether a power management event generates an SCI or SMI.<br>0 = Generate SMI, to execute platform specific software.<br>1 = Generate SCI, mapped to a traditional interrupt via Function 3 offset 42h and used by the operating system.</td><td>RW</td></tr>
</table>

**Power Management: ACPI Timer** **I/O Mapped Offset 0Bh–08h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-24 | ETM_VAL | 00000000 | **Extended Timer Value.**<br>When register in function 3 offset 41 bit 3 is high, these are the 8 MSBs of the ACPI power management timer. When bit 3 is low, this field always reads back as all zeros. | RW |
| 23-0 | TMR_VAL | 00000000<br>00000000<br>00000000 | **Timer Value.**<br>This field returns the running count of the power management timer. The timer is reinitialized to zero during a reset and continues counting until the 14.31818 MHz input to the chip is stopped. | RO |

This is either a 24- or a 32-bit counter, based on the state of register in function 3 offset 41 bit 3. The timer is a free-running up counter that is clocked off of a 3.579545 MHz clock. It does not count when in the system is in SOFF. When the MSB toggles (either bit[23] or bit[31]) then a power management event is generated. Refer to PM00[TMR_STS] and PM00 +02 bit[TMR_EN].

## 7.7.4 Processor Power Management Registers

**Power Management: CPU Clock Control** **I/O Mapped Offset 13h–10h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-5 | | 0 | **Reserved .** Always reads 0. | |
| 4 | NHT_EN | 0 | **Normal Throttling Enable.**<br>When high, normal throttling (duty cycle specified by bits [3:1]) is enabled. Normal throttling is lower priority than thermal throttling (as specified by function 3 offset 50); when thermal throttling is enabled, the throttling duty cycle is specified by function 3 offset 50. Throttling is disabled when in the C2, C3, POS, or SOFF states. When bit 4 is low, normal throttling is disabled. If STPCLK# = high, system is running. If STPCLK# = low; system is in the C3 state.<br>0 = STPCLK# normal<br>1 = STPCLK# throttled | RW |
| 3-1 | NHT_DTY | 000 | **Normal Throttling Duty Cycle.**<br>This 3-bit field determines the duty cycle of the STPCLK# signal when the system is in normal throttling mode.<br>Bits / STPCLK# Duty Cycle:<br>000 Reserved<br>001 12.5% active<br>010 25% active<br>011 37.5% active<br>100 50% active<br>101 62.5% active<br>110 75% active<br>111 87.5% active | RW |
| 0 | | 0 | **Reserved.** Always reads 0. | |

**Power Management: Processor Level 2** **I/O Mapped Offset 14h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | P_LVL2 | 00000000 | **Processor Level 2 Register.**<br>Reads from this register put the processor into C2 power state by asserting STPCLK# to suspend the processor. Reads from this register return 00h. Writes to this register have no effect. Wakeup from the Stop Clock state is done by returning from an interrupt (INTR, SMI, PWRBTN#, RTC wakeup, or when the SCI pin toggles). | RO |

**Power Management: Processor Level 3** **I/O Mapped Offset 15h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7 | P_LVL3 | 00000000 | **Processor Level 3 Register.**<br>Reads from this register put the processor in the C3 clock state with the STPCLK# signal asserted. Reads from this register return 00h. Writes to this register have no effect. Wakeup from the Stop Clock state is done by returning from an interrupt (INTR, SMI, PWRBTN#, RTC wakeup, or when the SCI pin toggles). | RO |

**Power Management: Resume Event Enable** **I/O Mapped Offset 17h-16h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15 | | 0 | **Reserved.** Always reads 0. | |
| 14 | RS14 | 0 | **Resume on Assertion of RI#.** Setting PM00 +28 bit [RI_STS]; this bit does not get set unless the RI# function is selected by PM00 +C1A. | RW |
| 13-11 | | 000 | **Reserved.** Always reads 0. | |
| 10 | RS10 | 0 | **Resume on system inactivity timer time out.** Setting PM00 +28h bit [SIT_STS]. | RW |
| 9 | RS9 | 0 | **Resume on assertion of IRQ8#.** Setting PM00 [RTC_STS]; regardless whether the RTC is internal or external. | RW |
| 8 | RS8 | 0 | **Resume on assertion of an unmasked IRQ.** When either bit of INTR[1:0] to the CPUs is set. | RW |
| 7 | RS7 | 0 | **Resume on USB-defined resume event.** Setting PM00 +24 bit [USB_RSM_STS]. | RW |
| 6 | RS6 | 0 | **Resume on the assertion of the SLPBTN#.** Setting PM00 [SLPBTN_STS]; this bit is not set unless the SLPBTN# function is selected by PM00 +C3h. | RW |
| 5 | RS5 | 0 | **Resume on assertion of SMBALERT#.** This function is disabled if the SMBALERT# function of the IRQ12 pin is not enabled by function 3 offset 46[IRQ12_SEL]. | RW |
| 4 | RS4 | 0 | **Resume on an SMBus master access to the SMBus host slave address.**<br>Setting PM00 +E0h bit [HSLV_STS]. | RW |
| 3 | RS3 | 0 | **Resume on an SMBus access match to the snoop address.**<br>Setting PM00 +E0h bit [SNP_STS]. | RW |
| 2 | RS2 | 0 | **Resume on assertion of EXTSMI# .** Setting PM00 +28h bit[EXTSMI_STS]; this bit is not set unless the EXTSMI# function is selected by PM00 +CCh. | RW |
| 1 | RS1 | 0 | **Resume on assertion of PME#.** Setting PM00 +28h bit[PME_STS]. | RW |
| 0 | RS0 | 0 | **Resume on assertion of PWRBTN#.** Setting PM00 bit[PWRBTN_STS]<br>0 = No action 1 = Resume on specified event. | RW |

**Power Management: Flag Write** **I/O Mapped Offset 18h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15-0 | FWRDATA | 00000000 00000000 | **Flag Write Data.**<br>Writes to this register are passed to the ISA data bus to be latched by external '373- like devices with the FLAGWR pin. The SA and SD ISA bus pins are valid at least 30 nanoseconds before and 20 nanoseconds after FLAGWR is asserted such that (1) if a given data bit is not changing, then there will be no glitches on the output of the latch for that bit and (2) if a given data bit is changing, then there will be only one edge on the output of the latch for that signal.<br>Reads provide the last data written to this register (internally latched).<br>*Do not read from PM00 +18h and PM00 +1Ah in a single 4-byte cycle. The two registers must be read separately with 2-byte cycles.*<br>*To use the FLAGWR pin, PM00 +CAh must be set up for the FLAGWR function.* | RW |

**Power Management: Flag Read** **I/O Mapped Offset 1Ah**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15-0 | FRDDATA | **00000000** | **Flag Read Data.**<br>Reads to this register are passed to the ISA data bus to be driven by external '244-like devices with the FLAGRD# pin. It follows the same logical timing as the IOR# signal.<br>*Do not read from PM00 +18h and PM00 +1Ah in a single 4-byte cycle. The two registers must be read separately with 2-byte cycles.*<br>*To use the FLAGRD# pin, PM00 +CBh must be set up for the FLAGRD# function.* | RO |

**Power Management: Soft Logic Test** **I/O Mapped Offset 1Ch**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-1 | Processor Level 2 | 0 | **Test Bits [7:1].** To be determined. | RW |
| 0 | | 0 | **Test Bit 0.**<br>Speed up slow counter. When set, the slow counter that is used to generate the clocks for several functions are replaced with the clock derived from RTCX_IN. These include the four clocks to the PM00 +DCh blink clock generator, the clocks to all the debounce circuits, the four clocks to the system inactivity timer, and the clock to the power-button override counter.<br>*These bits are reset by RST_SOFT and their value is retained while in the SOFF state.* | RW |

**Power Management: ACPI GP Status Register** **I/O Mapped Offset 20h**

| | | | | |
|---|---|---|---|---|
| 15 | USB_RSM _STS | 0 | **USB Resume Event Status.**<br>0 = no activity<br>1 = indicates a USB-defined resume event has occurred. This may occur while the system is in the SOFF power state. This bit is reset by RST_SOFT and its value is retained in the SOFF state. | 4 |
| 14 | RI_STS | 0 | **Ring Indicator Pin Status.**<br>The bit is set when the RI# pin is asserted (active state is dependent upon the GPIO14 input polarity). | RWC |
| 13-11 | | 0 | **Reserved.** Always reads 0. | |
| 10 | TH_STS | 0 | **Thermal Pin Status.**<br>The bit is set when the THERM# pin is asserted (active state is dependent upon the GPIO2 input polarity). The latch that drives this bit is the same as the GPIO2 input-path latch. | RWC |

**Power Management: ACPI GP Status Register** **I/O Mapped Offset 20h**

| | | | | |
|---|---|---|---|---|
| 9 | EXTSMI_STS | 0 | **External SMI Pin Status.**<br>The bit is set when the EXTSMI# pin is asserted (active state is dependent upon the GPIO12 input polarity).<br>Because the circuit that drives this bit is the same as the GPIO12 input-path latch it is possible to use polarity control or the debounce function.<br>This bit is reset by RST_SOFT and the value of this bit is retained while in the SOFF state. | RWC |
| 8 | PME_STS | 0 | **PME Pin Status.**<br>The bit is set when the PME# pin is asserted low.<br>This bit is reset by RST_SOFT and the value of this bit is retained while in the SOFF state. | RWC |
| 7-6 | | 0 | **Reserved.** Always reads 0. | |
| 5 | SIT_STS | 0 | **System Inactivity Timer (SIT) Timeout Status.**<br>The bit is set by the hardware when the system inactivity timer times out. | RWC |
| 4-0 | | 0 | **Reserved.** Always reads 0. | |

**Power Management: ACPI Interrupt Enable** **I/O Mapped Offset 23h–22h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15 | | 0 | **Reserved.** Always reads 0. | |
| 14 | RIE | 0 | **Ring Indicator Enable.** This bit enables the RI# pin to signal ACPI interrupts.<br>0 = Disabled 1 = Enabled | RW |
| 13-11 | | 0 | **Reserved.** Always reads 0. | |
| 10 | THE | 0 | **Therm Enable.** This bit enables the THERM# pin to signal ACPI interrupts.<br>0 = Disabled 1 = Enabled | RW |
| 9 | SMIE | 0 | **External SMI Enable.** This bit enables the SMI# pin to signal ACPI interrupts.<br>0 = Disabled 1 = Enabled | RW |
| 8 | PMEI | 0 | **PME Enable.** This bit enables the PME# pin to signal ACPI interrupts.<br>0 = Disabled 1 = Enabled | RW |
| 7-6 | | 0 | **Reserved.** Always reads 0. | |
| 5 | SITE | 0 | **System Inactivity Timer Enable.**<br>0 = Disabled 1 = Enabled | RW |
| 4-0 | | 0 | **Reserved.** Always reads 0. | |

**Power Management: LPT-USB Event Status** **I/O Mapped Offset 24h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7 | LPT3_STS | 0 | **LPT3 Status.**<br>0 = no activity<br>1 = indicates access to I/O space in the range 3BCh-3BFh. | RWC |
| 6 | LPT2_STS | 0 | **LPT2 Status.**<br>0 = no activity<br>1 = indicates access to I/O space in the range 278h-27Fh. | RWC |

**Power Management: LPT-USB Event Status** **I/O Mapped Offset 24h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 5 | LPT1_STS | 0 | **LPT1 Status.**<br>0 = no activity<br>1 = indicates access to I/O space in the range 378h-37Fh. | RWC |
| 4 | USB_RSM | 0 | **USB Resume Event Status.**<br>0 = no activity<br>1 = indicates a USB-defined resume event has occurred. This may occur while the system is in the SOFF power state. This bit is reset by RST_SOFT and its value is retained in the SOFF state. | RWC |
| 3 | USB_BLK | 0 | **USB Bulk Transfer Status.**<br>0 = no activity<br>1 = indicates a USB bulk transfer has occurred | RWC |
| 2 | USB_INT | 0 | **USB Interrupt Transfer Status.**<br>0 = no activity<br>1 = indicates a USB interrupt transfer has occurred | RWC |
| 1 | USB_ISO | 0 | **USB Isochronous Transfer Status.**<br>0 = no activity<br>1 = indicates a USB isochronous transfer has occurred | RWC |
| 0 | USB_CTL | 0 | **USB Control Transfer Status.**<br>0 = no activity<br>1 = indicates a USB isochronous transfer has occurred | RWC |

**Power Management: LPT-USB Event Interrupt Enable** **I/O Mapped Offset 25h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7 | LPT3_EN | 0 | **LPT3 Interrupt Enable.** 0 = Disabled 1 = Enabled | RW |
| 6 | LPT2_EN | 0 | **LPT2 Interrupt Enable.** 0 = Disabled 1 = Enabled | RW |
| 5 | LPT1_EN | 0 | **LPT1 Interrupt Enable.** 0 = Disabled 1 = Enabled. | RW |
| 4 | USB_RSM | 0 | **USB Resume Event Interrupt Enable.** 0 = Disabled 1 = Enabled. | RW |
| 3 | USB_BLK | 0 | **USB Bulk Transfer Interrupt Enable.** 0 = Disabled 1 = Enabled. | RW |
| 2 | USB_INT | 0 | **USB Interrupt Transfer Interrupt Enable.** 0 = Disabled 1 = Enabled. | RW |
| 1 | USB_ISO | 0 | **USB Isochronous Transfer Interrupt Enable.** 0 = Disabled 1 = Enabled. | RW |
| 0 | USB_CTL | 0 | **USB Control Transfer Interrupt Enable.** 0 = Disabled 1 = Enabled. | RW |

**Power Management: Power Supply Control** **I/O Mapped Offset 27h–26h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15 | USB_RSM | 0 | **USB Resume Event Interrupt Enable.** 0 = Disabled 1 = Enabled**.** | RW |
| 14 | RI_CTL | 0 | **Ring Indicator Control.**<br>This bit controls the RI# input pin used to trigger power plane control.<br>0 = (Disabled) PM00 +28h bit[RI_STS] does not affect the state of the PWRON# pin.<br>1 = Enables the PM00 +28h bit[RI_STS]-set-to-high event to be used as a trigger to set the PWRON# output pin into the active state. | RW |
| 13 | SBOD | 0 | **Sleep Button Override Disable.**<br>0 = (Enabled) The power button override event from the SLPBTN# pin is enabled to place the system into the SOFF mode.<br>1 = (Disabled) The power button override event from the SLPBTN# pin (holding SLPBTN# active for four seconds) will not automatically transition the system into SOFF. | RW |
| 12 | SLPBTN_CTL | 0 | **Sleep Button Control.**<br>This bit enables the SLPBTN# input pin to trigger power plane control.<br>0 = (Disabled) The PM00 bit [SLPBTN_STS] does not affect the state of the PWRON# pin.<br>1 = (Enabled) The PM00 bit [SLPBTN_STS] set-to-high event can be used as a trigger to set the PWRON# output pin into the active state. | RW |
| 11 | PBOD | 0 | **Power Button Override Disable.**<br>0 = (Enabled) The power button override event is enabled to place the system into the SOFF mode.<br>1 = (Disabled) The power button override event (holding PWRBTN# active for four seconds) will not automatically transition the system into SOFF. | RW |
| 10 | PME_CTL | 0 | **PME Control.**<br>This bit enables the PME# input pin (PCI power management event) to trigger power plane control.<br>0 = (Disabled) PM00 +28h bit [PME_STS] does not affect the state of the PWRON# pin.<br>1 = (Enabled) The PM00 +28h bit [PME_STS]-set-to-high event can be used as a trigger to set the PWRON# output pin into the active state. | RW |
| 9 | PB_CTL | 0 | **Power Button Control (RW).**<br>This bit enables the Power button used to trigger power plane control.<br>0 = (Disabled) The PM00 bit [PWRBTN_STS] does not affect the state of the PWRON# pin.<br>1 = (Enabled) The PM00 bit [PWRBTN_STS] set-to-high event can be used as a trigger to set the PWRON# output pin into the active state. | RW |
| 8 | RTC_PS_CTL | 0 | **Real Time Clock Alarm Power Control.**<br>This bit enables the real time clock alarm used to trigger power plane control.<br>0 = (Disabled) The PM00 bit [RTC_STS] does not affect the state of the PWRON# pin.<br>1 = (Enabled)The PM00 bit [RTC_STS] set-to-high event can be used as a trigger to set the PWRON# output pin into the active state. | RW |
| 7-4 | | 0000 | **Reserved .** Always reads 0. | |
| 3 | HSLV | 0 | **Host Slave SMBus Address Match Control.**<br>This bit enables the SMBus host-as-slave address match used to trigger power plane control.<br>0 = Disabled<br>1 = (Enabled) The SMBus host-as-slave address match event (PM00 +E0h bit [HSLV_STS]) can be used as the trigger to set the PWRON# output pin into the active state. | RW |

**Power Management: Power Supply Control** **I/O Mapped Offset 27h–26h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 2 | SNP | 0 | **Snoop Address Match Control.**<br>This bit enables the SMBus snoop address match used to trigger power plane control.<br>0 = Disabled<br>1 = (Enabled) The SMBus snoop address match event (PM00 +E0h bit[SNP_STS]) can be used as the trigger to set the PWRON# output pin into the active state. | RW |
| 1 | USB_RSM | 0 | **USB Resume Control.**<br>This bit enables the USB-defined resume event used to trigger power plane control.<br>0 = Disabled<br>1 = (Enabled) The USB-defined resume event (PM00 +24h bit[USB_RSM_STS]) can be used as the trigger to set the PWRON# output pin into the active state. | RW |
| 0 | EXTSMI_CTL | 0 | **External SMI Control.**<br>This bit enables the EXTSMI# pin used to trigger power plane control.<br>0 = (Disabled) PM00 +28h bit [EXTSMI_STS] does not affect the state of the PWRON# pin.<br>1 = (Enabled) The PM00 +28h bit [EXTSMI_STS] set-to-high event can be used as a trigger to set the PWRON# output pin into the active state. | RW |

The bits in this register are used to enable a corresponding trigger that will set PWRON# into the active state. Thus, when the specified trigger occurs, PWRON# will go low to power up most of the system, including the VDD3 power plane of the AMD-756. If PWRON# is already low, then these bits will not affect it. This register is reset by RST_SOFT and when a power button override event occurs.

***Note:*** *The value of this register is retained while in the SOFF state.*

**Power Management: Global SMI Status** **I/O Mapped Offset 29h–28h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15 | | 0 | **Reserved.** Always reads 0. | |
| 14 | RI_STS | 0 | **Ring Indicator Pin Status.**<br>The bit is set when the RI# pin is asserted (active state is dependent upon the GPIO14 input polarity). | RWC |
| 13-12 | | 0 | **Reserved.** Always reads 0. | |
| 11 | SMB_STS | 0 | **System Management Bus Event.**<br>The bit is set when a SMBus event occurs, including the completion of the current SMBus host access, host-as-slave access, slave detect access, and assertion of SMBALERT#. The status bits for this register are all the STS suffix bits found in PM00 +E0h and enabled in PM00 +E2h. | RWC |
| 10 | TH_STS | 0 | **Thermal Pin Status.**<br>The bit is set when the THERM# pin is asserted (active state is dependent upon the GPIO2 input polarity). The latch that drives this bit is the same as the GPIO2 input-path latch. | RWC |

**Power Management: Global SMI Status (continued)** **I/O Mapped Offset 29h–28h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 9 | EXTSMI_STS | 0 | **External SMI Pin Status.**<br>The bit is set when the EXTSMI# pin is asserted (active state is dependent upon the GPIO12 input polarity).<br>Because the circuit that drives this bit is the same as the GPIO12 input-path latch it is possible to use polarity control or the debounce function.<br>This bit is reset by RST_SOFT and the value of this bit is retained while in the SOFF state. | RWC |
| 8 | PME_STS | 0 | **PME Pin Status.**<br>The bit is set when the PME# pin is asserted low.<br>This bit is reset by RST_SOFT and the value of this bit is retained while in the SOFF state. | RWC |
| 7 | SMI_STS | 0 | **Software SMI Status).**<br>The bit is set when a write of any value is sent to PM00 +2Fh. This bit can result in SMI interrupts only (if enabled in PM00 +2Ah bit[SWISMI_EN]); it cannot be enabled to generate SCI interrupts. | RWC |
| 6 | BIOS_STS | 0 | **BIOS Status.**<br>The bit is set when the PM00 +04h bit[GBL_RLS] is set (by ACPI software to indicate the release of the SCI/SMI lock). BIOS_STS is cleared when a 1 is written to it; writing a 1 to BIOS_STS also causes the hardware to clear PM00 +04h bit[GBL_RLS]. This bit can result in SMI interrupts only (if enabled in PM00 +2Ah bit[BIOSSMI_EN]); it cannot be enabled to generate SCI interrupts. | RWC |
| 5 | SIT_STS | 0 | **System Inactivity Timer (SIT) Timeout Status.**<br>The bit is set by the hardware when the system inactivity timer times out. | RWC |
| 4 | USB_EVT | 0 | **USB Event Status.**<br>LPT access or USB transfer or resume event status. This bit is set when any of the bits in PM00 +24h that are enabled in PM00 +25h, go high to indicate a USB transfer or resume event or an LPT access. | RO |
| 3 | GPIO_EVT | 0 | **GPIO Interrupt Status.**<br>The bit is set when any of the enabled GPIO pin status bits specified by PM00 +D4h become active. | O |
| 2 | PM1_EVT | 0 | **PM1 Event Status.**<br>The bit is set when any of the enabled power management events specified by PM00 that can cause interrupts to become active. These include PM00 bits [RTC_STS, SLPBTN_STS, PWRBTN_STS, GBL_STS, and TMR_STS]. | RO |
| 1 | | 0 | **Reserved.** Always reads 0. | |
| 0 | TRP_SEVT | 0 | **Trap SMI Status.**<br>The bit is set when any of the enabled hardware trap status bits specified by PM00 +A8h become active. | RO |

Each of the EVT bits (bits[4:2, 0]) specify enabled status bits in other registers. These are not sticky bits. They reflect the combinatorial equation of: XXX_EVT = (status1 AND enable1) OR (status2 AND enable2) . . . OR (statusx AND enablex).

**Power Management: Global SMI Enable** **I/O Mapped Offset 2Bh–2Ah**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15 | | 0 | **Reserved.** Always reads 0. | |
| 14 | RI_SMIE | 0 | **Ring Indicator SMI Enable.**<br>0 = Disabled 1 = Enables the PM00 +28h bit [RI_STS] to generate an SMI. | RW |
| 13 | SB_SMIE | 0 | **Sleep Button SMI Enable.**<br>0 = Disabled 1 = Enables the PM00 +28h bit [SLPBTN_STS] to generate an SMI. | RW |
| 12 | PB_SMIE | 0 | **Power Button SMI Enable).**<br>0 = Disabled 1 = Enables the PM00 +28h bit [PWRBTN_STS] to generate an SMI. | RW |
| 11 | SMB_SMIE | 0 | **SM Bus SMI Enable.**<br>0 = Disabled 1 = Enables the PM00 +28h bit [SMBUS_EVT] to generate an SMI. | RW |
| 10 | TH_SMIE | 0 | **Thermal SMI Enable.**<br>0 = Disabled 1 = Enables the PM00 +28h bit [THERM_STS] to generate an SMI. | RW |
| 9 | EXT_SMIE | 0 | **External SMI SMI Enable.**<br>0 = Disabled 1 = Enables the PM00 +28h bit [EXTSMI_STS] to generate an SMI. | RW |
| 8 | PME_SMIE | 0 | **PME SMI Enable.**<br>0 = Disabled 1 = Enables the PM00 +28h bit [PME_STS] to generate an SMI. | RW |
| 7 | S_SMIE | 0 | **Software SMI Enable.**<br>0 = Disabled 1 = Enables the PM00 +28h bit [SWI_STS] to generate an SMI. | RW |
| 6 | BIOS_SMIE | 0 | **BIOS SMI Enable.**<br>0 = Disabled 1 = Enables the PM00 +28h bit [BIOS_STS] to generate an SMI. | RW |
| 5 | SIT_SMIE | 0 | **SIT SMI Enable.**<br>0 = Disabled 1 = Enables the PM00 +28h bit [SIT_STS] to generate an SMI. | RW |
| 4 | USB_SMIE | 0 | **USB Resume SMI Enable.**<br>0 = Disabled 1 = Enables any of the PM00 +28h bits [4:0] bits to generate an SMI if enabled by the corresponding bits in PM00 +25h bits [4:0]. | RW |
| 3 | GPIO_SMIE | 0 | **GPIO SMI Enable.**<br>0 = Disabled 1 = Enables the PM00 +28h bit [GPIO_EVT] to generate an SMI. | RW |
| 2 | PM1_SMIE | 0 | **PM1 SMI Enable.**<br>0 = Disabled 1 = Enables the PM00 +28h bit [PM1_EVT] to generate an SMI. | RW |
| 1 | | 0 | **Reserved.** Must remain 0 for proper operation. | RW |
| 0 | TRP_SMIE | 0 | **Trap SMI Enable.**<br>0 = Disabled 1 = Enables the PM00 +28h bit [TRP_EVT] to generate an SMI. | RW |

Each of these enable bits gates the ability for the status and event bits to generate SMI# interrupts. When set, these bits enable the corresponding event to generate an SMI interrupt, regardless of the state of PM00 +04 bit[SCI_EN].

**Power Management: Global SMI Control** **I/O Mapped Offset 2Dh–2Ch**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15-6 | | 0 | **Reserved.** Always reads 0. | |
| 5 | SMIACT | 0 | **SMI Active.**<br>If SMILK is high, then this bit is set on the leading edge of the SMI output. It holds the SMI# pin in the active state. | RWC |
| 4 | SMILK | 0 | **SMI Lock Control.**<br>0 = Latching disabled, the SMI# pin is controlled solely by the logic prior to SMIACT.<br>1 = Latching enabled, the SMI# pin is locked into the active state, by a latch before the output pad, after it is asserted. The latch is controlled by SMIACT. | RW |
| 3-2 | | 0 | **Reserved.** Always reads 0. | |
| 1 | BIOS_RLS | 0 | **BIOS SCI/SMI lock release.**<br>This bit is set by software to indicate the release of the SCI/SMI lock. When this bit is set, PM00 bit[GBL_STS] is set by the hardware. BIOS_RLS is cleared by the hardware when PM00 bit[GBL_STS] is cleared by software.<br>*If PM00 +02 bit[GBL_EN] is set, then setting this bit will generate an SCI or SMI interrupt*. | RW |
| 0 | SMI_EN | 0 | **BIOS SCI/SMI lock release.**<br>This bit is set by software to indicate the release of the SCI/SMI lock. When this bit is set, PM00 bit[GBL_STS] is set by the hardware. BIOS_RLS is cleared by the hardware when PM00 bit[GBL_STS] is cleared by software.<br>*If PM00 +02 bit[GBL_EN] is set, then setting this bit will generate an SCI or SMI interrupt*. | RW |

**Power Management: Software SMI Trigger** **I/O Mapped Offset 2Fh**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | SMI Command | 00000000 | **SMI Command.**<br>Reads or writes to this register set PM00+28 bit[SWI_STS]. If PM00+2A bit[SWI_EN] is set, then accesses to this port can be used to generate SMI interrupts. Reads of this register provide the data last written to it. | RW |

Offsets 40-94 are reserved and should not be written to.

**Power Management: System Inactivity Timer** **I/O Mapped Offset 9Bh–98h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-18 | | 0 | **Reserved.** Always reads 0. | |
| 17-16 | | 00 | **Clock Source.**<br>Specifies the clock to the system inactivity timer as in the following table.<br>Bits: 00 — Clock Period: 64 msec — Maximum Time (Clock x 255): 16.32 seconds<br>Bits: 01 — Clock Period: 1 sec — Maximum Time (Clock x 255): 255 seconds = 4.25 minutes<br>Bits: 10 — Clock Period: 16 sec — Maximum Time (Clock x 255): 68 minutes = 1.13 hours<br>Bits: 11 — Clock Period: 256 sec — Maximum Time (Clock x 255): 1088 minutes = 18.13 hours | RW |
| 15-8 | | 00000000 | **Current Count.**<br>System inactivity timer current count value is read here. | RO |
| 7-0 | | 00000000 | **Reload Value.**<br>Writes to this field cause the system inactivity counter to be reloaded with the value placed here. | WO |

**Power Management: Hardware Trap Status Bits** **I/O Mapped Offset ABh–A8h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-20 | | 0 | **Reserved.** Always read 0. | |
| 19 | PMM2_STS | 0 | **Programmable Memory Range Monitor 2 Trap Status.**<br>0 = No Event 1 = Event occurred | RWC |
| 18 | PMM1_STS | 0 | **Programmable Memory Range Monitor 1 Trap Status.**<br>0 = No Event 1 = Event occurred | RWC |
| 17 | PRM4_STS | 0 | **Programmable Range Monitor 4 Trap Status.** 0 = No Event 1 = Event occurred | RWC |
| 16 | PRM3_STS | 0 | **Programmable Range Monitor 3 Trap Status.** 0 = No Event 1 = Event occurred | RWC |
| 15 | PRM2_STS | 0 | **Programmable Range Monitor 2 Trap Status.** 0 = No Event 1 = Event occurred | RWC |
| 14 | PRM1_STS | 0 | **Programmable Range Monitor 1 Trap Status.** 0 = No Event 1 = Event occurred | RWC |
| 13 | USB_STS | 0 | **USB Trap Status.** 0 = No Event 1 = Event occurred | RWC |
| 12 | PCMCIA2_STS | 0 | **PCMCIA2 Trap Status.** 0 = No Event 1 = Event occurred | RWC |
| 11 | PCMCIA1_STS | 0 | **PCMCIA1 Trap Status.** 0 = No Event 1 = Event occurred | RWC |
| 10 | KBM_STS | 0 | **Keyboard Mouse Trap Status.** 0 = No Event 1 = Event occurred | RWC |
| 9 | VID_STS | 0 | **Video Trap Status.** 0 = No Event 1 = Event occurred | RWC |
| 8 | AUD_STS | 0 | **Audio Trap Status.** 0 = No Event 1 = Event occurred | RWC |
| 7 | COMB_STS | 0 | **COM B Trap Status.** 0 = No Event 1 = Event occurred | RWC |
| 6 | COMA_STS | 0 | **COM A Trap Status.** 0 = No Event 1 = Event occurred | RWC |

**Power Management: Hardware Trap Status Bits (continued)** **I/O Mapped Offset ABh–A8h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 5 | LPT_STS | 0 | **LPT Trap Status.** 0 = No Event 1 = Event occurred | RWC |
| 4 | FDD_STS | 0 | **Floppy Disk Drive Trap Status.** 0 = No Event 1 = Event occurred | RWC |
| 3 | DSS_STS | 0 | **Disk Secondary Slave Trap Status.** 0 = No Event 1 = Event occurred | RWC |
| 2 | DSM_STS | 0 | **Disk Secondary Master Trap Status.** 0 = No Event 1 = Event occurred | RWC |
| 1 | DPS_STS | 0 | **Disk Primary Slave Trap Status.** 0 = No Event 1 = Event occurred | RWC |
| 0 | DPM_STS | 0 | **Disk Primary Master Trap Status.** 0 = No Event 1 = Event occurred | RWC |

Each of these status bits is driven by a hardware trap detect timer. If the trap occurs, then the status bit is set. Each bit is cleared when written with a 1; writing a 0 has no effect. If a status bit is high and its corresponding enable bit is high, it will cause the hardware trap global event bit to go high in PM00 +28h bit[TRP_EVT].

**Power Management: Hardware Trap Enable Bits** **I/O Mapped Offset AFh–ACh**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-20 | | 0 | **Reserved.** Always read 0. | |
| 19 | PMM2_EN | 0 | **Programmable Memory Range Monitor 2 Trap Enable.** 0 = Disabled 1 = Enabled | RW |
| 18 | PMM1_EN | 0 | **Programmable Memory Range Monitor 1 Trap Enable.** 0 = Disabled 1 = Enabled | RW |
| 17 | PRM4_EN | 0 | **Programmable Range Monitor 4 Trap Enable.** 0 = Disabled 1 = Enabled | RW |
| 16 | PRM3_EN | 0 | **Programmable Range Monitor 3 Trap Enable.** 0 = Disabled 1 = Enabled | RW |
| 15 | PRM2_EN | 0 | **Programmable Range Monitor 2 Trap Enable.** 0 = Disabled 1 = Enabled | RW |
| 14 | PRM1_EN | 0 | **Programmable Range Monitor 1 Trap Enable.** 0 = Disabled 1 = Enabled | RW |
| 13 | USB_EN | 0 | **USB Trap Enable.** 0 = Disabled 1 = Enabled | RW |
| 12 | PCMCIA2_EN | 0 | **PCMCIA2 Trap Enable.** 0 = Disabled 1 = Enabled | RW |
| 11 | PCMCIA1_EN | 0 | **PCMCIA1 Trap Enable.** 0 = Disabled 1 = Enabled | RW |
| 10 | KBM_EN | 0 | **Keyboard Mouse Trap Enable.** 0 = Disabled 1 = Enabled | RW |
| 9 | VID_EN | 0 | **Video Trap Enable.** 0 = Disabled 1 = Enabled | RW |
| 8 | AUD_EN | 0 | **Audio Trap Enable.** 0 = Disabled 1 = Enabled | RW |
| 7 | COMB_EN | 0 | **COM B Trap Enable.** 0 = Disabled 1 = Enabled | RW |
| 6 | COMA_EN | 0 | **COM A Trap Enable.** 0 = Disabled 1 = Enabled | RW |
| 5 | LPT_EN | 0 | **LPT Trap Enable.** 0 = Disabled 1 = Enabled | RW |
| 4 | FDD_EN | 0 | **Floppy Disk Drive Trap Enable.** 0 = Disabled 1 = Enabled | RW |
| 3 | DSS_EN | 0 | **Disk Secondary Slave Trap Enable.** 0 = Disabled 1 = Enabled | RW |
| 2 | DSM_EN | 0 | **Disk Secondary Master Trap Enable.** 0 = Disabled 1 = Enabled | RW |

**Power Management: Hardware Trap Enable Bits (continued)** **I/O Mapped Offset AFh–ACh**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 1 | DPS_EN | 0 | **Disk Primary Slave Trap Enable.** 0 = Disabled 1 = Enabled | RW |
| 0 | DPM_EN | 0 | **Disk Primary Master Trap Enable.** 0 = Disabled 1 = Enabled | RW |

Each of these bits gates the corresponding bit in the hardware trap status register from generating an interrupt. A high enables the corresponding status bit to generate an interrupt; also, when an enabled hardware event status bit is set, the PM00 +28h bit[TRP_EVT] bit goes high. A low disables the corresponding status bit.

**Power Management: Hardware Trap Reload Enable for System Inactivity Timer** **I/O Mapped Offset B3h–B0h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-22 | | 0 | **Reserved.** Always read 0. | |
| 21 | BM_REQ | 0 | **Bus Master Request Inactivity Timer Reload Enable.** 0 = Disable reload 1 = Enable reload | RW |
| 20 | EXT_SMI | 0 | **External SMI Inactivity Timer Reload Enable** 0 = Disable reload 1 = Enable reload | RW |
| 19 | PMM2_RLE | 0 | **Programmable Memory Range Monitor 2.** Inactivity Timer Reload Enable 0 = Disable reload 1 = Enable reload | RW |
| 18 | PMM1_RLE | 0 | **Programmable Memory Range Monitor 1.** Inactivity Timer Reload Enable 0 = Disable reload 1 = Enable reload | RW |
| 17 | PRM4_RLE | 0 | **Programmable Range Monitor 4 Inactivity Timer Reload Enable.** 0 = Disable reload 1 = Enable reload | RW |
| 16 | PRM3_RLE | 0 | **Programmable Range Monitor 3 Inactivity Timer Reload Enable.** 0 = Disable reload 1 = Enable reload | RW |
| 15 | PRM2_RLE | 0 | **Programmable Range Monitor 2 Inactivity Timer Reload Enable.** 0 = Disable reload 1 = Enable reload | RW |
| 14 | PRM1_RLE | 0 | **Programmable Range Monitor 1 Inactivity Timer Reload Enable.** 0 = Disable reload 1 = Enable reload | RW |
| 13 | USB_RLE | 0 | **USB Inactivity Timer Reload Enable.** 0 = Disable reload 1 = Enable reload | RW |
| 12 | PCMCIA2_RLE | 0 | **PCMCIA2 Inactivity Timer Reload Enable.** 0 = Disable reload 1 = Enable reload | RW |
| 11 | PCMCIA1_RLE | 0 | **PCMCIA1 Inactivity Timer Reload Enable.** 0 = Disable reload 1 = Enable reload | RW |
| 10 | KBM_RLE | 0 | **Keyboard Mouse Inactivity Timer Reload Enable.** 0 = Disable reload 1 = Enable reload | RW |
| 9 | VID_RLE | 0 | **Video Inactivity Timer Reload Enable.** 0 = Disable reload 1 = Enable reload | RW |
| 8 | AUD_RLE | 0 | **Audio Inactivity Timer Reload Enable.** 0 = Disable reload 1 = Enable reload | RW |
| 7 | COMB_RLE | 0 | **COM B Inactivity Timer Reload Enable.** 0 = Disable reload 1 = Enable reload | RW |

**Power Management: Hardware Trap Reload Enable for System Inactivity Timer (continued)** **I/O Mapped Offset B3h–B0h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 6 | COMA_RLE | 0 | **COM A Inactivity Timer Reload Enable.** 0 = Disable reload 1 = Enable reload | RW |
| 5 | LPT_RLE | 0 | **LPT Inactivity Timer Reload Enable.** 0 = Disable reload 1 = Enable reload | RW |
| 4 | FDD_RLE | 0 | **Floppy Disk Drive Inactivity Timer Reload Enable.** 0 = Disable reload 1 = Enable reload | RW |
| 3 | DSS_RLE | 0 | **Disk Secondary Slave Inactivity Timer Reload Enable.** 0 = Disable reload 1 = Enable reload | RW |
| 2 | DSM_RLE | 0 | **Disk Secondary Master Inactivity Timer Reload Enable.** 0 = Disable reload 1 = Enable reload | RW |
| 1 | DPS_RLE | 0 | **Disk Primary Slave Inactivity Timer Reload Enable.** 0 = Disable reload 1 = Enable reload | RW |
| 0 | DPM_RLE | 0 | **Disk Primary Master Inactivity Timer Reload Enable.** 0 = Disable reload 1 = Enable reload | RW |

Each of these bits enables reloading of its corresponding inactivity timer.

**Power Management: IRQ Reload Enable for Inactivity Timer** **I/O Mapped Offset B7h–B4h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-16 | | 0 | **Reserved.** Must remain 0 for proper operation. | RW |
| 15-0 | | 0 | **IRQs Reload the system inactivity timer.**<br>Each of these bits corresponds to an IRQ number (e.g., bit[12] corresponds to IRQ12). The exception to this is bit[2], which corresponds to the logical OR of the two INTR[1:0] pins.<br>When set, each bit enables the corresponding interrupt signal to cause the system inactivity timer to reload while it is high. For example, if IRQRL[9] is high, then while IRQ9 is high, the system inactivity timer will be held at its reload value.<br>When cleared, bits of this register do not affect the system inactivity time | RW |

# General Purpose I/O Registers

**Power Management: GPIO Direction Control (GPIO_DIR)** **I/O Mapped Offset C0h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7 | | 0 | **Reserved.** Always reads 0. | |
| 6 | LTCH_STS | 0 | **Latch Status.** GPIO latch status.<br>This bit provides the current state of the latch associated with the input pad for the pin that corresponds to the register. | RWC |
| 5 | RTIN | 0 | **Real Time IN.**<br>This bit provides the current, not-inverted state of the pad for the pin that corresponds to the register. | RO |

**Power Management: GPIO Direction Control (GPIO_DIR) (continued)** **I/O Mapped Offset C0h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 4 | Debounce | 0 | **Debounce.** Debounce the input signal.<br>0 = No debounce<br>1 = the input signal is required to be held active without glitches for 12 to 16 milliseconds before being allowed to set the GPIO latch or being capable of being passed along to the circuitry being controlled by the output of the input path. | RW |
| 3-2 | Mode [1:0] | 00 | **Mode[1:0].** Pin mode select.<br>MODE[1] selects between the pin being defined as a GPIO function (MODE[1] low) or an alternate function (MODE[1] high). For GPIO pins, MODE[0] selects between inputs (MODE[0] low) and outputs (MODE[0] high). If MODE[1] is high, then MODE[0] selects between multiple alternate functions for the pins. Table 79 below shows the default states for these registers and the pin definitions base on the state of MODE[1:0]. The "Default" field shows the defaults for all the bits in the register. | RW |
| 1-0 | X[1:0] | 00 | **X[1:0].**<br>If the GPIO input path is not used by the pin (e.g. if the function is a GPIO function or if it is an configurable input pin like the PNP IRQs), then this field does not matter. If the GPIO input path is used by the pin, then based on whether this pin is a input or an output (selected by MODE also), this register defines I/O modes as described in Table 80 below. | RW |

Registers D1h-C0h all have this same format. There is one single-byte register for each GPIO pin. Registers PM00 +C3h (associated with SLPBTN#) and PM00 +CCh (associated with EXTSMI#) are reset by RST_SOFT and remain powered when in the SOFF state (powered by VDD_SOFT). The other registers are reset by PWRGD and off when in the SOFF state.

**Table 79. Pin Function**

| Pin | Control Register | Default | MODE | Function | Notes |
|---|---|---|---|---|---|
| GPIO[17:0] | | | b00 | General Purpose Input | |
| GPIO[17:0] | | | b01 | General Purpose Output | |
| GPIO[0] | PM00 +C0h | h08 (SMBUSC) | b1x | SMBUSC | 3 |
| GPIO[1] | PM00 +C1h | h08 (SMBUSD) | b1x | SMBUSD | 3 |
| GPIO[2] | PM00 +C2h | h0C (THERM#) | b10 | PMIRQ0; GPIO input path drives logic | |
| | | | b11 | THERM#; GPIO input path drives throttling logic | 1 |
| GPIO[3] | PM00 +C3h | h0C (SLPBTN#) | b10 | PMIRQ1; GPIO input path drives logic | 3 |
| | | | b11 | SLPBTN#; GPIO input path drives PM logic | 1,2 |
| GPIO[4] | PM00 +C4h | h08 (SUSPEND#) | b1x | SUSPEND# | |
| GPIO[5] | PM00 +C5h | h05 (GPIO output, high) | b10 | PNPCS0# | |
| | | | b11 | CPUSLEEP# | |

**Table 79. Pin Function (continued)**

| Pin | Control Register | Default | MODE | Function | Notes |
|---|---|---|---|---|---|
| GPIO[6] | PM00 +C6h | h05 (GPIO output, high) | b10 | PNPCS1# | |
| | | | b11 | CPUSTOP# | |
| GPIO[7] | PM00 +C7h | h05 (GPIO output, high) | b10 | PNPIRQ0; GPIO input path drives PnP logic | |
| | | | b11 | PCISTOP# | |
| GPIO[8] | PM00 +C8h | h04 (GPIO output, low) | b10 | PNPIRQ1; GPIO input path drives PnP logic | |
| | | | b11 | CACHE_ZZ | |
| GPIO[9] | PM00 +C9h | h05 (GPIO output, high) | b10 | PNPIRQ2; GPIO input path drives PnP logic | |
| | | | b11 | DCSTOP# | |
| GPIO[10] | PM00 +CAh | h04 (GPIO output, low) | b10 | PNPDRQ 11b FLAGWR | |
| GPIO[11] | PM00 +CBh | h05 (GPIO output, high) | b10 | PNPDAK# | |
| | | | b11 | FLAGRD# | |
| GPIO[12] | PM00 +CCh | h0C (EXTSMI#) | b10 | BMREQ# | 3 |
| | | | b11 | EXTSMI#; GPIO input path drives PM logic | |
| GPIO[13] | PM00 +CDh | h0C (KEYLOCK) | b10 | PRDY | |
| | | | b11 | KEYLOCK | |
| GPIO[14] | PM00 +CEh | h08 (RI#) | b1x | RI#; GPIO input path drives PM logic | 3 |
| GPIO[15] | PM00 +CFh | h04 (GPIO output, low) | ‘b1x | C32KHZ | |
| GPIO[16] | PM00 +D0h | h05 (GPIO output, high) | b10 | INTIRQ8# | |
| | | | b11 | SQWAVE | |
| GPIO[17] | PM00 +D1h | h00 (GPIO input) | b10 | MSIRQ | |
| | | | b11 | SERIRQ | |

*Note 1: the output of the input path for GPIO[17, 16, 3, 2] goes to the I/O APIC to drive the interrupt request inputs to some of the redirection-table entries. These signals, between the GPIO logic and the APIC, are not ever disabled, even if the pin’s function is other-than GPIO.*

*Note 2: When the SLPBTN# function is selected, then the X0 further decodes this input. If PM00 +C3h bits[3:0] = 0Ch, then the SLPBTN# function is selected; if PM00 +C3 bits[3:0] = 0Dh, then the EXTIRQ8#, external real time clock interrupt function is selected. EXTIRQ8# is passed to the PIC logic without going through the GPIO input path.*

*Note 3: PM00 +C0h, PM00 +C1h, PM00 +C3h, PM00 +CCh, and PM00 +CEh are all reset by RST_SOFT and their values are retained while in the SOFF state.*

**Table 80. I/O Mode**

| I/O (MODE) | Bit | Name | Function |
|---|---|---|---|
| Input | X0 | ACTIVEHI | When low, the pin is active low and the signal is inverted as it is brought into the input path; when high the pin is active high and therefore not inverted as it is brought through the input path. |
| Input | X1 | LATCH | When low the latched version of the signal is not selected. When high, the latch output is selected. |
| Output | X[1:0]=00b | | Output is forced low. |
| Output | X[1:0]=01b | | Output is forced high. |
| Output | X[1:0]=10b | | GPIO output clock 0 (specified by PM00 +DC bits[15:0]). |
| Output | X[1:0]=11b | | GPIO output clock 1 (specified by PM00 +DC bits[31:16]). |

**GPIO Pin Interrupt Status** **BAR0 + Offset D7h–D4h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-18 | | 0 | **Reserved.** Must remain 0 for proper operation. | RW |
| 17-0 | | 0 | **GPIO Status Bits.**<br>Each of these status bits is driven by the output of the input circuit associated with the GPIO pins. The latch associated with each GPIO input circuit is cleared when the corresponding bit in this register is written with a 1; writing a 0 has no effect. | RWC |

**GPIO Pin Interrupt Enable** **BAR0 + Offset DBh–D8h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-18 | | 0 | **Reserved.** Must remain 0 for proper operation. | RW |
| 17-0 | | 0 | **GPIO Enable Bits.**<br>Each of these enable bits gate the ability of the output of the GPIO pin input path to generate an interrupt. When these bits are low, the path from the corresponding status bit in PM00 +D4h to the SCI/SMI interrupts is forced to the inactive state. | RW |

**GPIO Output Clock 0 and 1** **BAR0 + Offset DFh–DCh**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-30 | CLK1_Base | 00 | **Clock 1 Base.** GPIO output clock timer base.<br>Specifies the clock for the counter that generates the GPIO output clock.<br><br>Bits: 00 — Clock Length: 250 microseconds (default)<br>Bits: 01 — Clock Length: 2 milliseconds<br>Bits: 10 — Clock Length: 16 milliseconds<br>Bits: 11 — Clock Length: 256 milliseconds | RW |

**GPIO Output Clock 0 and 1** **BAR0 + Offset DFh–DCh**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 29-23 | CLK1_HI | 0000000 | **CLK 1 High.** GPIO output clock 1 high time.<br>Specifies the high time for the GPIO output clocks in increments of the clock specified by CLK1 BASE (if the base is 16 milliseconds, then 0 specifies 16 milliseconds, 1 specifies 32 milliseconds, etc.).<br>0 = 1 clock 1 = 2 clocks | RW |
| 22-16 | CLK1_LO | 000000 | **Clock 1 Low.**<br>Specifies the low time for the GPIO output clocks in increments of the clock specified by CLK1 BASE (if the base is 16 milliseconds, then 0 specifies 16 milliseconds, 1 specifies 32 milliseconds, etc.).<br>0 = 1 clock 1 = 2 clocks | RW |
| 15-14 | CLK 0 Base | 00 | **Clock 0 Base.**<br>Specifies the clock for the counter that generates the GPIO output clock 0.<br>Bits: 00 = 250 microseconds (default); 01 = 2 milliseconds; 10 = 16 milliseconds; 11 = 256 milliseconds | RW |
| 13-7 | CLK0_HI | 0000000 | **CLK 0 High.**<br>Specifies the high time for the GPIO output clocks in increments of the clock specified by CLK0 BASE (if the base is 16 milliseconds, then 0 specifies 16 milliseconds, 1 specifies 32 milliseconds, etc.).<br>0 = 1 clock 1 = 2 clocks | RW |
| 6-0 | CLK0_LO | 0000000 | **Clock 0 Low.**<br>Specifies the low time for the GPIO output clock 0 in increments of the clock specified by CLK0 BASE (if the base is 16 milliseconds, then 0 specifies 16 milliseconds, 1 specifies 32 milliseconds, etc.).<br>0 = 1 clock 1 = 2 clocks | RW |

This register specifies the high time and the low time for the GPIO output clocks. These clocks can be selected as the output for any of the GPIO pins. These output clocks consist of a 7-bit down counter that is alternately loaded with the high time and the low time. The clock for the counters is selected by CLK[1,0]BASE.

# System Management Bus Registers

**SM Bus Global Status** **PM00 + Offset E1h–E0h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15-12 | | 0000 | **Reserved.** These bits must remain 0 for proper operation. | RW |
| 11 | SM_BSY | 0 | **SMBus Busy.**<br>0 = The SMBus is not busy.<br>1 = The SMBus is currently busy with a cycle generated by either the host or another SMBus master. | RO |
| 10 | SNBA_STS | 0 | **SM Bus Alert Status.**<br>This bit is set by the hardware when SMBALERT# is asserted low. This bit will not be set unless the SMBALERT# function is selected by function 3, offset 46, bits[10:9]. This bit generates an SMI or SCI interrupt if enabled to do so by PM00 +E2h bit[SMBA_EN]. | RWC |
| 9 | HSLV_STS | 0 | **Host-as-Slave Address Match Status.**<br>This bit is set by the hardware when an SMBus master (including the host controller) generates a SMBus write cycle with a 7-bit address that matches the one specified by PM00 +EEh. This bit is not set until the end of the acknowledge bit after the last byte is transferred over the SMBus cycle; however, if a time out occurs after the address match occurs and before last acknowledge, then this bit will not be set. This bit generates an SMI or SCI interrupt if enabled to do so by PM00 +E2h bit[HSLV_EN]. | RWC |
| 8 | SNP_STS | 0 | **Snoop Address Match Status.**<br>This bit is set by the hardware when an SMBus master (including the host controller) generates an SMBus cycle with a 7-bit address that matches the one specified by PM00 +EFh. This bit is not set until the end of the acknowledge bit after the last byte is transferred over the SMBus cycle; however, if a time out occurs after the address match occurs and before the last acknowledge, then this bit will not be set. This bit generates an SMI or SCI interrupt if enabled to do so by PM00 +E2 bit[SNP_EN]. | RWC |
| 7-6 | | 0 | **Reserved.** These bits must remain 0 for proper operation. | RW |
| 5 | TO_STS | 0 | **Time Out Error Status.**<br>This bit is set by the hardware when a slave device forces a time out by holding the SMBUSC pin low for more than 30 milliseconds. This bit generates an SMI or SCI interrupt if enabled to do so by PM00 +E2h bit[HCYC_EN]. | RWC |
| 4 | HCYC_STS | 0 | **Host Cycle Complete Status.**<br>This bit is set by the hardware when a host cycle completes successfully. This bit generates an SMI or SCI interrupt if enabled to do so by PM00 +E2h bit[HCYC_EN]. | RWC |
| 3 | HST_STS | 0 | **Host Controller Busy.**<br>0 = The SMB host controller is not busy.<br>1 = The SMBus host controller is currently busy with a cycle. | RO |
| 2 | PRERR_STS | 0 | **Protocol Error Status.**<br>This bit is set by the hardware when a slave device does not generate an acknowledge at the appropriate time during a host SMBus cycle. This bit generates an SMI or SCI interrupt if enabled to do so by PM00 +E2h bit[HCYC_EN]. | RWC |
| 1 | COL_STS | 0 | **Collision Status.** Host collision status.<br>This bit is set by the hardware when a host transfer is initiated while the SMBus is busy. This bit is cleared when a 1 is written to it; writing a 0 to this bit has no effect. This bit will generate an SMI or SCI interrupt if enabled to do so by PM00 +E2h bit[HCYC_EN]. | RWC |

**SM Bus Global Status (continued)** **PM00 + Offset E1h–E0h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 0 | ABRT_STS | 0 | **Abort Status.** Host transfer abort status.<br>This bit is set by the hardware after a host transfer is aborted by PM00 +E2h bit[ABORT] command. This bit generates an SMI or SCI interrupt if enabled to do so by PM00 +E2h bit[HCYC_EN]. | RWC |

Bits[9:8] are powered by the VDD_SOFT plane and reset by RST_SOFT rather than PWRGD.

**SM Bus Global Enable Register** **PM00 + Offset E3h–E2h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15-11 | | 0 | **Reserved.** These bits must remain 0 for proper operation. | RW |
| 10 | SNBA_EN | 0 | **SM Bus Alert Enable.**<br>1 = specifies that an SMI or SCI interrupt will be generated when the SMBALERT# pin is asserted. When low, no interrupts are generated when this pin is set. This bit has no effect unless the SMBALERT# function is selected by function 3, offset 46, bits[10:9]. | RW |
| 9 | HSLV_EN | 0 | **Host-as-Slave Address Match Enable.**<br>Host-as-slave address match interrupt enable.<br>0 = (Disabled) no interrupts are generated when PM00 +E0h [HSLV_STS] is set.<br>1 = (Enabled) specifies that an SMI or SCI interrupt will be generated when that bit is set. | RW |
| 8 | SNP_EN | 0 | **Snoop Address Match Enable.**<br>0 = (Disabled) no interrupts are generated when this bit is set.<br>1 = (Enabled) specifies that an SMI or SCI interrupt will be generated when PM00 +E0h bit[SNP_STS] is set high. | RW |
| 7-6 | | 0 | **Reserved.** These bits must remain 0 for proper operation. | RW |
| 5 | ABORT | 0 | **Abort Current Host Command.**<br>When this bit is set by software, the SMBus logic generates a stop event on the SMBus pins as soon as possible (this may take a while if the SMBus slave has been instructed to generate zeros during a read cycle). After the stop event is generated, PM00 +E0h bit[ABRT_STS] is set. | WO |
| 4 | HCYC_EN | 0 | **Host Cycle Interrupt Enable .**<br>0 = (Disabled) no interrupts are generated when these bits are set.<br>1 = (Enabled) the SMBus host controller status bits, PM00 +E0 bits [TO_STS, HCYC_STS, PRERR_STS, COL_STS, ABRT_STS], are enabled to generate SMI or SCI interrupts. | RWC |
| 3 | HOST_STC | 0 | **Host Start Command.**<br>When this bit is set by software, the SMBus host logic initiates the SMBus cycle specified by CYCTYPE. Writes to this field are ignored while PM00 +E0h bit[HST_BSY] is active. | WO |
| 2-0 | CYC_Type | 000 | **Cycle Type.** Host-generated SMBus cycle type.<br>This field specifies the type of SMBus cycle that is generated when it is initiated by the HOSTST command. This field is decoded as shown in Table 81 below. | RW |

**Table 81. SM Bus Cycle Type Encoding**

| Cycle Type | SM Bus Cycle Type | Registers (For each of these, the slave address is specified by PM00 +E4h bits[7:1] and receive or read versus send or write is specified by PM00 +E4 bit[0]) |
|---|---|---|
| 000 | Quick command | Data bit in PM00 +E4h bit[0] |
| 001 | Receive or send byte | Data in PM00 +E6h bits[7:0]. If the address in PM00 +E4h is 'b0001_1001 and data received is 'b111_0xxx, then another byte will be received in PM00 +E6h bits[15:8]; see the SMBALERT description in the system management section of this document. |
| 010 | Read or write byte | Command in PM00 +E8h; data in PM00 +E6h bits[7:0] |
| 011 | Read or write word | Command in PM00 +E8h; data in PM00 +E6h bits[15:0] |
| 100 | Process call | Command in PM00 +E8h; write data is placed in PM00 +E6h bits[15:0]; then this data is replaced with the read data in the second half of the command |
| 101 | Read or write block | Command in PM00 +E8h; count data in PM00 +E6h bits[5:0]; block data in the PM00 +E9h FIFO |
| 11x | Reserved | |

Writes to the cycle type field are ignored while PM00 +E0h bit[HST_BSY] is active.

**SM Bus Host Address Register** **PM00 + Offset E5h–E4h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15-8 | HST10BA | 00000000 | **Host 10 Bit Address LSBs .**<br>These bits store the second byte of the address, used in 10-bit SMBus host-as-master transfers. If HSTADDR == 'b1111_0xx, then the cycle is specified to use 10-bit addressing. If HSTADDR is any other value, then HST10BA is not utilized. HST10BA are the upper 8 bits and the least significant 2 bits come from the HSTADDR field. | RW |
| 7-1 | HSTADDR | 0000000 | **Host Cycle Address.**<br>These bits specify the 7-bit address to the SMBus generated by the host (as a master) during SMBus cycles that are initiated by PM00 +E2 bit[HOSTST]. | RW |
| 0 | RC | 0 | **Read Cycle.**<br>0 = specifies that the cycle generated by a write to PM00 +E2 bit[HOSTST] is a write cycle or send command.<br>1 = specifies that the cycle generated by a write to PM00 +E2 bit[HOSTST] is a read cycle or receive command. | RW |

**SM Bus Host Data Register** **PM00 + Offset E7h–E6h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15-0 | HSTDATA | 00000000 00000000 | **Host Data).** This register is written to by software to specify the data to be passed to the SMBus during write and send cycles. It is read by software to specify the data passed to host controller by the SMBus during read and receive cycles. Bit[0] specifies the data written or read during the quick command cycle. Bits[7:0] specify the data for byte read and write cycles, send byte cycles, and receive byte cycles. Bits[15:0] are used for word read and write cycles and process calls. Bits[5:0] are used to specify the count for block read and write cycles. | RW |

**SM Bus Host Command Field Register** **PM00 + Offset E8h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | Host Command | 00000000 | **Host Command.** This specifies the command field passed to the SMBus by the host controller during read byte, write byte, read word, write word, process call, block read, and block write cycles. Host cycles are initiated by PM00 + E2h bit[HOSTST]. | WO |

**SM Bus Host Block Data FIFO Access Port** **PM00 + Offset E9h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | Host Command | 00000000 | **Host Data FIFO.** For block write commands, software loads 1 to 32 bytes into this port before sending them to the SMBus via the PM00 + E2h bit[HOSTST] command. For block read commands, software read 1 to 32 bytes from this port after the block read cycle is complete. If, during a block read or write, an error occurs, then the FIFO is flushed by the hardware. Read and write accesses to this port while the host is busy (PM00 +E0h bit[HST_BSY]) are ignored. | RW |

**SM Bus Host as Slave Data Register** **PM00 + Offset EBh–EAh**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15-0 | HSLVDATA | 00000000 00000000 | **Host-as-Slave-Data.** When the SMBus logic determines that the current SMBus cycle is directed to the host's slave logic (because the address matches PM00 + EEh), then the data transmitted to the AMD-756 during the cycle is latched in this register. Also, if the address matches the snoop address in PM00 + EFh, then the cycle is assumed to be a write word and the data is stored in this register. | RO |

The register above is powered by the VDD_SOFT plane and reset by RST_SOFT rather than PWRGD.

**SM Bus Host-as-Slave Device Address Register** **PM00 + Offset EDh–ECh**

| Bit | Name | Default | Description | 0-state |
|---|---|---|---|---|
| 15-8 | HSLV10DA | 00000000 | **Host-as-Slave 10 Bit Address LSBs.**<br>These bits store the second byte of the address, used in 10-bit SMBus host-as-master transfers. If HSTADDR == 'b1111_0xx, then the cycle is specified to use 10-bit addressing. If HSTADDR is any other value, then HST10BA is not utilized. HST10BA are the upper 8 bits and the least significant 2 bits come from the HSTADDR field. | RO |
| 7-1 | HSLVDA | 0000000 | **Host-as-Slave Device Address.**<br>This field stores the second byte of the device address used in 10-bit SMBus transfers to the host as a slave. If HSLVDA == 'b1111_0xx, then the cycle is specified by the SMBus specification to transmit a 10-bit device address to the host-as-slave logic and the second byte of that device address is stored in this field. If HSLVDA is any other value, then HSLV10BA is ignored. | RO |
| 0 | SNPL | 0 | **Snoop Command LSB.**<br>If the SMBus cycle address matches PM00+EFh, then the cycle is assumed to be a write word. The LSB of the command field for the cycle is placed in this bit (and the other 7 bits are placed in HSLVDA). | RO |

The register above is powered by the VDD_SOFT plane and reset by RST_SOFT rather than PWRGD.

**SM Bus Host-as-Slave Host Address Register** **PM00 + Offset EEh**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-1 | HSLVDDR | 0001000 | **Host-as-Slave Address.**<br>The SMBus logic compares the address generated by masters over the SMBus to this field to determine if there is a match (also, for a match to occur, the read-write bit is required to specify a write command). If a match occurs, then the cycle is assumed to be a write word command to the host, with the slave's device address transmitted during the normal command phase. The device address is captured in PM00+ECh and the data is capture in PM00+EAh for the cycle. After the cycle is complete, PM00+E0h bit[HSLV_STS] is set. | RW |
| 0 | | 0 | **Reserved.** This bit must remain 0 for proper operation. | RW |

The register above is powered by the VDD_SOFT plane and reset by RST_SOFT rather than PWRGD.

**SM Bus Snoop Address Register** **PM00 + Offset EFh**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-1 | SNPADDR | 0001000 | **Snoop Address.**<br>The SMBus logic compares the address generated by masters over the SMBus to this field to determine if there is a match (regardless as to whether it is a read or a write). If there is a match, then PM00+E0h bit[SNP_STS] is set high after the cycle completes. If the address specified here matches PM00+EEh, then PM00+E0h bit[SNP_STS] will not be set high. | RW |
| 0 | | 0 | **Reserved.** This bit must remain 0 for proper operation. | RW |

The register above is powered by the VDD_SOFT plane and reset by RST_SOFT rather than PWRGD.

# 7.8 Function 4 Registers (USB)

This section describes the universal serial bus (USB) control registers of the AMD-756 peripheral bus controller. This controller is compatible with the USB specifications.

## 7.8.1 Function 4 USB Configuration

**Vendor ID** **Function 4 Offset 01h–00h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15-0 | Vendor ID | 00010000<br>00100010 | Vendor ID = 1022h | RO |

**Device ID** **Function 4 Offset 03h–02h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15-0 | Device ID | 01110100<br>00000100 | Device ID = 740Ch | RO |

**Command Register** **Function 4 Offset 05h–04h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15-10 | | 000000 | **Reserved.** Always reads zero. | |
| 9 | B2B | 0 | **Fast back-to-back enable.** Input port value | Always 0 |
| 8 | SERR | 0 | **SERR# Response.** SERR# is asserted when it detects an address parity error.<br>0 = SERR# disabled 1 = SERR# enabled | RW |
| 7 | WCC | 0 | **Wait Cycle Control.**<br>The AMD-756 controller does not insert a wait state between the address and data on the AD lines. | Always 0 |
| 6 | PERR | 0 | **Parity Error Response.** The AMD-756 does not have a PERR# pin. | Always 0 |
| 5 | VGA | 0 | **VGA Palette Snooping.** Always reads 0. | Always 0 |
| 4 | MWI | 0 | **Memory Write Invalidate Command.**<br>The Memory Write and Invalidate Command will only occur if the cache line size is set to 32 bytes and the memory write is exactly one cache line.<br>0 = Disabled 1 = Enable memory write and invalidate commands | RW |
| 3 | SCM | 0 | **Special Cycle Monitoring.** Always reads 0. | Disabled |
| 2 | BM | 0 | **Bus Master.** 0 = Disabled 1 = Enabled | RW |
| 1 | MEM | 0 | **Memory Space.**<br>0 = Disabled<br>1 = Enables the ability to access the memory mapped space of the specified IDE controller. | RW |
| 0 | I/O | 0 | **I/O Space.**<br>0 = The device does not respond to any I/O addresses for either compatible or native mode.<br>1 = Enabled to respond to I/O addresses | RW |

**Status** **Function 4 Offset 07h–06h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15 | DPE | 0 | **Parity Error Detected.** Always reads 0. | R |
| 14 | SERR | 0 | **SERR Error.** Always reads 0. | R |
| 13 | RIA | 0 | **Received Initiator Abort.** This bit is set by the USBC initiator when its transaction is terminated with Initiator Abort.<br>0 = PCI transactions proceeding normally.<br>1 = The AMD-751 system controller has detected that a transaction was terminated before completion. | RWC |
| 12 | RTA | 0 | **Received Target Abort (RWC).** The USBC receives a target abort.<br>This bit is set by simultaneously deasserting DEVSEL# and asserting STOP#.<br>0 = No abort received 1 = Transaction aborted by target | |
| 11 | STA | 0 | **Signaled Target Abort.** Always reads 0. The USBC signals target abort.<br>0 = No abort received 1 = Transaction aborted by USBC | No abort signaled |
| 10-9 | DEVSEL# Timing | 01 | **DEVSEL# Timing (always reads 01).**<br>This field indicates that the slowest DEVSEL# timing is medium.<br>The AMD-756 peripheral bus controller only implements the medium timing.<br>Bits / DEVSEL# Speed: 00 Fast; 01 Medium; 10 Slow; 11 Reserved | Always reads 01 |
| 8 | PPE | 0 | **PCI Parity Error Detected.** Always reads 0. The AMD-756 peripheral bus controller does not have a PERR# pin. | Always reads 0 |
| 7 | FBBC | 0 | **Fast Back-to-Back Capability.** The AMD-756 peripheral bus controller can accept fast back-to-back transactions from the different agents. | Always reads 1 |
| 6-0 | | 0 | **Reserved.** Always reads 0. | |

**Revision ID** **Function 4 Offset 08h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | Silicon Revision Code | 00000101 | The current silicon revision code is 05h = revision A. | RO |

**Programming Interface** **Function 4 Offset 09h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | | 00010000 | This register defines the programming interface used as Open HCI. | |

**Sub Class Code** **Function 4 Offset 0Ah**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | | 0000011 | This register defines the Sub Class used as USB. | |

**Base Class Code** **Function 4 Offset 0Bh**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | | 00001100 | The Base Class Code 0Ch indicates a serial bus controller. | |

**Cache Line Size** **Function 4 Offset 0Ch**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | | 00001000 | This register identifies the system cache line size in units of 32-bit words. The USBC will only store the value of bit 3 in this register since the cache line size of 32 bytes is the only value applicable to the design. Any value other than 08h written to this register will be read back as 00h. | R/W |

**Latency Timer** **Function 4 Offset 0Dh**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | | 00010000 | **Latency Timer.** This register identifies the value of the latency timer in PCI clocks. | R |

**Header Type** **Function 4 Offset 0Eh**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | | 00000000 | This register contains the value 00h. | RO |

**BIST** **Function 4 Offset 0Fh**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | | 00000000 | This register contains the value 00h. The USBC dos not implement built in self test. | RO |

**Base Address Register 0** **Function 4 Offset 13h–10h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-12 | Base Address High | 0 | **Base Address High.**<br>This field is loaded by BIOS software to determine the base address A[30:11] of the memory-mapped USB-OHCI registers. | RW |
| 11-4 | Base Address Low | 0 | **Base Address Low.**<br>This field is cleared to indicate that 4 Kbytes is allocated to USB memory-mapped control registers and that the registers reside at a 4-Kbyte boundary per PCI Specification 2.1. | always reads 0 |
| 3 | PRE | 1 | **Prefetchable.**<br>This bit indicates that this range can not be prefetched.<br>Note: Discrepancy between definition and figure for value of bit. | always reads 0 |
| 2-1 | TYPE | 00 | **Type.**<br>These bits are cleared to indicate that BAR0 is 32 bits wide and mapping can be performed anywhere in the 32-bit address space. | always read 0 |

**Base Address Register 0 (continued)** **Function 4 Offset 13h–10h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 0 | MEM | 0 | **Memory.** This bit is cleared to indicate that BAR0 maps into memory space. | always reads 0 |

This register is used by the USB driver software to set the memory location of the OHCI memory-mapped control registers.

**Interrupt Line Register** **Function 4 Offset 3Ch**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | | 00000000 | This register contains the default value 00h. It identifies which of the system interrupt controllers the devices interrupt pin is connected to. The value of this register is used by device drivers and has no direct meaning to USBC. | RO |

**Interrupt Pin Register** **Function 4 Offset 3Dh**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | | 00000100 | This register identifies which interrupt pin a device uses. Since USBC uses INTD#, this value is set to 04h. | RO |

**Min Grant Register** **Function 4 Offset 3Eh**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | | 00000000 | This register contains the value 00h. It specifies the desired settings for how long a burst USBC needs assuming a clock rate of 33 MHz. The value specifies a period of time in units of 1/4 microsecond. | RO |

**Max Latency Register** **Function 4 Offset 3Fh**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-0 | | 00000000 | This register contains the value 00h. It specifies the desired settings for how often USBC needs access to the PCI bus assuming a clock rate of 33 MHz. The value specifies a period of time in units of 1/4 microsecond. | RO |

**Operational Mode Register** **Function 4 Offset 44h**

| Bit | Name | Default | Description | 0-state |
|---|---|---|---|---|
| 7-1 | | 000000 | **Reserved.**<br>This register selects which operational mode is enabled. Bits defined as write-only are read as 0's. | WO |
| 0 | DB16 | 0 | **Data Buffer Region 16 (RW).**<br>0 = 323-byte data buffer.<br>1 = 16-byte data buffer. This bit is set when OverCurrentIndicator changes. | RW |

## 7.8.2 USB Memory Mapped Registers (Open HCI Registers)

These memory mapped registers are relative to Function 4 Offset 10 (BAR0).

**HcRevision** **BAR0 + Offset 01h–00h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15-9 | | 0 | **Reserved.** Always read 0. | RO |
| 8 | LEG | 1 | **Legacy .** Always reads 1.<br>Indicates that the legacy support registers are present in USB Controller. | RO |
| 7-4 | Major Revision | 0001 | **Major Revision.**<br>Indicates the OpenHCI Specification major revision number implemented. | RO |
| 3-0 | Minor Revision | 0000 | **Minor Revision.**<br>Indicates the OpenHCI Specification minor revision number implemented. | RO |

**HcControl** **BAR0 + Offset 05h–04h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15-11 | | 0 | **Reserved.** Always read 0. | |
| 10 | RWE | 0 | **Remote Wakeup Enable.** Always reads 1.<br>Indicates that remote wakeup is always enabled.<br>Note: Discrepancy between default definitions in old figure and in text. | RO |
| 9 | RW | 1 | **Remote Wakeup Connected .** Always reads 1.<br>Indicates that the USBC supports a remote wakeup signal. | RO |
| 8 | IR | 0 | **Interrupt Routing.**<br>0 = Interrupts routed to normal interrupt mechanism (INT)<br>1 = Interrupts routed to SMI | RW |
| 7-6 | State | 00 | **Host Controller Functional State .**<br>Sets the Host Controller state:<br>Bits / Host Controller State:<br>00 USB reset<br>01 USB resume<br>10 USB operational<br>11 USB suspend<br>*The Host Controller may force a state change from USBSUSPEND to USBRESUME after detecting resume signaling from a downstream port.* | RW |
| 5 | BLE | 0 | **Bulk List Enable.**<br>0 = bulk list processing disabled<br>1 = bulk list processing enabled | RW |

**HcControl (continued)** **BAR0 + Offset 05h–04h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 4 | CLE | 0 | **Control List Enable.**<br>0 = control list processing disabled<br>1 = control list processing enabled | RW |
| 3 | IE | 0 | **Isochronous Enable.**<br>0 = Disables the Isochronous List when the Periodic List is enabled (so Interrupt EDs may be serviced). While processing the Periodic List, the Host Controller will check this bit when it finds an isochronous endpoint descriptor (ED).<br>1 = Enables the Isochronous List. | RW |
| 2 | PLE | 0 | **Periodic List Enable.**<br>This bit enables processing of the Periodic (interrupt and isochronous) list. The Host Controller checks this bit prior to attempting any periodic transfers in a frame.<br>0 = Disabled 1 = Enabled | RW |
| 1-0 | CBSR | 00 | **Control Bulk Service Ratio (RW).**<br>Specifies the number of Control Endpoints serviced for every Bulk Endpoint. Encoding is N+1 where N is the number of Control Endpoints.<br>**Bits** / **Number of Control Endpoints**<br>00 / 1<br>01 / 2<br>10 / 3<br>11 / 4 | RW |

**HcCommand** **BAR0 + Offset 08h**

| Bit | Name | Default | Description | 0-state |
|---|---|---|---|---|
| 7-4 | | 0 | **Reserved.** Always reads 0. | |
| 3 | OCR | 0 | **Ownership Change Request.**<br>1 = Sets the OwnershipChange field in HcInterruptStatus register. The bit is cleared by software. | RW |
| 2 | BLF | 0 | **Bulk List Filled.**<br>0 =Begin processing the head of the bulk list. Cleared by the Host Controller each time it begins processing the head of the Bulk List.<br>1 = Indicates there is an active ED on the Bulk List. The bit may be set by either software or the Host Controller. | RW |
| 1 | CLF | 0 | **Control List Filled.**<br>0 = Begin processing the head of the control list. The bit is cleared by the Host Controller each time it begins processing the head of the Control List.<br>1= Indicates there is an active ED on the Control List. The bit may be set by either software or the Host Controller. | RW |
| 0 | HCR | 0 | **Host Controller Reset (RW).**<br>0 = No reset, cleared by the Host Controller upon completion of the reset operation.<br>1 = Start a reset operation. | RW |

**HcStatus** **BAR0 + Offset 0Ah**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 7-2 | | 000000 | **Reserved.** Always reads 0. | |
| 1-0 | | 00 | **Schedule Overrun Count.**<br>This field increments every time the SchedulingOverrun bit in HcInterruptStatus register is set. The count wraps from '11' to '00'.<br>Bits / Schedule Overrun Count:<br>00 — 0<br>01 — 1<br>10 — 2l<br>11 — 3 | RW |

**HcInterrupt Status** **BAR0 + Offset 0Dh–0Ch**

| Bit | Name | Default | Description | Access Type | 1-state |
|---|---|---|---|---|---|
| 15-7 | | 0 | **Reserved.** Must remain 0 for proper operation. | RW | |
| 6 | RHS | 0 | **Root Hub Status Change.**<br>This bit is set when the content of HcRhStatus or the content of any HcRhPortStatus register has changed. | RO | |
| 5 | FNO | 0 | **Frame number Overflow.**<br>This bit is set when bit 15 of FrameNumber changes value from '0' to '1' or from '1' to '0'. | RW | |
| 4 | ERR | 0 | **Unrecoverable Error.** (Always reads 0. Not implemented. | | |
| 3 | RD | 0 | **Resume Detected.** This bit is set when the Host Controller detects resume signaling on a downstream port.<br>0 = Resume not detected 1 = Resume detected | RW | |
| 2 | SOF | 0 | **Start of Frame.**<br>This bit is set when the Frame Management block signals a 'Start of Frame' event. | RW | |
| 1 | WBDH | 0 | **Writeback Done Head.**<br>This bit is set after the Host Controller has written HcDoneHead to HccaDoneHead. | RW | |
| 0 | SO | 0 | **Scheduling Overrun.**<br>0 = No overrun<br>1 = List processor determines a schedule overrun has occurred | RW | |

### HcOwnership Status — BAR0 + Offset 0Fh–0Eh

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15 | | 0 | **Reserved.** Must remain 0 for proper operation. | RW |
| 14 | | 0 | **Ownership Change.** This bit is set when the OwnershipChangeRequest bit of HcCommandStatus is set. | RO |
| 13-0 | | 0 | **Reserved.** Must remain 0 for proper operation. | RW |

### HcInterrupt Enable — BAR0 + Offset 11h–10h

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15-7 | | 0 | **Reserved.** Must remain 0 for proper operation. | RW |
| 6 | RHSE | 0 | **Root Hub Status Change Enable.** 0 = Ignore 1 = Enable interrupt generation | RW |
| 5 | FNOE | 0 | **Frame Number Overflow Enable.** 0 = Ignore 1 = Enable interrupt generation | RW |
| 4 | ERRE | 0 | **Unrecoverable Error.** Always reads 0. Not implemented. | |
| 3 | RDE | 0 | **Resume Detected Enable.** 0 = Ignore 1 = Enable interrupt generation | RW |
| 2 | SOFE | 0 | **Start of Frame Enable.** 0 = Ignore 1 = Enable interrupt generation | RW |
| 1 | WBDE | 0 | **Writeback Done Head Enable.** 0 = Ignore 1 = Enable interrupt generation | RW |
| 0 | SOE | 0 | **Scheduling Overrun Enable.** 0 = Ignore 1 = Enable interrupt generation | RW |

### HcOwnership Enable — BAR0 + Offset 13h–12h

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15 | MIE | 0 | **Master Interrupt Enable.** 0 = Ignore 1 = Enable interrupt generation | RW |
| 14 | OCE | 0 | **Ownership Change Enable.** 0 = Ignore 1 = Enable interrupt generation | RO |
| 13-0 | | 0 | **Reserved.** Must remain 0 for proper operation. | RW |

### HcInterrupt Disable — BAR0 + Offset 15h–14h

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15-7 | | 0 | | |
| 6 | RHSD | 0 | **Root Hub Status Change Disable.** 0 = Ignore 1 = Disable interrupt generation | RW |
| 5 | FNOD | 0 | **Frame Number Overflow Disable.** 0 = Ignore 1 = Disable interrupt generation | RW |
| 4 | ERRD | 0 | **Unrecoverable Error.** Always reads 0. Not implemented. | |
| 3 | RDD | 0 | **Resume Detected Disable.** 0 = Ignore 1 = Disable interrupt generation | RW |
| 2 | SOFD | 0 | **Start of Frame Disable.** 0 = Ignore 1 = Disable interrupt generation | RW |
| 1 | WBDD | 0 | **Writeback Done Head Disable.** 0 = Ignore 1 = Disable interrupt generation | RW |
| 0 | SOD | 0 | **Scheduling Overrun Disable.** 0 = Ignore 1 = Disable interrupt generation | RW |

**HcOwnership Disable** **BAR0 + Offset 17h–16h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15 | MID | 0 | **Master Interrupt Disable.** 0 = Ignore 1 = Disable interrupt generation | RW |
| 14 | OCD | 0 | **Ownership Change Disable.** 0 = Ignore 1 = Disable interrupt generation | RO |
| 13-0 | | 0 | **Reserved.** Must remain 0 for proper operation. | RW |

**HcHCCA** **BAR0 + Offset 1Bh–18h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-8 | HCCA | 0 | **Pointer to HCCA Base Address .** | RW |
| 7-0 | | 0 | **Reserved.** Must remain 0 for proper operation. | RW |

**HcPeriod Current ED** **BAR0 + Offset 1Fh–1Ch**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-4 | HCCA | 0 | **Period Current ED.** Pointer to the current periodic list ED. | RW |
| 3-0 | | 0 | **Reserved.** Must remain 0 for proper operation. | RW |

**HcControl Head ED** **BAR0 + Offset 23h–20h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-4 | CHA | 0 | **Control Head Address.** Pointer to the control list head ED. | RW |
| 3-0 | | 0 | **Reserved.** Must remain 0 for proper operation. | RW |

**HcControl Current ED** **BAR0 + Offset 27h–24h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-4 | HCC | 0 | **Current Control ED.** Pointer to the current control list ED. | RW |
| 3-0 | | 0 | **Reserved.** Must remain 0 for proper operation. | RW |

**HcBulk Head ED** **BAR0 + Offset 2Bh–28h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-4 | BLH | 0 | **Bulk List Head ED.** Pointer to the bulk list head ED. | RO |
| 3-0 | | 0 | **Reserved.** Must remain 0 for proper operation. | RW |

**HcBulk Current ED** **BAR0 + Offset 2Fh–2Ch**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-4 | BCL | 0 | **Bulk Current List ED.** Pointer to the current bulk list ED. | RO |
| 3-0 | | 0 | **Reserved.** Must remain 0 for proper operation. | RW |

### HcDone Head ED — BAR0 + Offset 33h–30h

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-4 | DLH | 0 | **Done List Head ED.** Pointer to the done list head ED. | RO |
| 3-0 | | 0 | **Reserved.** Must remain 0 for proper operation. | RW |

### HcFrame Interval — BAR0 + Offset 37h–34h

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31 | FI | 0 | **Frame Interval Toggle.** This bit is toggled by HCD whenever it loads a new value into the Frame Interval Field. | |
| 30-16 | FS Largest Data Packet | 0 | **FS Largest Data Packet.** Must remain 0 for proper operation. | RW |
| 15-14 | | 0 | **Reserved.** Must remain 0 for proper operation. | RW |
| 13-0 | Frame Interval | 10111011 011111 | **Frame Interval.** This field specifies the length of a frame as (bit times - 1). For 12,000 bit times in a frame, a value of 11,999 is stored here. | RW |

### HcFrame Remaining — BAR0 + Offset 3Bh–38h

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31 | FR | 0 | **Frame Remaining Toggle.** This bit is toggled by HCD whenever it loads a new value into the Frame Remaining field. | RW |
| 30-14 | | 0 | **Reserved.** Must remain 0 for proper operation. | RW |
| 13-0 | Frame Remaining | 0 | **Bits 13–0Frame Remaining (RW).** This field is a 14 bit decrementing counter used to time a frame. When the Host Controller is in the USBOPERATIONAL state the counter decrements each 12 MHz clock period. When the count reaches 0, the end of a frame has been reached. The counter reloads with FrameInterval at that time. In addition, the counter loads when the Host Controller transitions into the USBOPERATIONAL state. | RW |

### HcFrame Number — BAR0 + Offset 3Fh–3Ch

| Bit | Name | Default | Description | 0-state |
|---|---|---|---|---|
| 31-16 | | 0 | **Reserved.** Must remain 0 for proper operation. | RW |
| 15-0 | Frame Number | 0 | **Frame Number.** This field is a 16 bit incrementing counter. The count is incremented coincident with the loading of FrameRemaining. The count will roll over from 'FFFFh' to '0h.' Writing to this register while the USB controller is operational will have unexpected results. | RW |

### HcPeriodic Start — BAR0 + Offset 43h–40h

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-14 | | 0 | **Reserved.** Must remain 0 for proper operation. | RW |
| 13-0 | Periodic Start | 0 | **Periodic Start.** This field contains a value used by the List Processor to determine where in a frame the Periodic List processing must begin. | RW |

**HcLS Threshold** **BAR0 + Offset 47h–44h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-12 | | 0 | **Reserved.** Must remain 0 for proper operation. | RW |
| 11-0 | LST | 0 | **Low Speed Threshold.**<br>This field contains a value used by the Frame Management block to determine whether or not a low speed transaction can be started in the current frame. | RW |

**HcRh Descriptor A1** **BAR0 + Offset 49h–48h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15-13 | | 0 | **Reserved.** Must remain 0 for proper operation. | RW |
| 12 | NOCP | 0 | **No Over-current Protection Mode.**<br>0 = Over-current status is reported 1 = Over-current status is not reported<br>*This bit should be set to support the external system port over current implementation.* | RW |
| 11 | OCP | 0 | **Over-current Protection Mode.**<br>0 = Global over-current protection 1 = Individual over-current protection<br>*This bit is valid only when no over-current protection is cleared.* | RW |
| 10 | DT | 0 | **Device Type.** 0 = Not a compound device 1 = compound device | RO |
| 9 | NPS | 0 | **No Power Switching.** 0 = Ports are power switched 1 = Ports are always on<br>*This bit should be set to support external system port power switching.* | RW |
| 8 | PSM | 0 | **Power Switching Mode.** 0 = Global switching 1 = Individual switching<br>*This bit is valid only when no power switching is cleared.* | RW |
| 7-0 | Number of Down Stream Ports | 00000100 | **Number of Down Stream Ports.** 4 ports are supported. | RO |

**HcRh Descriptor A2** **BAR0 + Offset 4Bh–4Ah**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15-10 | | 0 | **Reserved.** Always read 0. | |
| 9-8 | Power Good Time | 01 | **Power Good Time .**<br>Power switching is effective within 2 ms. The field value is the number of 2 ms intervals. Only bits [9:8] are implemented as R/W. The remaining bits are read only as '0'. It is not expected that these bits be written to anything other than 1h, but limited adjustment is provided. This field should always be written to a non-zero value. | RW |
| 7-0 | | 0 | **Reserved.** Must remain 0 for proper operation. | RW |

| Bits | Power Good Time |
|---|---|
| 00 | 0 milliseconds |
| 01 | 2 milliseconds (default) |
| 10 | 4 milliseconds |
| 11 | 6 milliseconds |

**HcRh Descriptor B** **BAR0 + Offset 4Fh–4Ch**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-16 | Power Control Mask | 0 | **Power Control Mask.**<br>NoPowerSwitching is cleared and PowerSwitchingMode is set (individual port switching).<br>0 =The port only responds to global power switching commands (Set/ClearGlobalPower).<br>1 = The port only responds to individual port power switching commands (Set/ClearPortPower).<br>Bit 16 corresponds to port zero and is reserved. Bit 17 corresponds to port 1, bit 18 to port 2, etc.<br>*Unimplemented ports are reserved, read/write '0'.* | RW |
| 15-0 | Device Removable | 0 | **Bits 15 –0Device Removable.**<br>0 = Device Removable<br>1 = Device not removable<br>*Bit 0 corresponds to port zero and is reserved. Bit 1 corresponds to port 1, bit 2 to port 2, etc.*<br>*Unimplemented ports are reserved, read/write '0'.* | RW |

**HcRh Status A** **BAR0 + Offset 51h–50h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15 | RWE | 0 | **Remote Wakeup Enable.**<br>0 = Disabled<br>1 = Enables ports' ConnectStatusChange as a remote wakeup event. and also reports remote wake up status.<br>*Writing a '0' has no effect. This function is disabled by writing offset 52 bit 15.* | RW |
| 14-2 | | 0 | **Reserved.** Always read 0. | |
| 1 | OCI | 0 | **Over Current Indicator.** This bit reflects the state of the OVRCUR pin.<br>0 = No over-current condition 1 = Over-current condition<br>*This field is only valid if NoOverCurrentProtection and OverCurrentProtectionMode are cleared.* | RO |
| 0 | LPS | 0 | **Local Power Status .** Always reads 0. Not supported.<br>*(read) Always reads '0'.*<br>*(write) ClearGlobalPower Writing a '1' issues a ClearGlobalPower command to the ports. Writing a '0' has no effect.* | |

**HcRh Status B** **BAR0 + Offset 53h–52h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15 | CRW | 0 | **Clear Remote Wakeup.**<br>This bit disables ports' ConnectStatusChange as a remote wakeup event.<br>0, 1 = no effect<br>*Status is reported in offset 50 bit 15.* | WO |
| 14-2 | | 0 | **Reserved.** Always read 0. | |

**HcRh Status B (continued)** **BAR0 + Offset 53h–52h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 1 | OCIC | 0 | **Over Current Indicator Change.**<br>0 = No change 1 = OverCurrentIndicator changes.<br>*This field is only valid if NoOverCurrentProtection and OverCurrentProtectionMode are cleared.* | RO |
| 0 | LPSC | 0 | **Local Power Status change (Always reads 0).** Not supported.<br>*(read) Always reads '0'.*<br>*(write) SetGlobalPower Writing a '1' issues a SetGlobalPower command to the ports. Writing a '0' has no effect.* | |

**HcRH Port 1 Status** **BAR0 + Offset 57h–54h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-21 | | 0 | **Reserved.** These bits must remain 0 for proper operation. | RW |
| 20 | PRSC | 0 | **Port Reset Status Change.**<br>0 = Port reset is not complete. 1 = Port reset signal is complete. | RWC |
| 19 | POCC | 0 | **Port Over Current Indicator Change.**<br>0 = No change. 1 = OverCurrentIndicator changed. | RWC |
| 18 | PSSC | 0 | **Port Suspend Status Change.**<br>0 = Port has not resumed. 1 = The selective resume sequence for the port is completed. | RWC |
| 17 | PESC | 0 | **Port Enable Status Change.**<br>0 = Port has not been disabled.<br>1 = The port has been disabled due to a hardware event (cleared PortEnableStatus). | RWC |
| 16 | CCSC | 0 | **Connection Change Status.**<br>0 = No connect or disconnect event detected.<br>1 = indicates hardware detection of a connect or disconnect event.<br>*If DeviceRemoveable is set, this bit is set to '1'.* | RW |
| 15-10 | | 0 | **Reserved.** Always read 0. | |
| 9 | LSD | 0 | **Low Speed Device Attached .** This bit defines the speed of the attached device.<br>It is only valid when CurrentConnectStatus is set.<br>0 = Full speed device 1 = Low speed device | RW |
| 8 | PPS | 0 | **Port Power Status.**<br>This bit reflects the power state of the port regardless of the power switching mode.<br>0 = Port power is off. 1 = Port power is on.<br>*If NoPowerSwitching is set, this bit is always read as '1'.* | RWC |
| 7-5 | | 0 | **Reserved.** These bits must remain 0 for proper operation. | RW |
| 4 | PRS | 0 | **Port Reset Status .**<br>0 = Port reset signal is not active 1 = Port reset active<br>*Writing a '1' to the SetPortReset bit sets PortResetStatus. Writing a '0' has no effect.* | RWC |

**HcRH Port 1 Status (continued)** **BAR0 + Offset 57h–54h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 3 | POCI | 0 | **Port Over Current Indicator.**<br>PortOverCurrentIndicator supports global over-current reporting. This bit reflects the state of the OVRCUR pin dedicated to this port. This field is only valid if NoOverCurrentProtection is cleared and OverCurrentProtectionMode is set.<br>0 = No over-current condition 1 = Over-current condition<br>*Writing a '1' to the ClearPortSuspend bit initiates the selective resume sequence for the port. Writing a '0' has no effect.* | RWC |
| 2 | PSS | 0 | **Port Suspend Status.**<br>This bit is set when the port is selectively suspended.<br>0 = Port is not suspended. 1 = Port is selectively suspended.<br>*Writing a '1' to the SetPortSuspend bit sets PortSuspendStatus. Writing a '0' has no effect.* | RWC |
| 1 | PES | 0 | **Port Enable Status.**<br>0 = Port disabled 1= Port enabled<br>*Writing a '1' to the SetPortEnable bit sets PortEnableStatus. Writing a '0' has no effect.* | RWC |
| 0 | CCS | 0 | **Current Connection Status.** 0 = No device connected 1 = Device connected<br>*(read) If DeviceRemoveable is set (not removable) this bit is always '1'.*<br>*(write) Writing a '1' to the ClearPortEnable bit clears PortEnableStatus. Writing a '0' has no effect.* | RWC |

**HcRH Port 2 Status BAR0 + Offset 5Bh–58h**

**HcRH Port 3 Status BAR0 + Offset 5Fh–5Ch**

**HcRH Port 4 Status BAR0 + Offset 63h–60h**

These registers have the same bit definitions as the above.

**HcEnable and Control** **BAR0 + Offset 101h–100h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 15-9 | | 0 | **Reserved.** Always read 0. | |
| 8 | A20S | 1 | **A20 State.**<br>Indicates current state of Gate A20 on keyboard controller. Used to compare against value written to 60h when GateA20Sequence is active.<br>0 = A20 inactive 1 = A20 active | RW |
| 7 | IRQ12 | 0 | **IRQ12 Active.**<br>0 = No transition<br>1 = Indicates that a positive transition on IRQ12 from keyboard controller has occurred.<br>Software can write a 1 to this bit to clear it to 0 SW write of a 0 has no effect. | RWC |

**HcEnable and Control (continued)** **BAR0 + Offset 101h–100h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 6 | IRQ1 | 0 | **IRQ1 Active.**<br>0 = No transition<br>1 = Indicates that a positive transition on IRQ1 from keyboard controller has occurred.<br>Software can write a 1 to this bit to clear it to 0 SW write of a 0 has no effect. | RWC |
| 5 | GA20S | 0 | **Gate A20 Sequence.**<br>0 = Indicates a data value other than D1h was written to I/O port 64h<br>1 = Indicates a data value of D1h was written to I/O port 64h | RO |
| 4 | EIRQE | 0 | **External IRQ Enable.**<br>0 = Disabled<br>1 = Enabled IRQ1 and IRQ12 from the keyboard controller cause an emulation interrupt. The function controlled by this bit is independent of the setting of the EmulationEnable bit in this register. | RW |
| 3 | IRQE | 0 | **IRQ Enable.**<br>The USBC will generate IRQ1 or IRQ12 as long as the OutputFull bit in HceStatus is set to 1. If the AuxOutputFull bit of HceStatus is 0 then IRQ1 is generated and if it is 1, then an IRQ12 is generated. See Table 82 below for details.<br>0 = IRQ1 1 = IRQ12 | RW |
| 2 | CP | 0 | **Character Pending.**<br>0 = Disabled<br>1 = An emulation interrupt is generated when the OutputFull bit of the HC EnableandStatus register is cleared. | RW |
| 1 | EI | 0 | **Emulation Interrupt.**<br>This bit is a static decode of the emulation interrupt condition.<br>0 = No interrupt 1 = Interrupt | RO |
| 0 | EE | 0 | **Emulation Enable.**<br>0 = Disabled<br>1 = The USBC is enabled for legacy emulation. The USBC will decode accesses to I/O registers 60H and 64H and generate IRQ1 and/or IRQ12. These interrupts invoke the emulation software. | RW |

**Table 82. Bit Relations**

| IRQ Enable | OutputFull in HceStatus | AuxOutputFull in HceStatus | Character Pending in HceControl | Action |
|---|---|---|---|---|
| 0 | x | x | x | none |
| 1 | 0 | x | 1 | emulation interrupt |
| 1 | 1 | 0 | x | IRQ1 |
| 1 | 1 | 1 | x | IRQ12 |

**HceInput** **BAR0 + Offset 107h–104h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-8 | | 0 | **Reserved.** Always reads 0. | |
| 7-0 | Data In | x | **Data In.** This register captures data that is written to port 60h and 64h. | RW |

**HceOutput** **BAR0 + Offset 10Bh–108h**

| Bit | Name | Default | Description | Access Type |
|---|---|---|---|---|
| 31-8 | | 0 | **Reserved.** Always reads 0. | |
| 7-0 | Data Out | xx0xxxxx | **Data Out.**<br>This register holds data that is returned when an I/O read of port 60h is performed by application software. | RW |

**HC-Control and Status** **BAR0 + Offset 10Ch**

| Bit | Name | Default | Description | 0-state | 1-state |
|---|---|---|---|---|---|
| 7 | PAR | 0 | **Parity.**<br>0 = No parity error 1 = Parity error on keyboard/mouse data | RW | |
| 6 | T | 0 | **Timeout.**<br>0 = No timeout 1 = Timeout occurred | RW | Timeout occurred |
| 5 | AOF | 0 | **Aux Output Full.**<br>IRQ12 is asserted whenever this bit, the OutputFull, and the IRQEn bit s are set.<br>0 = Auxiliary output not full 1 = Auxiliary output full | RW | Aux output full |
| 4 | IS | 0 | **Inhibit Switch.** This bit reflects the state of the keyboard inhibit switch.<br>0 = Inhibited 1 = Not inhibited | RW | Not inhibited |
| 3 | CD | 0 | **Command Data.**<br>0 = Indicates an I/O write to port 60h<br>1 = Indicates an I/O write to port 64h | RO | |
| 2 | FLAG | 0 | **Flag.** This bit is used as a system flag by software to indicate a warm or cold boot.<br>0 = Cold boot 1 = Warm boot | RW | Warm boot |
| 1 | IF | 0 | **Input Full .**<br>0 = No I/O write<br>1 = Except for the case of a Gate A20 sequence, one indicates an I/O write to address 60h or 64h. While this bit is set to 1 and emulation is enabled, an emulation interrupt condition exists.<br>(See Gate A20 sequence in ch5 for a detailed description.) | RW | |
| 0 | OF | 0 | **Output Full .**<br>The USBC will clear this bit on a read of I/O port 60h. If IRQEn is set and AuxOutputFull is cleared then an IRQ1 is generated as long as this bit is set. If IRQEn is set and AuxOutputFull is set then IRQ12 will be generated as long as this bit is set. While this bit is 0 and CharacterPending in HceControl is set, an emulation interrupt condition exists.<br>0 = Indicates a read of I/O port 60h.<br>1 = Generate IRQ12 if IRQEn is set and AuxOutputFull is cleared. | RW | |

## 7.8.3 I/O APIC Registers

**Memory Mapped Registers**

The I/O APIC registers are accessed by using the two 32-bit registers— IOREGSEL and IOWIN. These registers are located at fixed memory addresses FEC0_0000h and FEC0_0010h. To access any one of the IOAPIC registers listed in the table, the IO Regsel register is written with the address of the APIC register. The IOWIN register is then used to read or write the data from the IOAPIC register addressed by the IOREGSEL register. All IOAPIC registers are accessed by 32-bit loads and stores.

**IOREGSEL** **FEC0_0000h**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | | | | | | | | | | APIC Register Address | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31–8** **Reserved (Always read 0)**

**Bits 7-0** **APIC Register Address**—This field contains the index to IOAPIC registers to be accessed via the IOWIN register.

**IOWIN** **FEC0_0010h**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| APIC Register Data | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | x | x | x | x | x | x | x | x |

**Bits 31–0** **APIC Register data**—This register contains the read/write data from the IOAPIC register addressed in the IOREGSEL register.

**APIC ID-I/O APIC Identification Register**

This register contains the 4-bit APIC ID. The ID serves as a physical name of the IOAPIC. All APIC devices using the APIC bus should have a unique APIC ID. The APIC bus arbitration ID for the I/O unit is also written during a write to the APICID register (same data is loaded into both). This register must be programmed with the correct ID value before using the IOAPIC for message transmission.

## IOAPIC Identification — Index 00h

| | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Reserved | | | | ID | | | | Reserved | | | | | | | | | | | | | | | | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | x | x | x | x | x | x | x | x |

**Bits 31–8** **Reserved (Always read 0)**

**Bits 27–24** **IOAPIC ID (RW)**—This field contains the IOAPIC identification.

**Bits 23–0** **Reserved (Always read 0)**

**APIC ID-I/O APIC Version Register**

The IOAPIC version register identifies the APIC hardware version. Software can use this to provide compatibility between different APIC implementations and their versions. In addition, this register provides the maximum number of entries in the I/O redirection table.

## IOAPIC Version — Index 01h

| | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Reserved | | | | | | | | Max Rdir Entries | | | | | | | | Reserved | | | | | | | | APIC Version | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |

**Bits 31–24** **Reserved (Always read 0)**

**Bits 23–16** **Max Redirection Entries**—This field contains the entry number (0 being the lowest entry) of the highest entry in the I/O redirection table. The value is equal to the number of interrupt input pins for the IOAPIC minus one. The range of values is 0 through 239. For this IOAPIC, the value is 17h.

**Bits 15–8** **Reserved (Always read 0)**

**Bits 7–0** **APIC Version (RO)**—This 8-bit field identifies the implementation version. The version number assigned to the IOAPIC is 11h.

### APICARB Register

The APICARB register contains the bus arbitration priority for the IOAPIC. This register is loaded when the IOAPIC ID register is written.

The APIC uses a one wire arbitration to win bus ownership. A rotating priority scheme is used for arbitration. The winner of the arbitration becomes the lowest priority agent and assumes an arbitration ID of 0.

All other agents, except the agent whose arbitration ID is 15, increment their arbitration IDs by one. The agent whose ID was 15 takes the winner's arbitration ID and increments it by one. Arbitration IDs are changed (incremented or assumed) only for messages that are transmitted successfully. (Except in the case of low-priority messages where arbitration ID is changed even if the message was not successfully transmitted.) A message is transmitted successfully if no checksum error or acceptance error is reported for that message. The APICARB register is always loaded with IOAPIC ID during a *level-triggered INIT with de-assert* message.

**IOAPIC Arbitration** **Index 02h**

| | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reset | Reserved | | | | ARB_ID | | | | Reserved | | | | | | | | | | | | | | | | | | | | | | | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31–28 Reserved (Always read 0)**

**Bits 27–24 IOAPIC ARB_ID (RW)**—This field contains the IOAPIC Arbitration ID.

**Bits 23–0 Reserved (Always read 0)**

### Redirection Table Entry Registers

There are 48 I/O redirection table entry registers—located in doubleword pairs at even and odd index numbers. Each register is a dedicated entry for each interrupt input signal. Unlike IRQ pins of the 8259A, the notion of interrupt priority is completely unrelated to the position of the physical interrupt input signal on the APIC. Instead, software determines the vector (and therefore the priority) for each corresponding interrupt input signal. For each interrupt signal, the operating system can also specify the signal polarity (low active or high active), whether the interrupt is signaled as edges or levels, as well as the destination and delivery mode of the interrupt. The information in the redirection table is used to translate the corresponding interrupt pin information into an inter-APIC message.

The IOAPIC responds to an edge-triggered interrupt as long as the interrupt is wider than one CLK cycle. The interrupt input is asynchronous. Therefire, setup and hold times need to be guaranteed for at lease one rising edge of the CLK input. Once the interrupt is detected, a delivery status bit internal to the IOAPIC is set. A new edge on that interrupt input pin will not be recognized until the IOAPIC unit broadcasts the corresponding message over the APIC bus and the message has been accepted by the destination(s) specified in the destination field. That new edge only results in a new invocation of the handler if its acceptance by the destination APIC causes the interrupt request register bit to go from 0 to 1. (In other words, if the interrupt wasn't already pending at the destination.)

### Redirection Table Descriptor Even Doublewords — Even Indexes 10h to 3Eh

| | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reset | DEST | | | | | | | | Reserved | | | | | | | | | | | | | | | | | | | | | | | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | x | x | x | x | x | x | x | x |

**Bits 31–24 Destination Field (RW)**—If the destination mode of this entry is physical mode (bit 11 below = 0), bits [27:24] contain an APIC ID. If logical mode is selected (bit 11 below = 1), the destination field potentially defines a set of processors. Bits [31:24] of the destination field specify the logical destination address.
Bits 27 through 24 of the destination field specify the 4-bit APIC ID. When DESTMOD=1 (logical mode), destinations are identified by matching on the logical destination under the control of the destination format register and logical destination register in each local APIC.

**Bits 23–0 Reserved (Always read 0)**

## Redirection Table Descriptor Odd Doublewords — Odd Indexes 11h to 3Fh

| | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reset | Reserved | | | | | | | | | | | | | | | IM | TM | RIRR | POLE | DS | Mode | DELIVERY MODE | | | INTERRUPT VECTOR | | | | | | | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | x | x | x | x | x | x | x | x |

**Bits 31–17** **Reserved (Always read 0)**

**Bit 16** **Interrupt Mask (RW)**—When this bit is 1, edge-sensitive interrupts signaled on a masked interrupt pin are ignored (i.e., not delivered or held pending).

0 = Interrupt occurs (not masked) (default)

1 = Mask interrupt

Level-asserts or negates occurring on a masked level-sensitive pin are ignored and have no side effects. Changing the mask bit from unmasked to masked after the interrupt is accepted by a local APIC has no effect on that interrupt. This behavior is identical to the case where the device withdraws the interrupt before that interrupt is posted to the processor. It is software's responsibility to handle the case where the mask bit is set after the interrupt message has been accepted by a local APIC unit but before the interrupt is dispensed to the processor. When this bit is 0, the interrupt is not masked. An edge or level on an interrupt pin that is not masked results in the delivery of the interrupt to the destination.

**Bit 15** **Trigger Mode (RW)**—The trigger mode field indicates the type of signal on the interrupt pin that triggers an interrupt.

0 = Edge (default)

1 = Level

**Bit 14** **Remote IRR (RO)**—This bit is used for level-triggered interrupts. Its meaning is undefined for edge-triggered interrupts. For level-triggered interrupts, this bit is set to 1 when local APIC(s) accepts the level interrupt sent by the IOAPIC. The remote IRR bit is set to 0 when an EOI message with a matching interrupt vector is received from a local APIC.

**Bit 13** **Interrupt Input Pin Polarity (RW)**—This bit specifies the polarity of the interrupt signal.

0 = Active high (default)

1 = Active low

**Bit 12** **Delivery Status (RO)**—The delivery status bit contains the current status of the delivery of this interrupt. Delivery status is read-only and writes to this bit (as part of a 32-bit word) do not affect this bit.

0 = IDLE (there is currently no activity for this interrupt) (default)

1 = Send pending (The interrupt has been injected but its delivery is temporarily held up due to the APIC bus being busy or the inability of the receiving APIC unit to accept that interrupt at that time.)

**Bit 11** **Destination Mode (RW)**—This field determines the interpretation of the Destination field. When DESTMOD=0 (physical mode), a destination APIC is identified by its ID.
0 = Physical mode IOREDTBLx[27:24] = APIC ID
1 = Logical mode IOREDTBLx[31:24] = Set of processors

**Bits 10–8** **Delivery Mode (RW)**—The delivery mode is a 3-bit field that specifies how the APICs listed in the destination field should act upon reception of this signal. Note that certain delivery modes only operate as intended when used in conjunction with a specific trigger mode.

**Table 83. Delivery mode restrictions**

| Bits [10:8] | Mode | Description |
|---|---|---|
| 000 | Fixed Priority | Deliver the signal on the INTR signal of all processor cores listed in the destination. Trigger mode for fixed-priority delivery mode can be edge or level. |
| 001 | Lowest Priority | Deliver the signal on the INTR signal of the processor core that is executing at the lowest priority among all the processors listed in the specified destination. Trigger mode for lowest priority. Delivery mode can be edge or level. |
| 010 | SMI | A delivery mode equal to SMI requires an edge-trigger mode. The vector information is ignored but must be programmed to all zeroes for future compatibility. |
| 011 | Reserved | |
| 100 | NMI | Deliver the signal on the NMI signal of all processor cores listed in the destination. Vector information is ignored. NMI is treated as an edge-triggered interrupt, even if it is programmed as a level-triggered interrupt. For proper operation, this redirection table entry must be programmed to edge-triggered interrupt. |
| 101 | INIT | Deliver the signal to all processor cores listed in the destination by asserting the INIT signal. All addressed local APICs will assume their INIT state. INIT is always treated as an edge-triggered interrupt, even if programmed otherwise. For proper operation, this redirection table entry must be programmed to edge-triggered interrupt. |
| 110 | Reserved | |
| **111** | ExtINT | Deliver the signal to the INTR signal of all processor cores listed in the destination as an interrupt that originated in an externally connected (8259A-compatible) interrupt controller. The INTA cycle that corresponds to this ExtINT delivery is routed to the external controller that is expected to supply the vector. A delivery mode of ExtINT requires an edge-trigger mode. |

**Bits 7–0** **Interrupt Vector (RW)**—The vector field is an 8-bit field containing the interrupt vector for this interrupt. Vector values range from 10h to FEh.

# 8 Electrical Data

## 8.1 Absolute Ratings

Long-term reliability and functional integrity of the AMD-756 peripheral bus controller will be adversely affected if it is subjected to conditions exceeding the absolute ratings listed in Table 84.

**Table 84. Absolute Ratings**

| Parameter | Minimum | Maximum | Comments |
|---|---|---|---|
| $T_{CASE}$ (under bias) | -65°C | +110°C | |
| $T_{STORAGE}$ | –65°C | +150°C | |
| $V_{DD3}$ | –0.5 V | 4.0 V | |
| $V_{PIN}$ | –0.5 V | $V_{DD3}$+0.5V and < 4.0V | |

**Warning**: Stress above the parameters listed can cause permanent damage to the device. Functional operation of this device should be restricted to the described conditions.

## 8.2 Operating Ranges

The functional operation of the AMD-756 peripheral bus controller should be within the limits of the voltage and temperature parameters defined in Table 85.

**Table 85. Operating Ranges**

| Parameter | Minimum | Typical | Maximum | Comments |
|---|---|---|---|---|
| $V_{DD3}$ | 3.0 V | 3.3 V | 3.6 V | (note 1) |
| $T_{CASE}$ | 0°C | | 70°C | |

***Note:***
1. *$V_{DD3}$ is referenced from $V_{SS}$*

# 8.3 DC Characteristics

The DC characteristics of the AMD-756 peripheral bus controller are listed in the following tables.

**Table 86. USB DC Characteristics**

| Symbol | Parameter Description | Preliminary Data | | Comments |
|---|---|---|---|---|
| | | Min | Max | |
| $V_{BUS}$ | Supply Voltage for Powered Host Port | 4.75 V | 5.25 V | |
| $I_{BUS}$ | Supply Curretn for Powered Host Port | 500 mA | | |
| $V_{OL}$ | Output Low Voltage | | 0.3 V | $I_{OL}$ = 2.0 mA load |
| $V_{OH}$ | Output High Voltage | 2.8 V | 3.6 V | $I_{OH}$ = -0.25 mA load |
| $V_{DI}$ | Differential Input Sensitivity | 0.2 V | | |
| $V_{SE}$ | Single ended signal "0" | 0.6 V | 1.6 V | |
| $I_{LO}$ | Output Leakage Current | −10 µA | ±10 µA | $0 \leq V_{IN} \leq V_{DD3.}$ |
| $C_{IN}$ | Transceiver Capacitance | | 30 pF | |
| ***Notes:*** | | | | |

**Table 87. IDE DC Charateristics**

| Symbol | Parameter Description | Preliminary Data | | Comments |
|---|---|---|---|---|
| | | Min | Max | |
| $V_{IL}$ | Input Low Voltage | | 0.8 V | |
| $V_{IH}$ | Input High Voltage | 2.0 V | | |
| $V_{OL}$ | Output Low Voltage | | 0.5 V | $I_{OL}$ = 4mA load |
| $V_{OH}$ | Output High Voltage | 2.4 V | | $I_{OH}$ = .4 mA load |
| $I_{OL}$ | Driver Sink Current | 4 mA | | |
| $I_{OH}$ | Driver Source Current | 400 µA | | |
| $C_{IN}$ | Input Capacitance | | 25 pF | |
| $C_{OUT}$ | Output Capacitance | | 25 pF | |

***Notes:***

**Table 88. IDE Driver Types and Required Pull-ups**

| Signal | Source | Driver type (see note 1) | Pull-up at host (see note 2) | Pull-up at each device (see note 2) | Notes |
|---|---|---|---|---|---|
| Reset | Host | TP | | | |
| DD (15:0) | Bidir | TS | | | 3 |
| DMARQ | Device | TS | 5,6 kΩ PD | | 4 |
| DIOR- DIOW- | Host | TS | | | |
| IORDY | Device | TS | 1,0 kΩ | | 5 |
| CSEL | Host | | Ground | 10 kΩ | 6 |
| DMACK- | Host | TP | | | |
| INTRQ | Device | TS | | | |
| DA (2:0) | Host | TP | | | |
| PDIAG- | Device | TS | | 10 kΩ | |
| CS0- CS1- | Host | TP | | | |
| DASP- | Device | OC | | 10 kΩ | |

*Notes:*

*1 TS=Tri-state; OC=Open Collector; TP=Totem-pole; PU=Pull-up; PD=Pull-down; VS=Vendor specific*

*2 All resistor values are minimum (lowest) allowed.*

*3 Devices shall not have a pull-up resistor on DD7. It is recommended that a host have a 10 kΩ pull-down resistor and not a pull-up resistor on DD7 to allow a host to recognize the absence of a device at power-up. It is intended that this recommendation become mandatory in a future revision of this standard.*

*4 ATA-3 defines this line to be tri-stated whenever the device is not selected or is not executing a DMA data transfer. When enabled by DMA transfer, it shall be driven high and low by the device.*

*5 This signal should only be enabled during DIOR/DIOW cycles to the selected device.*

*6 When used as CSEL, this line is grounded at the Host and a 10 kΩ pull-up is required at both devices.*

**Table 89. ISA Bus DC Charateristics**

| Symbol | Parameter Description | Preliminary Data | | Comments |
|---|---|---|---|---|
| | | Min | Max | |
| $V_{IL}$ | Input Low Voltage | –0.50 V | 0.8 V | |
| $V_{IH}$ | Input High Voltage | 2.0 V | $V_{DD3}$+0.5 V | note 1 |
| $V_{OL}$ | Output Low Voltage | | 0.5 V | $I_{OL}$ = 4.0-mA load |
| $V_{OH}$ | Output High Voltage | 2.4 V | | $I_{OH}$ = 1.0-mA load |
| $I_{LI}$ | Input Leakage Current | | .4 mA | |
| $I_{LO}$ | Output Leakage Current | | 12 mA | |
| $C_{OUT}$ | I/O Capacitance (per slot) | | 20 pF | |

***Notes:***
1. *$V_{DD3}$ refers to the voltage being applied to $V_{DD3}$ during functional operation.*

**Table 90. PCI Bus DC Charateristics**

| Symbol | Parameter Description | Preliminary Data | | Comments |
|---|---|---|---|---|
| | | Min | Max | |
| $V_{IL}$ | Input Low Voltage | –0.5 V | $0.3V_{DD3}$ V | |
| $V_{IH}$ | Input High Voltage | $0.5V_{DD3}$ V | $V_{DD3}$+0.5 V | note 1 |
| $V_{OL}$ | Output Low Voltage<br>AD[31:]. C/BE#[3:0], and PREQ#<br>FRAME#, IRDY#, TRDY#, DEVEL#, STOP#, | 2.0V | $0.1V_{DD3}$ V | $I_{OL}$ = 1.5 mA load |
| $V_{OH}$ | Output High Voltage | $0.9V_{DD3}$ V | | $I_{OH}$ = -0.5 mA load |
| $I_{LI}$ | Input Leakage Current | | ±10 µA | $0 < V_{in} < V_{DD3}$ |
| $C_{IN}$ | Input Capacitance | | 10 pF | |
| $C_{CLK}$ | PCLK Pin Capacitance | 5 pF | 12 pF | |
| $C_{OUT}$ | I/O Capacitance | | 20 pF | |
| $C_{CLK}$ | CLK Capacitance | | 10 pF | |
| $C_{TIN}$ | Test Input Capacitance (TDI, TMS, TRST) | | 10 pF | |
| $C_{TOUT}$ | Test Output Capacitance (TDO) | | 15 pF | |
| $C_{TCK}$ | TCK Capacitance | | 10 pF | |

***Notes:***
1. *$V_{DD3}$ refers to the voltage being applied to $V_{DD3}$ during functional operation.*

# 8.4 Power Dissipation

Table 91 shows typical and maximum power dissipation of the AMD-756 peripheral bus controller during normal and reduced power states. The measurements are taken with PCLK = 33 MHz, $V_{DD3}$ = 3.3V.

**Table 91. Typical and Maximum Power Dissipation**

| Clock Control State | **Typical** (Note 1) | **Maximum** (Note 2) | Comments |
|---|---|---|---|
| Normal (Thermal Power) | TBD W | TBD W | Note 3 |

***Notes:***

1. *Typical power is measured during instruction sequences or functions associated with normal system operation.*
2. *Maximum power is determined for the worst-case instruction sequence or function for the listed clock control states.*
3. *The maximum power dissipated in the normal clock control state must be taken into account when designing a solution for thermal dissipation for the AMD-756 peripheral bus controller processor.*

# 9 Switching Characteristics

This section summarizes the AMD-756 peripheral bus controller signal switching characteristics. Valid delay, float, setup, and hold timing specifications are listed.

The setup and hold time requirements for the AMD-756 peripheral bus controller input signals presented here must be met by any device that interfaces with it to assure the proper operation of the AMD-756 peripheral bus controller.

All signal timings are based on the following assumptions:

- The target signals are input or output signals that are switching from logical 0 to 1, or from logical 1 to 0.
- Measurements are taken from the time the reference signal (HCLK, PCLK, or RESET) passes through 1.5 V to the time the target signal passes through 1.5 V.
- All signal slew rates are 1 V/ns, from 0 V to 3 V (rising) or 3 V to 0 V (falling).
- Parameters are within the operating range listed in Table 84 on page 293.
- The load capacitance ($C_L$) on each signal is 0 pF.
- Valid delay and float timings for output signals during functional operation are relative to the rising edge of the given clock.

# 9.1 OSC Switching Characteristics

**Table 92. OSC Switching Characteristics for 14.3182-MHz Bus**

| Symbol | Parameter Description | Preliminary Data | | Figure | Comments |
|---|---|---|---|---|---|
| | | Min | Max | | |
| | Frequency | | 14.3182 MHz | | |
| $t_1$ | Clock period | 67 ns | 70 ns | 37 | |
| $t_2$ | Clock high time | 20 ns | | 37 | |
| $t_3$ | Clock low time | 20 ns | | 37 | |

***Note:***
*Jitter frequency power spectrum peaking must occur at frequencies greater than (CCLK frequency)/3 or less than 500 KHz.*

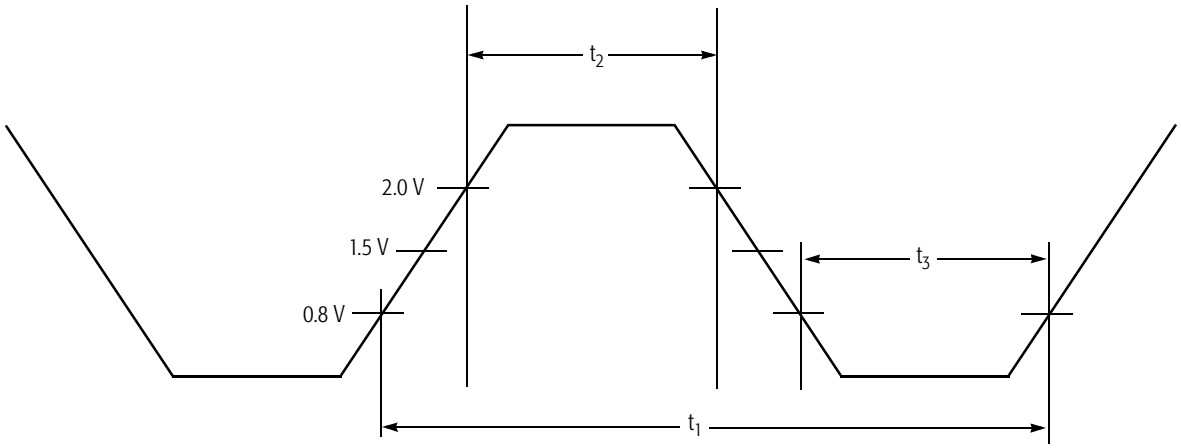


**Figure 37. OSC Waveform**

# 9.2 PCI Interface Timing

**Table 93. PCLK Switching Characteristics for 33-MHz PCI Bus**

| Symbol | Parameter Description | Preliminary Data | | Figure | Comments |
|---|---|---|---|---|---|
| | | Min | Max | | |
| $t_1$ | PCLK cycle time | 30 ns | ∞ | 38 | |
| $t_2$ | PCLK high time | 11.0 ns | | 38 | |
| $t_3$ | PCLK low time | 11.0 ns | | 38 | |
| $t_4$ | PCLK fall time | 1 V/ns | 4V/ns | 38 | |
| $t_5$ | PCLK rise time | 1 V/ns | 4V/ns | 38 | |
| | PCLK period stability | | ± 250 ps | | See note |

***Note:***
*Jitter frequency power spectrum peaking must occur at frequencies greater than (CCLK frequency)/3 or less than 500 KHz.*

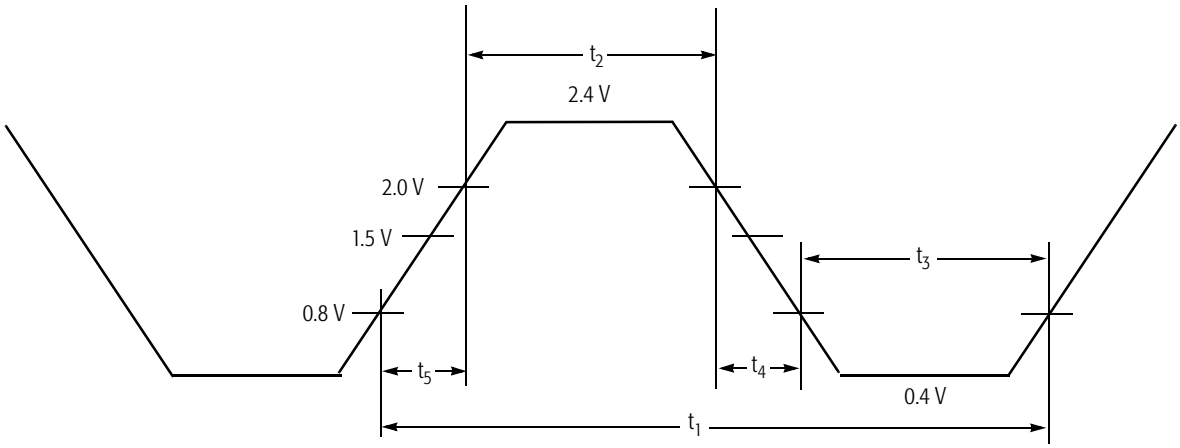


**Figure 38. PCLK Waveform**

**Table 94. PCI Interface Timing**

| Symbol | Parameter Description | Preliminary Data | | Figure | Comments |
|---|---|---|---|---|---|
| | | Min | Max | | |
| $t_{su}$ | AD[31:0] setup time | 7 ns | | 39 | |
| | PGNT# setup time | 12 ns | | 39 | |
| | FRAME#, STOP#, TRDY#, DEVSEL#, IRDY#, C/BE[3:0]# setup time | 7 ns | | 39 | |
| $t_h$ | AD[31:0], FRAME#, STOP#, TRDY#, DEVSEL#, IRDY#, C/BE[3:0]#, PGNT# hold time | 1 ns | | 39 | |
| $t_{vd}$ | AD[31:0], C/BE[3:0]# valid delay | 2 ns | 15 ns | 39 | |
| | FRAME#, STOP#, TRDY#, DEVSEL#, IRDY# valid delay | 2 ns | 11 ns | 39 | |
| | PREQ# valid delay | 2 ns | 12 ns | 39 | |
| $t_{fd}$ | FRAME, #STOP#, TRDY#, DEVSEL#, IRDY#, C/BE[3:0]# float delay | | 28 ns | | |
| $t_{lat}$ | PREQ# to PGNT# latency | 2 clks | | | |

***Note:***
*Measurements are taken with a 50pF load, unless otherwise noted.*

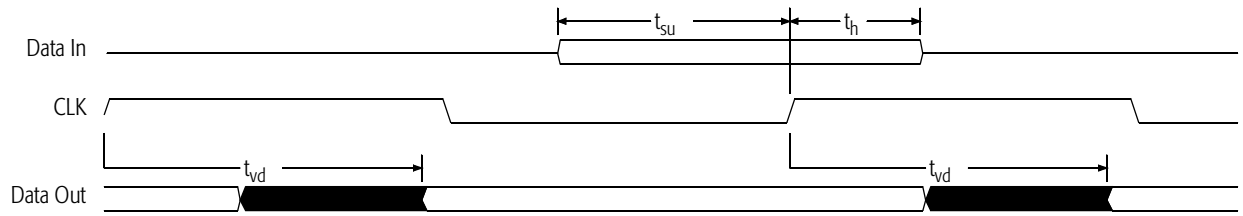


**Figure 39. Setup, Hold, and Valid Delay Timing Diagram**

# 9.3 USB Interface Timing

**Table 95. USBCLK Switching Characteristics for USB Bus**

| Symbol | Parameter Description | Preliminary Data | | Figure | Comments |
|---|---|---|---|---|---|
| | | Min | Max | | |
| | Frequency | 48 MHz | 48 MHz | | |
| $t_1$ | USBCLK cycle time | 20.8 ns | 20.8 ns | 40 | |
| $t_2$ | USBCLK high time | 9.4 ns | | 40 | |
| $t_3$ | USBCLK low time | 9.4 ns | | 40 | |
| | 12 MHz USBDATA Frequency | 11.99 Mb/s | 12.01 Mb/s | | |
| | 1.5 MHz USBDATA Frequency | 1.48 Mb/s | 1.52 Mb/s | | |
| $t_4$ | 12 MHz USBDATA Data Transition fall time | 4 ns | 20 ns | 41 | |
| $t_5$ | 12 MHz USBDATA Data Transition rise time | 4 ns | 20 ns | 41 | |
| $t_4$ | 1.5 MHz USBDATA Data Transition fall time | 75 ns | 300 ns | 41 | |
| $t_5$ | 1.5 MHz USBDATA Data Transition rise time | 75 ns | 300 ns | 41 | |
| $t_6$ | Source differential skew | | 5 ns | 41 | |
| $t_6$ | Receiver differential skew | | 10 ns | 41 | |
| | Driver jitter | | 3 ns | | |
| | Receiver jitter | | 25 ns | | |
| | Single-ended driver skew | | 10 ns | | |

***Note:***
*Jitter frequency power spectrum peaking must occur at frequencies greater than (CCLK frequency)/3 or less than 500 KHz.*

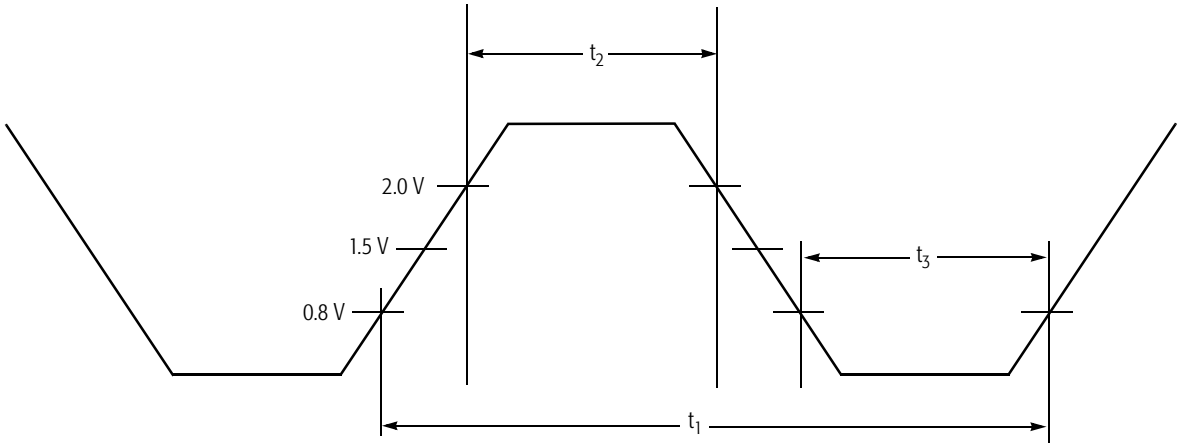


**Figure 40. USBCLK Waveform**

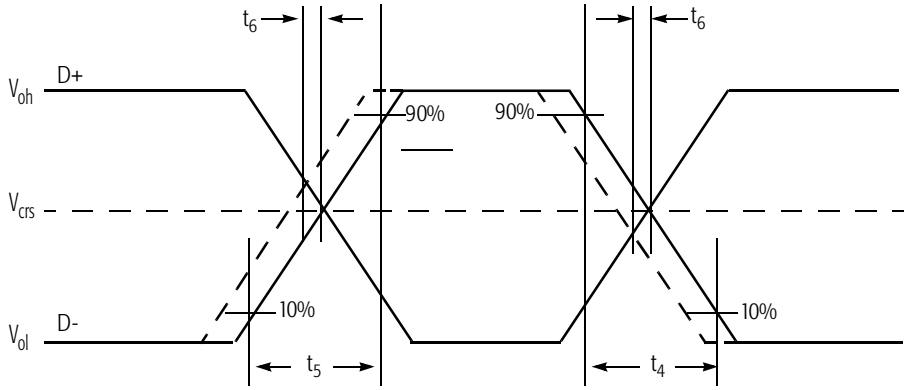


**Figure 41. USB DATA Waveform**

# 9.4 ISA Interface Timing

**Table 96. BCLK Switching Characteristics for 8-MHz Bus**

| Symbol | Parameter Description | Preliminary Data | | Figure | Comments |
|---|---|---|---|---|---|
| | | Min | Max | | |
| $t_1$ | Clock period | 125 ns | | 42 | |
| $t_2$ | Clock high time | 49 ns | | 42 | |
| $t_3$ | Clock low time | 49 ns | | 42 | |
| $t_4$ | Clock rise time | | 4 ns | 42 | |
| $t_5$ | Clock fall time | | 4 ns | 42 | |
| ***Note:*** | | | | | |

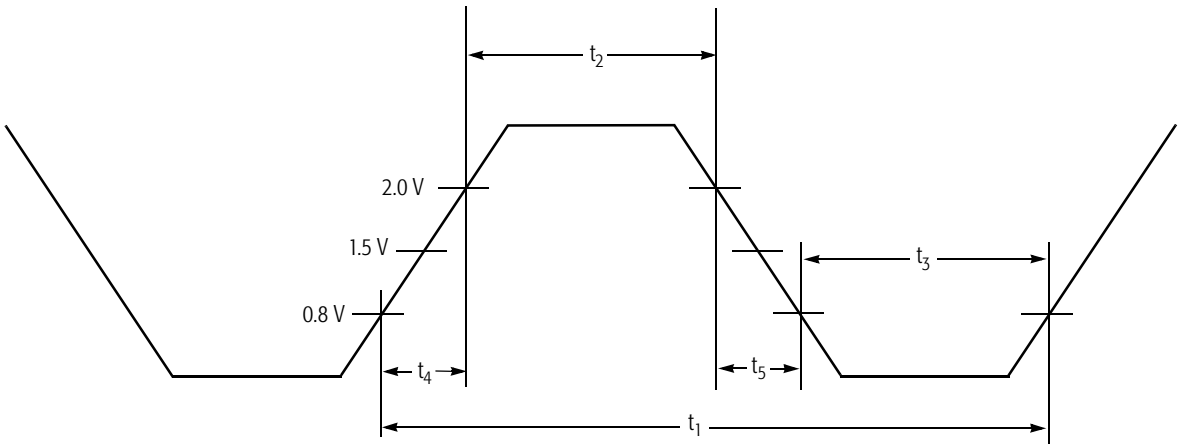


**Figure 42. BCLK Waveform**

**Table 97. ISA Master Interface Timing**

| Symbol | Parameter Description | Preliminary Data | | Figure | Comments |
|---|---|---|---|---|---|
| | | Min | Max | | |
| $t_{su1}$ | LA[23:20] setup to BALE | 150 ns | | 43 | |
| $t_{su2}$ | LA[23:20] setup to MEMx# | 173 ns | | 43 | |
| $t_{su3}$ | SA[19:0] setup to BALE | 37 ns | | 43 | |
| $t_{su4}$ | SA[19:0] setup to MEMx# | 34 ns | | 43 | |
| $t_{su5}$ | SD[15:0] setup to MEMR# | 24 ns | | 43 | |
| $t_{su6}$ | SD[15:0] setup to MEMW# | –40 ns | | 43 | |
| $t_{h1}$ | LA[23:17] hold from BALE | 26 ns | | 43 | |
| $t_{h2}$ | MEMCS16# hold from LA[23:20] | 0 ns | | 43 | |
| $t_{h3}$ | SA[19:0] hold from MEMx# | 41 ns | | 43 | |
| $t_{h4}$ | SD[15:0] hold from MEMR# | 0 ns | | 43 | |
| $t_{h5}$ | SD[15:0] hold from MEMW# | 45 ns | | 43 | |
| $t_{vd1}$ | LA[23:17] to MEMCS16# valid delay | | 94 ns | 43 | |
| $t_{vd2}$ | MEMR# to SD[15:0] valid delay | | 150 ns | 43 | |
| $t_{vd3}$ | MEMx# to BALE valid delay | 44 ns | | 43 | |
| $t_{vd4}$ | MEMx# to IOCHRDY valid delay | | 78 ns | 43 | |
| $t_{vd5}$ | MEMx# to SMEMx | | 16 ns | 43 | |
| $t_{vd6}$ | SA[19:0], SBHE# to MEMCS16# valid delay | | 35 ns | 43 | |
| $t_{pw1}$ | BALE pulse width (High) | 50 ns | | 43 | |
| $t_{pw2}$ | IOCHRDY inactive pulse width | 120 ns | | 43 | |
| $t_{pw3}$ | MEMx# active pulse width (Low) | 225 ns | | 43 | |
| $t_{pw4}$ | MEMx# inactive pulse width (High) | 163 ns | | 43 | |
| $t_{fd1}$ | MEMR# to SD[15:0] float delay | | 41 ns | 43 | |
| $t_{fd2}$ | MEMW# to SD[15:0] float delay | | 105 ns | 43 | |

***Note:***
*Measurements are taken with no load.*

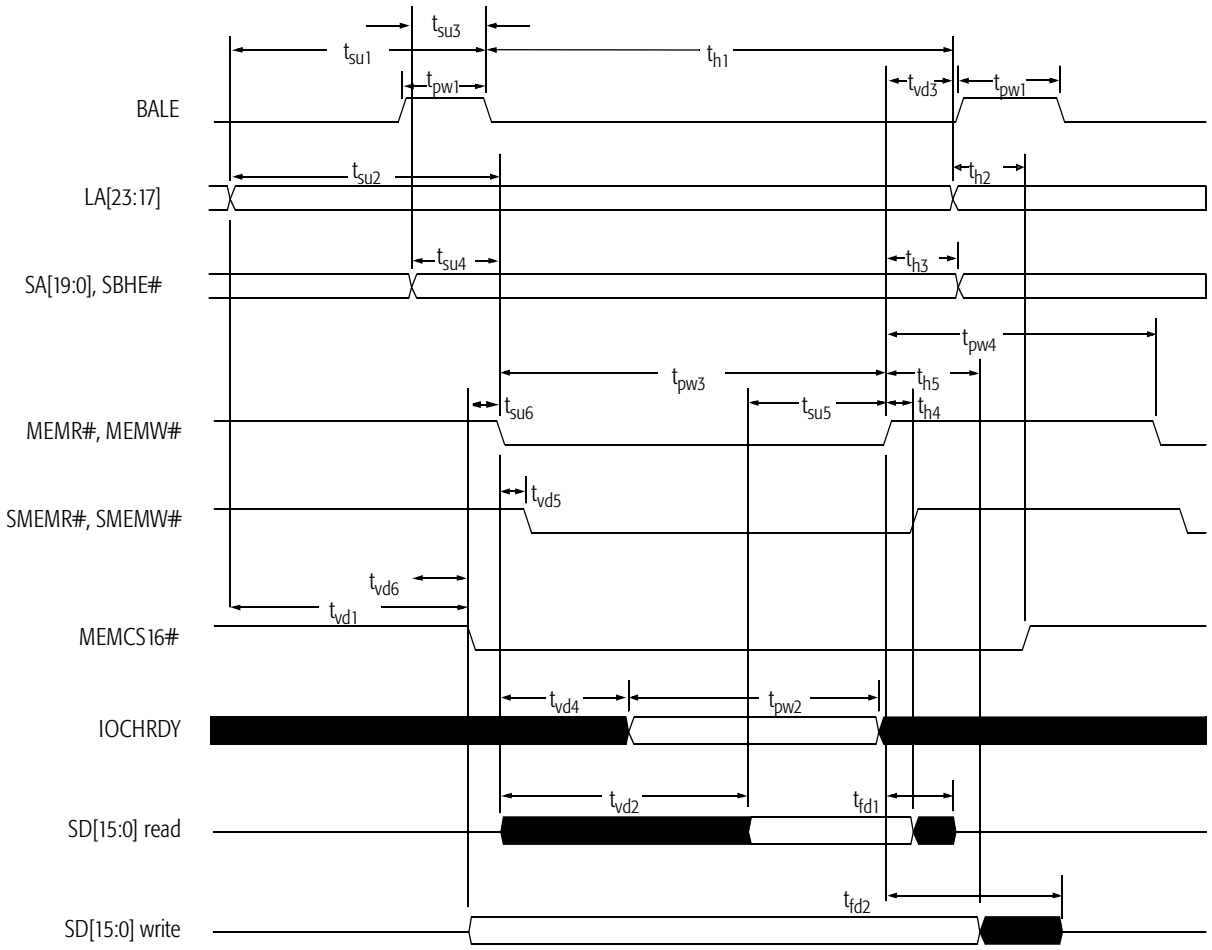


**Figure 43. ISA Master Interface Timing**

**Table 98. ISA 8-Bit Slave Interface Timing**

| Symbol | Parameter Description | Preliminary Data | | Figure | Comments |
|---|---|---|---|---|---|
| | | Min | Max | | |
| $t_{su1}$ | AEN setup to BALE | 111 ns | | 44 | |
| $t_{su2}$ | AEN setup to IOx# | 111 ns | | 44 | |
| $t_{su3}$ | SA[19:0] setup to BALE | 37 ns | | 44 | |
| $t_{su4}$ | SA[19:0] setup to IOx# | 100 ns | | 44 | |
| $t_{su5}$ | SD[15:0] setup to IOR# | 24 ns | | 44 | |
| $t_{su6}$ | SD[15:0] setup to IOW# | –40 ns | | 44 | |
| $t_{h1}$ | AEN hold to IOx# | 41 ns | | 44 | |
| $t_{h2}$ | SA[19:0] hold to IOx# | 41 ns | | 44 | |
| $t_{h3}$ | SD[15:0] hold to IOR# | 0 ns | | 44 | |
| $t_{h4}$ | SD[15:0] hold to IOW# | 45 ns | | 44 | |
| $t_{vd1}$ | IOR# to SD[15:0] valid delay | | 500 ns | 44 | |
| $t_{vd2}$ | IOx# to BALE valid delay | 44 ns | | 44 | |
| $t_{vd3}$ | IOx# to IOCHRD valid delay | | 366 ns | 44 | |
| $t_{vd4}$ | SA[19:0] to IOCS16# valid delay | | 91 ns | 44 | |
| $t_{pw1}$ | BALE pulse width | 50 ns | | 44 | |
| $t_{pw2}$ | IOCHRDY inactive pulse width | 120 ns | | 44 | |
| $t_{pw3}$ | IOx# active pulse width | 160 ns | | 44 | |
| $t_{pw4}$ | IOx# inactive pulse width | 163 ns | | 44 | |
| $t_{fd1}$ | IOR# to SD[15:0] float delay | | 41 ns | 44 | |
| $t_{fd2}$ | IOW# to SD[15:0] float delay | | 105 ns | 44 | |

***Note:***
*Measurements are taken with no load.*

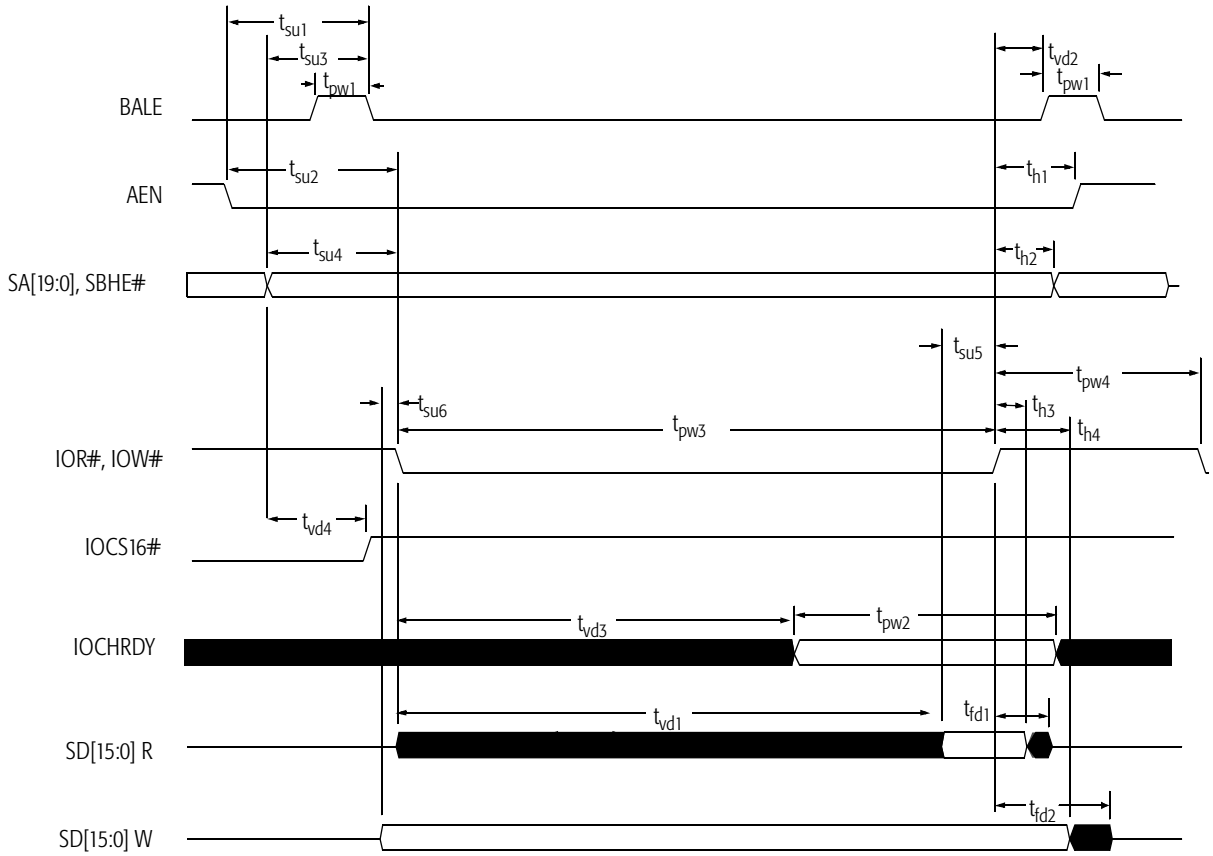


**Figure 44. ISA 8-Bit Slave Interface Timing**

**Table 99. ISA 16-Bit Slave Interface Timing**

| Symbol | Parameter Description | Preliminary Data | | Figure | Comments |
|---|---|---|---|---|---|
| | | Min | Max | | |
| $t_{su1}$ | AEN setup to BALE | 150 ns | | 45 | |
| $t_{su2}$ | AEN setup to IOx# | 150 ns | | 45 | |
| $t_{su3}$ | SA[19:0] setup to IOx# | 34 ns | | 45 | |
| $t_{su4}$ | SA[19:0] setup to BALE | 37 ns | | 45 | |
| $t_{su5}$ | SD[15:0] setup to IOR# | 24 ns | | 45 | |
| $t_{su6}$ | SD[15:0] setup to IOW# | –40 ns | | 45 | |
| $t_{h1}$ | AEN hold from IOx# | 26 ns | | 45 | |
| $t_{h2}$ | IOCS16# hold from SA[19:0] | 0 ns | | 45 | |
| $t_{h3}$ | SA[19:0] hold from IOx# | 41 ns | | 45 | |
| $t_{h4}$ | SD[15:0] hold from IOR# | 0 ns | | 45 | |
| $t_{h5}$ | SD[15:0] hold from IOW# | 45 ns | | 45 | |
| $t_{vd1}$ | IOx# to IOCHRDY valid delay | | 78 ns | 45 | |
| $t_{vd2}$ | IOx# to BALE valid delay | 44 ns | | 45 | |
| $t_{vd3}$ | IOx# to IOCS16# valid delay | | 16 ns | 45 | |
| $t_{vd4}$ | SA[19:0] to IOCS16# valid delay | | 35 ns | 45 | |
| $t_{vd5}$ | IOR# to SD[15:8] valid delay | 1.5 ns | 8.5 ns | 45 | |
| $t_{pw1}$ | BALE pulse width | 50 ns | | 45 | |
| $t_{pw2}$ | IOCHRDY inactive pulse width | 120 ns | | 45 | |
| $t_{pw3}$ | IOx# active pulse width | 160 ns | | 45 | |
| $t_{pw4}$ | IOx# inactive pulse width | 163 ns | | 45 | |
| $t_{fd1}$ | IOR# to SD[15:0] float delay | | 41 ns | 45 | |
| $t_{fd2}$ | IOW# to SD[15:0] float delay | | 105 ns | 45 | |

***Note:***
*Measurements are taken with no load.*

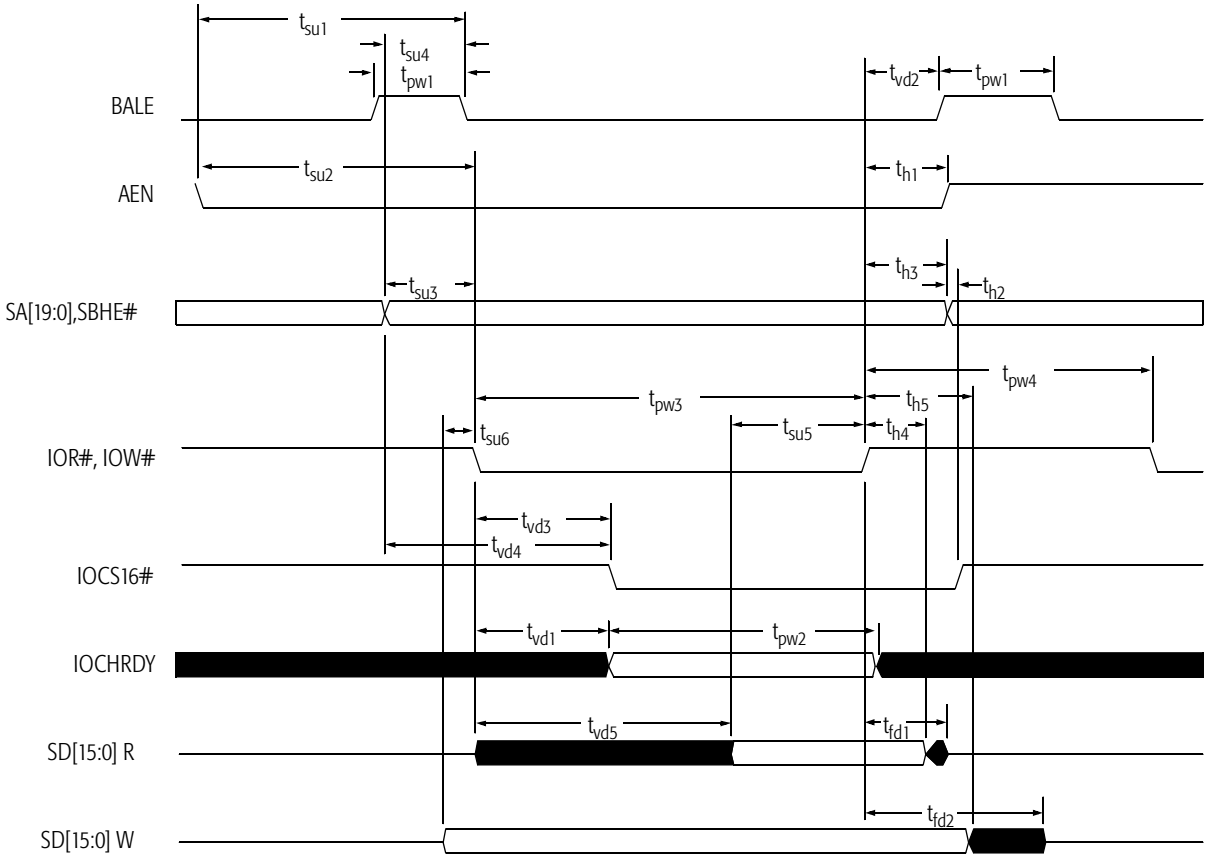


**Figure 45. ISA 16-Bit Slave Interface Timing**

**Table 100. ISA Master-to-PCI Access Timing**

| Symbol | Parameter Description | Preliminary Data | | Figure | Comments |
|---|---|---|---|---|---|
| | | Min | Max | | |
| $t_{su1}$ | LA[23:17] setup to MEMx# | 23 ns | | 46 | |
| $t_{su2}$ | SA[19:0] setup to MEMx# | 23 ns | | 46 | |
| $t_{su3}$ | SD[15:0] setup to MEMR# | 15 ns | | 46 | |
| $t_{su4}$ | SD[15:0] setup to MEMW# | –54 ns | | 46 | |
| $t_{h1}$ | LA[23:17] hold from MEMx# | 30 ns | | 46 | |
| $t_{h2}$ | SA[19:0] hold from MEMx# | 30 ns | | 46 | |
| $t_{h3}$ | SD[15:0] hold from MEMR# | 0 ns | | 46 | |
| $t_{h4}$ | SD[15:0] hold from MEMW# | 14 ns | | 46 | |
| $t_{vd1}$ | IOCHRDY to SD[15:0] valid delay | | 69 ns | 46 | |
| $t_{vd2}$ | LA[23:17] to MEMCS16# valid delay | | 31 ns | 46 | |
| $t_{vd3}$ | MEMx# to IOCHRDY valid delay | | 85 ns | 46 | |
| $t_{pw1}$ | IOCHRDY inactive pulse width | 120 ns | | 46 | |
| $t_{pw2}$ | MEMx# active pulse width | 214 ns | | 46 | |
| $t_{pw3}$ | MEMx# inactive pulse width | 92 ns | | 46 | |
| $t_{fd1}$ | MEMR# to SD[15:8] float delay | | 55 ns | 46 | |
| $t_{fd2}$ | MEMW# to SD[15:8] float delay | | 50 ns | 46 | |

***Note:***
*Measurements are taken with no load.*

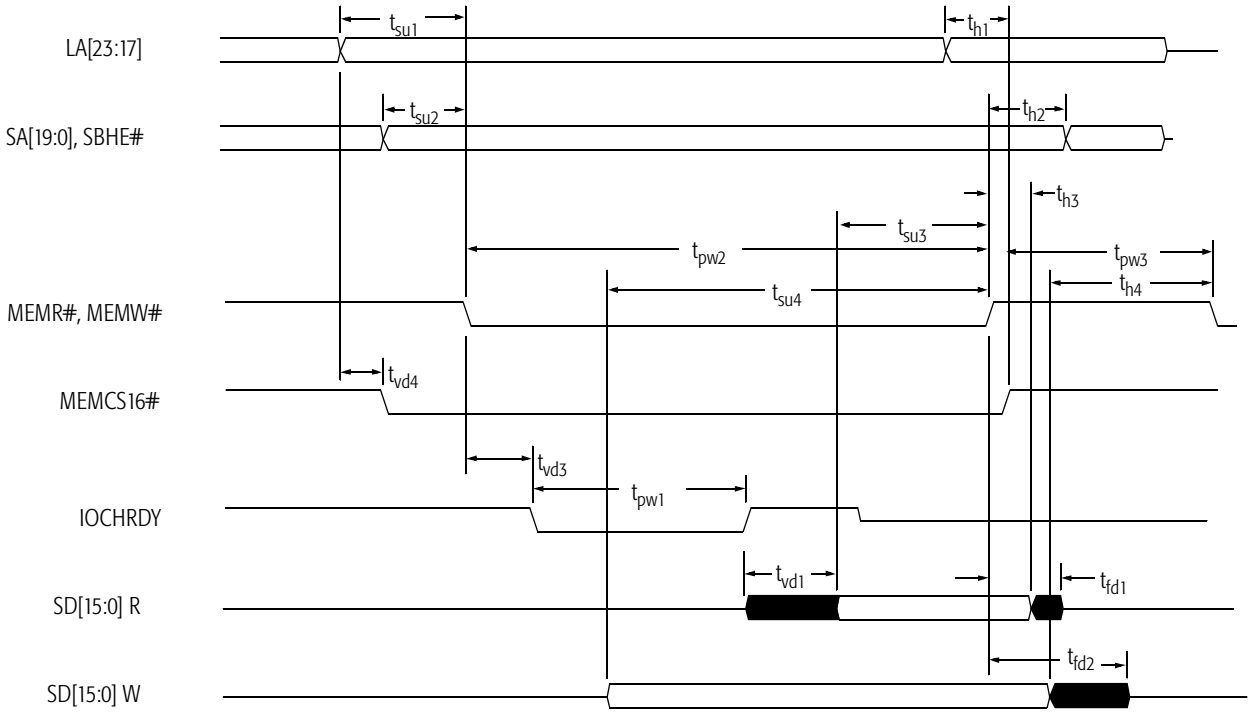


**Figure 46. ISA Master-to-PCI Access Timing**

**Table 101. Other ISA Master Timing**

| Symbol | Parameter Description | Preliminary Data | | Figure | Comments |
|---|---|---|---|---|---|
| | | Min | Max | | |
| $t_{vd1}$ | DREQ to DACK# valid delay | 240 ns | | 47 | |
| $t_{vd2}$ | DACK# to address, data, and control valid delay | 71 ns | | 47 | |
| $t_{fd1}$ | DACK# to address, data, and control float delay | 0 ns | | 47 | |

***Note:***
*Measurements are taken with no load.*

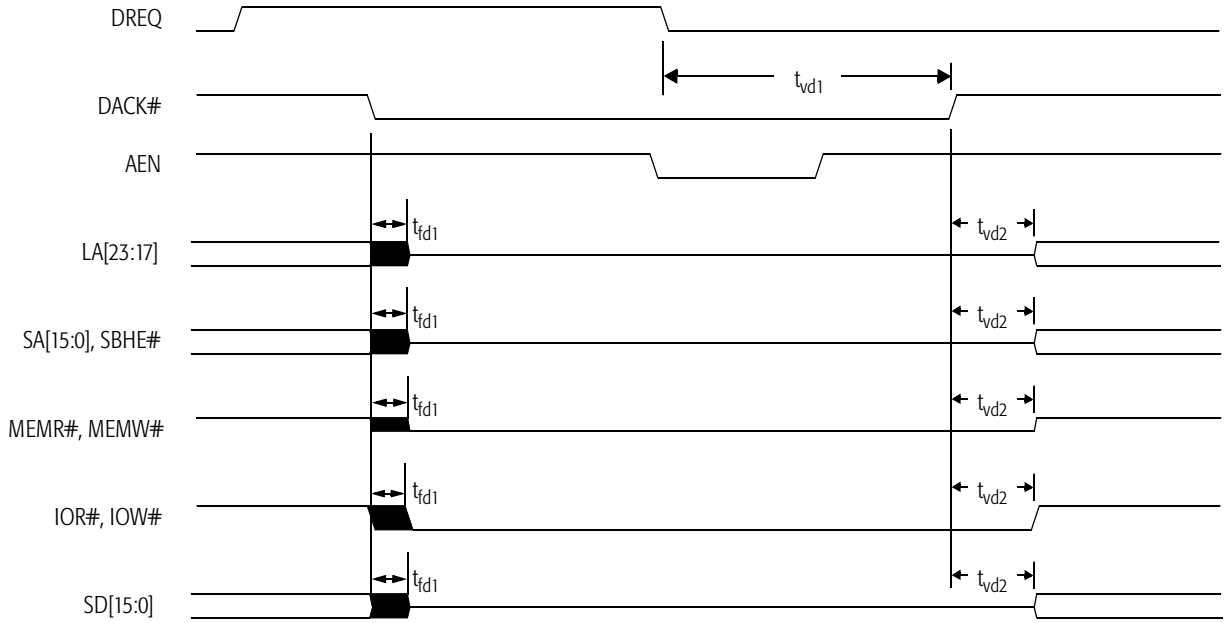


**Figure 47. Other ISA Master Timing**

# 9.5 DMA Interface Timing

**Table 102. DMA Read Cycle Timing**

| Symbol | Parameter Description | Preliminary Data | | Figure | Comments |
|---|---|---|---|---|---|
| | | Min | Max | | |
| $t_{su1}$ | AEN setup to IOW# | 111 ns | | 48 | |
| $t_{su2}$ | DACK# setup to IOW# | 312 ns | | 48 | |
| $t_{su3}$ | MEMR# setup to IOW# | –26 ns | | 48 | |
| $t_{su4}$ | SA[19:0],LA[23:20] setup to MEMR# | 99 ns | | 48 | |
| $t_{su5}$ | SD[15:0] setup to IOW# | 225 | | 48 | |
| $t_{su6}$ | TC setup to IOW# | 511 | | 48 | |
| $t_{h1}$ | AEN hold from IOW# | 41 ns | | 48 | |
| $t_{h2}$ | DACK# hold from IOW# | 155 ns | | 48 | |
| $t_{h3}$ | MEMR# hold from IOW# | 40 ns | | 48 | |
| $t_{h4}$ | MEMR# hold from IOCHRDY | 120 ns | | 48 | |
| $t_{h5}$ | SA[19:0], LA[23:20] hold from MEMR# | 51 ns | | 48 | |
| $t_{h6}$ | SD[15:0] hold from IOW# | 36 ns | | 48 | |
| $t_{h7}$ | TC hold from IOW# | 71 ns | | 48 | |
| $t_{vd1}$ | IOW# to DREQ inactive valid delay | | 315 ns | 48 | |
| $t_{vd2}$ | MEMR# to SMEMR# valid delay | | 15 ns | 48 | |
| $t_{vd3}$ | MEMR# to IOCHRDY valid delay | | 315 ns | 48 | |
| $t_{pw1}$ | IOCHRDY inactive pulse width | 125 ns | | 48 | |
| $t_{pw2}$ | IOW# active pulse width | 495 ns | | 48 | |
| $t_{pw3}$ | IOW# inactive pulse width | 465 ns | | 48 | |
| $t_{pw4}$ | MEMR# active pulse width | 495 ns | | 48 | |
| $t_{pw5}$ | MEMR# inactive pulse width | 465 ns | | 48 | |
| $t_{pw6}$ | TC active pulse width | 700 ns | | 48 | |

***Note:***
*Measurements are taken with no load.*

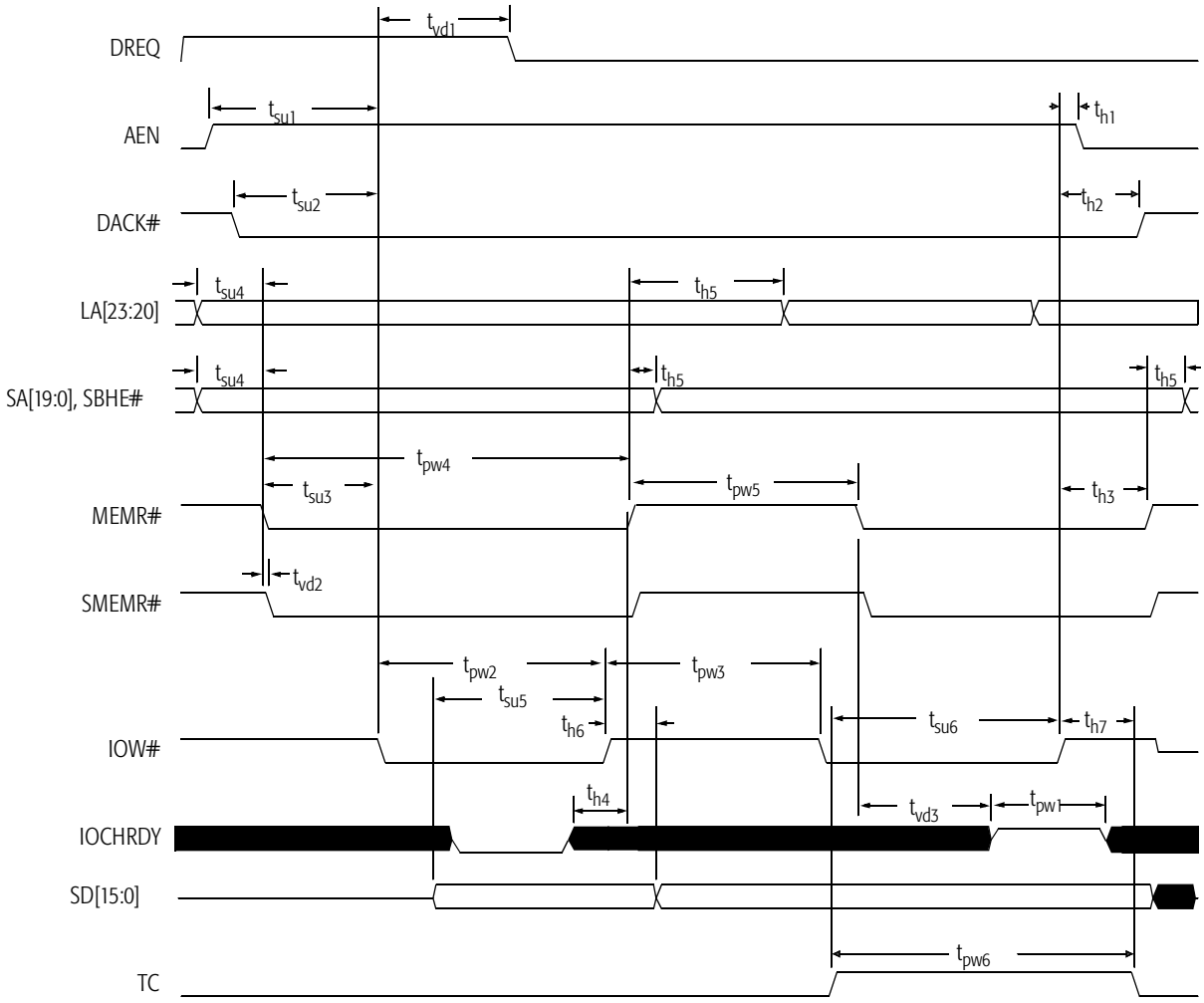


**Figure 48. DMA Read Cycle Timing**

**Table 103. DMA Write Cycle Timing**

| Symbol | Parameter Description | Preliminary Data | | Figure | Comments |
|---|---|---|---|---|---|
| | | Min | Max | | |
| $t_{su1}$ | AEN setup to IOR# | 111 ns | | 49 | |
| $t_{su2}$ | DACK# setup to IOR# | 73 ns | | 49 | |
| $t_{su3}$ | SA[19:0],LA[23:20] setup to MEMW# | 99 ns | | 49 | |
| $t_{su4}$ | TC setup to IOR# | 511 ns | | 49 | |
| $t_{h1}$ | AEN hold from IOR# | 41 ns | | 49 | |
| $t_{h2}$ | DACK# hold from IOR# | 100 ns | | 49 | |
| $t_{h3}$ | MEMW# hold from IOR# | 40 ns | | 49 | |
| $t_{h4}$ | SA[19:0], LA[23:20] hold from MEMW# | 51 ns | | 49 | |
| $t_{h5}$ | SD[15:0] hold from IOR# | 0 ns | | 49 | |
| $t_{h6}$ | TC hold from IOR# | 71 ns | | 49 | |
| $t_{vd1}$ | IOR# to DRQ valid delay | | 558 ns | 49 | |
| $t_{vd2}$ | IOR# to MEMW# valid delay | 230 ns | | 49 | |
| $t_{vd3}$ | IOR# to SD[15:0] valid delay | | 237 ns | 49 | |
| $t_{vd4}$ | MEMW# to IOCHRDY valid delay | | 315 ns | 49 | |
| $t_{vd5}$ | MEMW# to SMEMW# valid delay | | 15 ns | 49 | |
| $t_{pw1}$ | IOCHRDY active pulse width | 125 ns | | 49 | |
| $t_{pw2}$ | IOR# active pulse width | 760 ns | | 49 | |
| $t_{pw3}$ | IOR# inactive pulse width | 160 ns | | 49 | |
| $t_{pw4}$ | MEMW# active pulse width | 495 ns | | 49 | |
| $t_{pw5}$ | MEMW# inactive pulse width | 465 ns | | 49 | |
| $t_{pw6}$ | TC active pulse width | 700 ns | | 49 | |

***Note:***
*Measurements are taken with no load.*

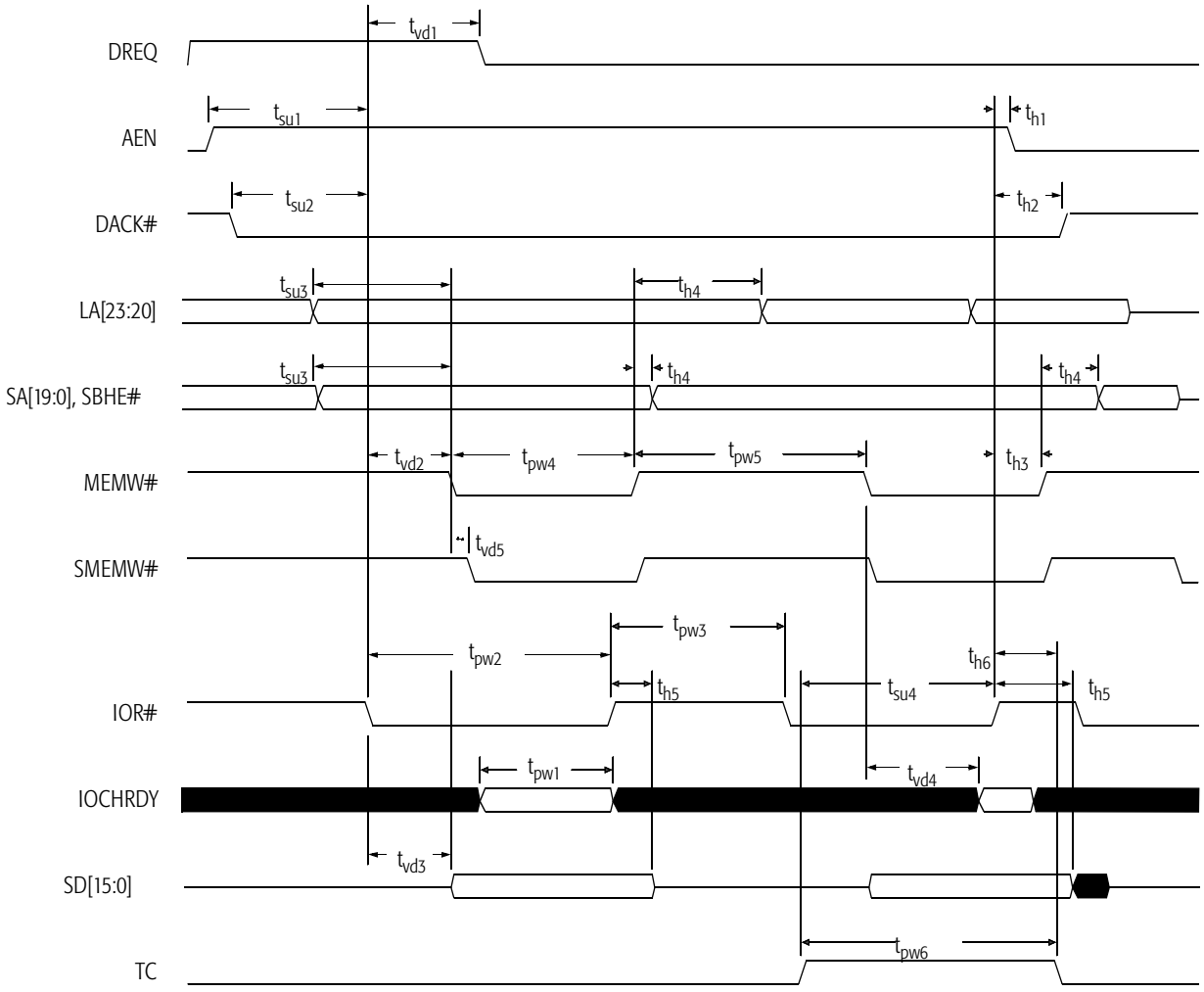


**Figure 49. DMA Write Cycle Timing**

**Table 104. Type F DMA Interface Timing**

| Symbol | Parameter Description | Preliminary Data | | Figure | Comments |
|---|---|---|---|---|---|
| | | Min | Max | | |
| $t_{su1}$ | SD[15:0] setup to IOW# | 70 ns | | 50 | |
| $t_{su2}$ | TC setup to final IOx# | 40 ns | | 50 | |
| $t_{h1}$ | DACK# hold from IOR# | 30 ns | | 50 | |
| $t_{h2}$ | DACK# hold from IOW# | 30 ns | | 50 | |
| $t_{h3}$ | DREQ hold from IOR# | 82 ns | | 50 | |
| $t_{h4}$ | DREQ hold from IOW# | 82 ns | | 50 | |
| $t_{h5}$ | SD[15:0] hold from IOR# | 2 ns | | 50 | |
| $t_{h6}$ | TC hold from IOW# | 0 ns | | 50 | |
| $t_{vd1}$ | AEN to IOR# valid delay | 111 ns | | 50 | |
| $t_{vd2}$ | AEN to IOW# valid delay | 111 ns | | 50 | |
| $t_{vd3}$ | DACK# to IOR# valid delay | 77 ns | | 50 | |
| $t_{vd4}$ | DACK# to IOW# valid delay | 77 ns | | 50 | |
| $t_{vd5}$ | IOR# to SD[15:0]valid delay | | 96 ns | 50 | |
| $t_{pw1}$ | IOR# active pulse width | 110 ns | | 50 | |
| $t_{pw2}$ | IOR# inactive pulse width | 115 ns | | 50 | |
| $t_{pw3}$ | IOW# active pulse width | 110 ns | | 50 | |
| $t_{pw4}$ | IOW# inactive pulse width | 115 ns | | 50 | |
| $t_{fd1}$ | IOR# to SD[15:8] float delay | | 61 ns | 50 | |

***Note:***
*Measurements are taken with no load.*

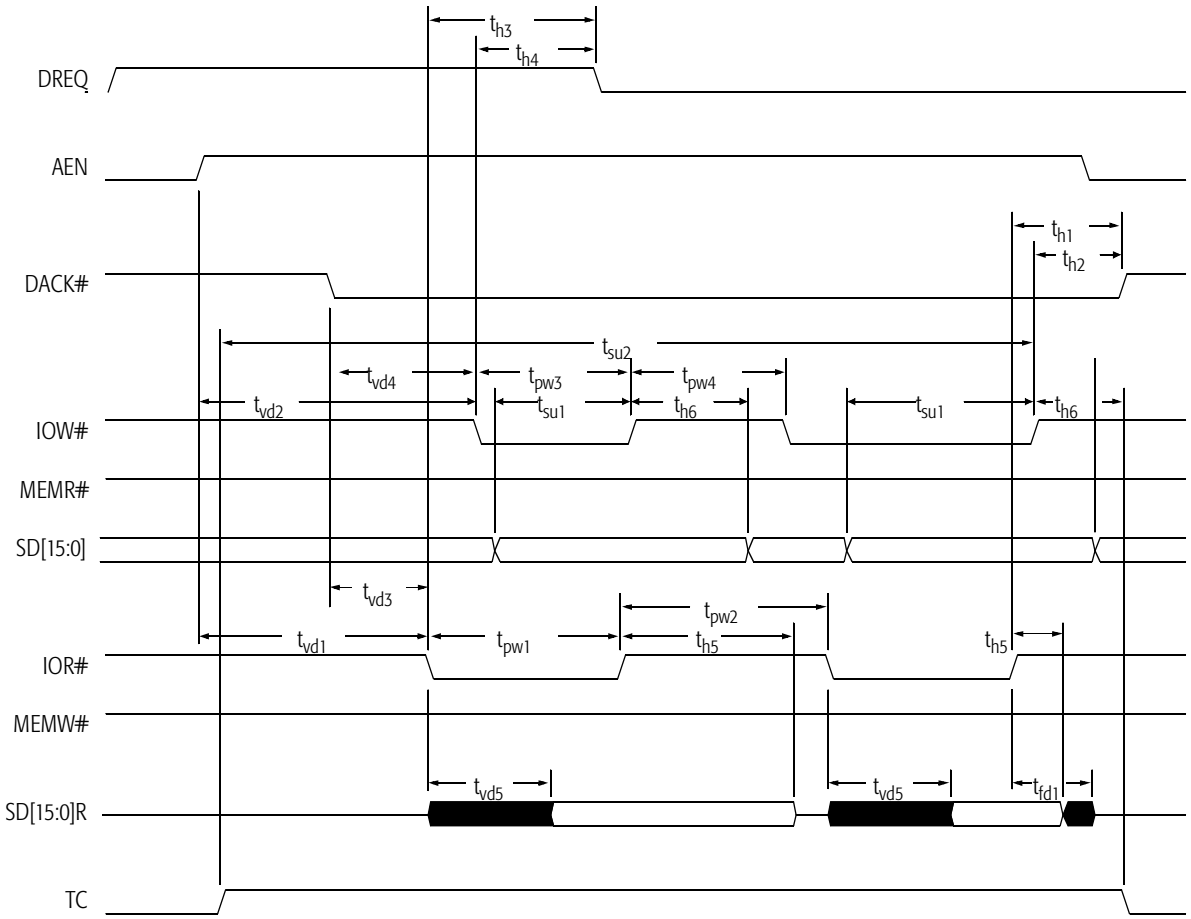


**Figure 50. Type F DMA Interface Timing**

# 9.6 EIDE Interface Timing

**Table 105. EIDE PIO**

| Symbol | Description | | Mode 0 | Mode 1 | Mode 2 | Mode 3 | Mode 4 |
|---|---|---|---|---|---|---|---|
| $t_{cyc1}$ | Cycle time (DIOW/R# to DIOW/R#) | min | 600 | 383 | 240 | 180 | 120 |
| $t_{rec1}$ | DIOx# recovery time | min | – | – | – | 70 | 25 |
| $t_{su1}$ | DA[2:0] setup to DIOx# | max | 70 | 50 | 30 | 30 | 25 |
| $t_{su2}$ | DDATA[15:0] read setup to DIOR# | min | 50 | 350 | 20 | 20 | 20 |
| $t_{su3}$ | DDATA[15:0] read setup to PCLK | min | 10 | 10 | 10 | 10 | 10 |
| $t_{su4}$ | DDATA[15:0] write setup to DIOW# | min | 60 | 45 | 30 | 30 | 20 |
| $t_{su5}$ | IORDY setup to PCLK | min | 20 | 20 | 20 | 20 | 20 |
| $t_{h1}$ | DDATA[15:0] read hold from DIOR# | min | 5 | 5 | 5 | 5 | 5 |
| $t_{h2}$ | DDATA[15:0] read hold from PCLK | min | 4 | 4 | 4 | 4 | 4 |
| $t_{h3}$ | DDATA[15:0] write hold from DIOW# | min | 30 | 20 | 15 | 10 | 10 |
| $t_{h4}$ | IORDY# hold from PCLK | min | 5 | 5 | 5 | 5 | 5 |
| $t_{vd1}$ | PCLK to DA[2:0], DCSx# valid delay | min | 2 | 2 | 2 | 2 | 2 |
| $t_{vd1}$ | PCLK to DA[2:0], DCSx# valid delay | max | 20 | 20 | 20 | 20 | 20 |
| $t_{vd2}$ | PCLK to DDATA[15:0] valid delay | min | 2 | 2 | 2 | 2 | 2 |
| $t_{vd2}$ | PCLK to DDATA[15:0] valid delay | max | 20 | 20 | 20 | 20 | 20 |
| $t_{vd3}$ | PCLK to SOE#, MASTER# valid delay | min | 2 | 2 | 2 | 2 | 2 |
| $t_{vd3}$ | PCLK to SOE#, MASTER# valid delay | max | 20 | 20 | 20 | 20 | 20 |
| $t_{pw1}$ | 8-bit DIOx# pulse width | min | 290 | 290 | 290 | 80 | 70 |
| $t_{pw1}$ | 16-bit DIOx# pulse width | min | 165 | 125 | 100 | 80 | 70 |

***Note:***
*All timings are in nanoseconds and reference figure* **51**.

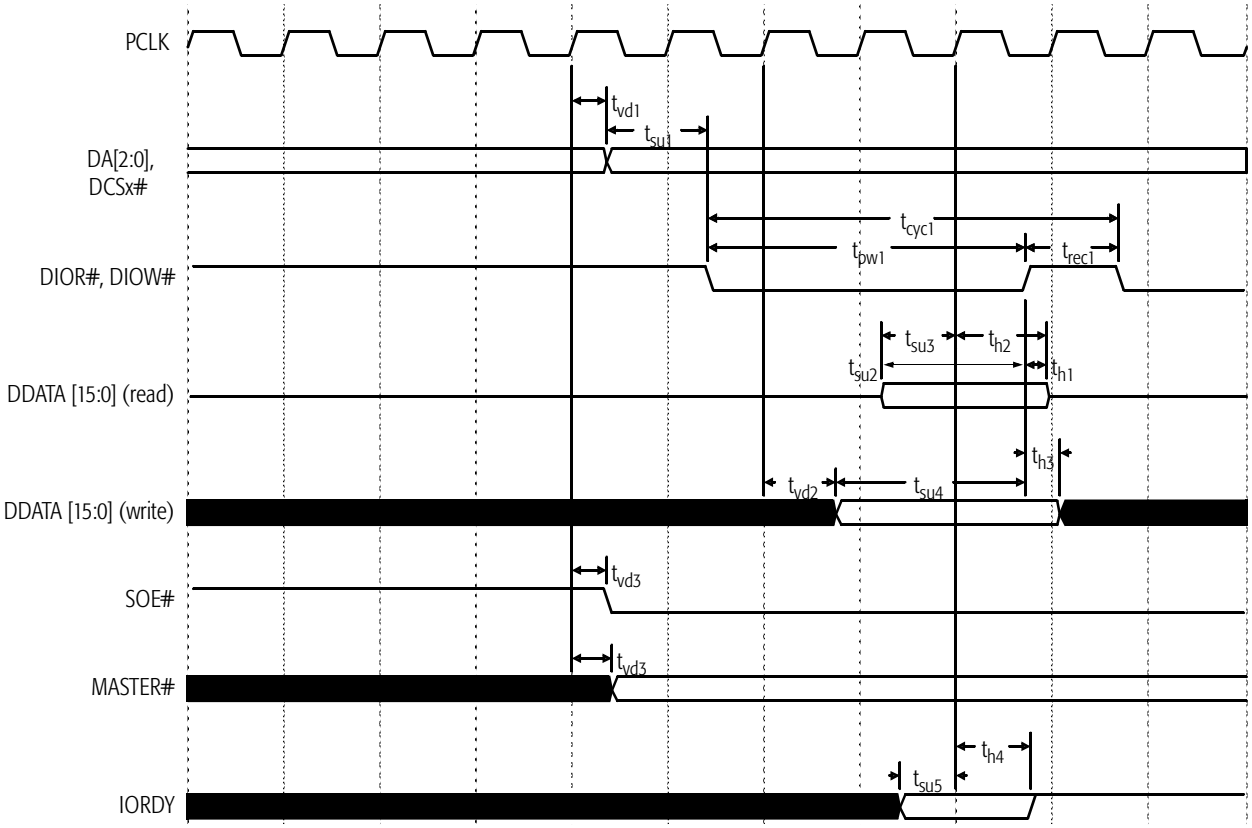


**Figure 51. EIDE PIO**

**Table 106. EIDE DMA**

| Symbol | Description | | Single-Word | | | Multi-Word | | |
|---|---|---|---|---|---|---|---|---|
| | | | Mode 0 | Mode 1 | Mode 2 | Mode 0 | Mode 1 | Mode 2 |
| $t_{cyc1}$ | Cycle time (DDACK# to DDACK#) | min | 960 | 480 | 240 | 480 | 150 | 120 |
| $t_{su1}$ | DDACK# setup to DIOx# | min | 0 | 0 | 0 | 0 | 0 | 0 |
| $t_{su2}$ | DDATA [15:0] setup to IOWR# | min | 250 | 100 | 35 | 100 | 30 | 20 |
| $t_{su3}$ | DDATA[15:0] setup to PCLK | min | 10 | 10 | 10 | 10 | 10 | 10 |
| $t_{su4}$ | DDRQ setup to PCLK | min | 10 | 10 | 10 | 10 | 10 | 10 |
| $t_{h1}$ | DDACK# hold from DIOx# | min | 0 | 0 | 0 | 20 | 5 | 5 |
| $t_{h2}$ | DDATA [15:0] hold from IORD# | min | 5 | 5 | 5 | 5 | 5 | 5 |
| $t_{h3}$ | DDATA [15:0] hold from IOWR# | min | 5 | 5 | 5 | 5 | 5 | 5 |
| $t_{h4}$ | DDATA[15:0] hold from PCLK | min | 4 | 4 | 4 | 4 | 4 | 4 |
| $t_{h5}$ | DDRQ hold from PCLK | min | 2 | 2 | 2 | 2 | 2 | 2 |
| $t_{vd1}$ | DDRQ to DDACK# valid delay | max | 200 | 100 | 80 | | | 35 |
| $t_{vd2}$ | DDRQ to DIOR# valid delay | min | | | | 120 | 40 | 35 |
| $t_{vd3}$ | DDRQ to DIOW# valid delay | min | | | | 40 | 40 | 35 |
| $t_{vd4}$ | PCLK to DDACK# valid delay | min | 2 | 2 | 2 | 2 | 2 | 2 |
| $t_{vd4}$ | PCLK to DDACK# valid delay | max | 20 | 20 | 20 | 20 | 20 | 20 |
| $t_{vd5}$ | PCLK to DDATA[15:0] valid delay | min | 2 | 2 | 2 | 2 | 2 | 2 |
| $t_{vd5}$ | PCLK to DDATA[15:0] valid delay | max | 20 | 20 | 20 | 20 | 20 | 20 |
| $t_{vd6}$ | PCLK to MASTER#, SOE# valid delay | min | 2 | 2 | 2 | 2 | 2 | 2 |
| $t_{vd6}$ | PCLK to MASTER#, SOE# valid delay | max | 20 | 20 | 20 | 20 | 20 | 20 |
| $t_{pw1}$ | DIOx# active pulse width | min | 480 | 240 | 120 | 215 | 80 | 70 |
| $t_{pw2}$ | DIOR# inactive pulse width | min | – | – | – | 50 | 50 | 25 |
| $t_{pw3}$ | DIOW# inactive pulse width | min | – | – | – | 215 | 50 | 25 |

***Note:***
*All timings are in nanoseconds and reference figure* **52**

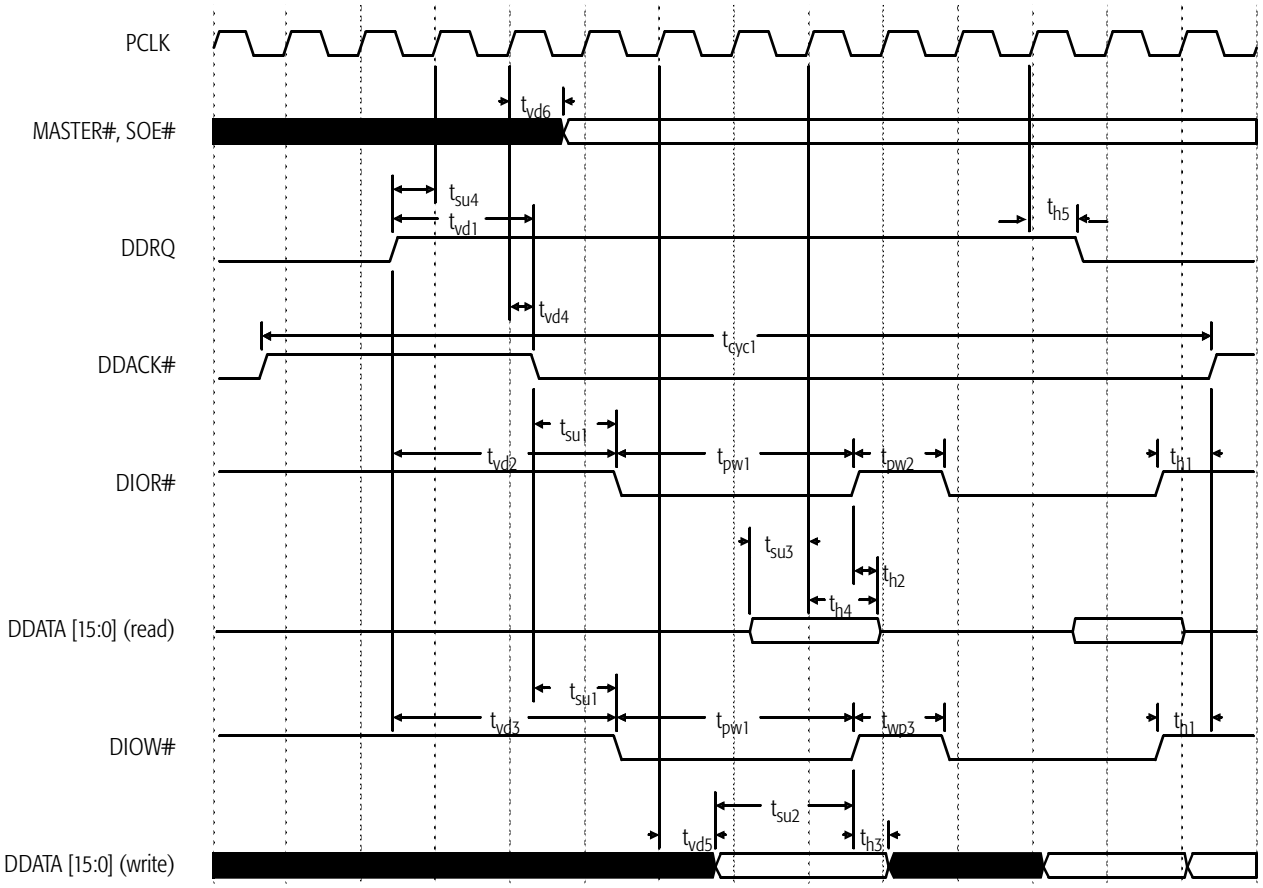


**Figure 52. EIDE DMA**

# 9.7 Ultra DMA-33 IDE Bus Interface Timing

**Table 107. Ultra DMA-33 IDE Bus Interface Timing**

| Symbol | Parameter Description | Preliminary Data | | Figure | Comments |
|---|---|---|---|---|---|
| | | Min | Max | | |
| $t_{dh1}$ | Data hold time for read initial (rise) | 6 ns | | 28 | |
| $t_{dvh5}$ | Data hold time for write terminating | 6 ns | | 32 | |
| $t_{dvh2}$ | Data hold time for write initial (fall) | 18 ns | | 31 | |
| $t_{dvh2}$ | Data hold time for write initial | 18 ns | | 31 | |
| $t_{dvh4}$ | Data hold time for read terminating | 6 ns | | 29 | |
| $t_{rdh}$ | Data hold time during PIO and DMA read | 20 ns | | 25 | |
| $t_{wdh}$ | Data hold time during PIO and DMA write | 20 ns | | 25 | |
| $t_{ds1}$ | Data setup time for read initial | 5 ns | | 27 | |
| $t_{dvs2}$ | Data setup time for write initial (fall) | 43 ns | | 31 | |
| $t_{dvs2}$ | Data setup time for write initial | 42 ns | | 31 | |
| $t_{dvs4}$ | Data setup time for read terminating | 34 ns | | 29 | |
| $t_{dvs5}$ | Data setup time for write terminating | 34 ns | | 32 | |
| $t_{rds}$ | Data setup time during PIO and DMA read | 30 ns | | 25 | |
| $t_{wds}$ | Data setup time during PIO and DMA write | 30 ns | | 25 | |
| $t_{env1}$ | Envelope time for read initial | 20 ns | 70 ns | 27 | |
| $t_{env2}$ | Envelope time for write initial (rise) | 20 ns | 70 ns | 31 | |
| $t_{li4}$ | Limited interlock time (to STOP) | 0 ns | 150 ns | 29 | |
| $t_{li4}$ | Limited interlock time (to Host DMARDY) | 0 ns | 150 ns | 29 | |
| $t_{li5}$ | Limited interlock time (to STOP) | 0 ns | 150 ns | 32 | |
| $t_{li5}$ | Limited interlock time (to Host STROBE) | 0 ns | 150 ns | 32 | |
| $t_{li5}$ | Limited interlock time | 0 ns | 150 ns | 25 | |
| $t_{mli5}$ | Limited interlock time with minimum | 20 ns | | 32 | |
| $t_{mli6}$ | Limited interlock time with minimum | 20 ns | | 30 | |
| $t_{rfs}$ | READY to final STROBE time | | 50 ns | 28 | |
| $t_{rp}$ | READY to Pause time | 100 ns | | 28 | |
| $t_{za4}$ | Delay time required for output drives turning on | 20 ns | | 29 | |
| $t_{za6}$ | Delay time required for output drives turning on | 34 ns | | 32 | |
| $t_2$ | Delay time of PCLK to DCS3#, DCS1# | 2 ns | 20 ns | 25 | |
| $t_3$ | Delay time of PCLK to DA2-DA0 | 2 ns | 20 ns | 25 | |
| $t_4$ | Delay time of PCLK to DIOW# | 2 ns | 20 ns | 25 | |
| $t_5$ | Delay time of PCLK to DIOR# | 2 ns | 20 ns | 25 | |

# 10 Pin Designations

## 10.1 Pin Designation Table

The 272 pins of the AMD-756 peripheral bus controller are listed in the Table 108, grouped according to their functions.

**Table 108. Functional Grouping**

| PCI Bus Interface | | ISA Bus Control | | | | EIDE Interface | |
|---|---|---|---|---|---|---|---|
| **Pin Name** | **Pin No.** | **Pin Name** | **Pin No.** | **Pin Name** | **Pin No.** | **Pin Name** | **Pin No.** |
| AD0 | C6 | AEN | G3 | RSTDRV | A5 | DADDRP2 | V18 |
| AD1 | B6 | BALE | B5 | SA0 | B2 | DADDRP1 | Y17 |
| AD2 | A7 | BCLK | V2 | SA1 | A1 | DADDRP0 | Y18 |
| AD3 | A6 | DACK0# | K2 | SA2 | A2 | DADDRS2 | Y19 |
| AD4 | C7 | DACK1# | N3 | SA3 | U10 | DADDRS1 | W17 |
| AD5 | D6 | DACK2# | V8 | SA4 | U9 | DADDRS0 | W18 |
| AD6 | B7 | DACK3# | K3 | SA5 | U8 | DCS1P# | W19 |
| AD7 | D7 | DACK5# | M2 | SA6 | U7 | DCS1S# | V19 |
| AD8 | B8 | DACK6# | P2 | SA7 | U6 | DCS3P# | Y20 |
| AD9 | C8 | DACK7# | T2 | SA8 | V3 | DCS3S# | W20 |
| AD10 | B9 | DRQ0 | L2 | SA9 | U3 | DDACKP# | Y16 |
| AD11 | D8 | DRQ1 | P3 | SA10 | T3 | DDACKS# | W16 |
| AD12 | A9 | DRQ2 | E4 | SA11 | T4 | DDATAP0 | Y11 |
| AD13 | B10 | DRQ3 | L3 | SA12 | R4 | DDATAP1 | Y10 |
| AD14 | A10 | DRQ5 | N2 | SA13 | P4 | DDATAP2 | Y9 |
| AD15 | B11 | DRQ6 | R2 | SA14 | M3 | DDATAP3 | Y8 |
| AD16 | D12 | DRQ7 | U2 | SA15 | M4 | DDATAP4 | W7 |
| AD17 | C12 | IOCHCK# | A4 | SA16 | H3 | DDATAP5 | Y5 |
| AD18 | B14 | IOCHRDY | G4 | SBHE# | B1 | DDATAP6 | W4 |
| AD19 | C13 | IOCS16# | D2 | SD0 | F3 | DDATAP7 | Y2 |
| AD20 | B15 | IOR# | K4 | SD1 | F4 | DDATAP8 | W2 |
| AD21 | A14 | IOW# | J3 | SD2 | E3 | DDATAP9 | Y3 |
| AD22 | A15 | IRQ3 | V7 | SD3 | D3 | DDATAP10 | W5 |
| AD23 | B16 | IRQ4 | V6 | SD4 | C3 | DDATAP11 | Y6 |
| AD24 | C14 | IRQ5 | V5 | SD5 | B3 | DDATAP12 | W8 |
| AD25 | A16 | IRQ6 | V4 | SD6 | B4 | DDATAP13 | W9 |
| AD26 | D14 | IRQ7 | U5 | SD7 | A3 | DDATAP14 | W10 |
| AD27 | B17 | IRQ9 | C4 | SD8 | M1 | DDATAP15 | W11 |
| AD28 | C15 | IRQ10 | E2 | SD9 | N1 | DDATAS0 | U18 |
| AD29 | A17 | IRQ11 | F2 | SD10 | P1 | DDATAS1 | V16 |
| AD30 | B18 | IRQ12 | G2 | SD11 | R1 | DDATAS2 | V15 |
| AD31 | A18 | IRQ14 | J2 | SD12 | T1 | DDATAS3 | V14 |
| C/BE0# | A8 | IRQ15 | H2 | SD13 | U1 | DDATAS4 | V13 |
| C/BE1# | C9 | LA17 | J1 | SD14 | V1 | DDATAS5 | U13 |
| C/BE2# | A13 | LA18 | H1 | SD15 | Y1 | DDATAS6 | V12 |
| C/BE3# | D13 | LA19 | G1 | SMEMR# | J4 | DDATAS7 | U12 |
| DEVSEL# | C10 | LA20 | F1 | SMEMW# | H4 | DDATAS8 | W3 |
| FRAME# | C11 | LA21 | E1 | SPKR | V11 | DDATAS9 | Y4 |
| IDSEL | B13 | LA22 | D1 | TC | V9 | DDATAS10 | W6 |
| IRDY# | D10 | LA23 | C1 | | | DDATAS11 | Y7 |
| PAR | D9 | MASTER# | W1 | | | DDATAS12 | U14 |
| PCIRST# | D15 | MEMCS16# | C2 | | | DDATAS13 | U15 |
| PCLK | A20 | MEMR# | K1 | | | DDATAS14 | U16 |
| PGNT# | A19 | MEMW# | L1 | | | DDATAS15 | V17 |
| PIRQA# | C16 | OSC | U20 | | | DDRQP | Y12 |
| PIRQB# | C18 | REFRESH# | R3 | | | DDRQS | W12 |
| PIRQC# | D16 | ROM_KBCS# | N20 | | | DIORP# | Y14 |
| PIRQD# | C17 | | | | | DIORS# | W14 |
| PREQ# | B19 | | | | | DIOWP# | Y13 |
| SERR# | A11 | | | | | DIOWS# | W13 |
| STOP# | B12 | | | | | DRDYP# | Y15 |
| TRDY# | A12 | | | | | DRDYS# | W15 |

** Multifunction pin*

**Table 108. Functional Grouping (continued)**

| USB Interface | | Miscellaneous | | Keyboard Interface | | | |
|---|---|---|---|---|---|---|---|
| **Pin Name** | **Pin No.** | **Pin Name** | **Pin No.** | **Pin Name** | **Pin No.** | **Pin Name** | **Pin No.** |
| USBCLK | C19 | USBIRQ (PICCLK) | T17 | KBCK(KA20G) | K17 | | |
| USBP3 | D18 | KBCIRQ (PICD0#) | R20 | KBDT (KBRC#) | J20 | | |
| USBP2 | D20 | PITIRQ (PICD1#) | R19 | KEYLOCK(DBRDY) | K19 | | |
| USBP1 | E18 | SCIIRQ (WSC#) | N18 | MSCK(EKIRQ1) | K18 | | |
| USBP0 | E20 | SPARE1 | C5 | MSDT(EKIRQ12) | K20 | | |
| USBN3 | D19 | TEST# | D5 | | | | |
| USBN2 | E17 | | | | | | |
| USBN1 | E19 | | | | | | |
| USBN0 | F17 | | | | | | |
| USBOC1#(IRQ12) | G2 | | | | | | |
| USBOC0# | B20 | | | | | | |

| Power & Ground | | RTC | | CPU Interface | | Power Management | |
|---|---|---|---|---|---|---|---|
| **Pin Name** | **Pin No.** | **Pin Name** | **Pin No.** | **Pin Name** | **Pin No.** | **Pin Name** | **Pin No.** |
| GND | J9 | RTCXIN | F20 | A20M# | M18 | C32KHZ | V20 |
| | J10 | RTCXOUT | G17 | CPURST | N19 | CACHEZZ | P20 |
| | J11 | | | FERR# | T19 | (PNPIRQ1) | |
| | J12 | | | IGNNE# | T18 | CPUSLEEP# | L18 |
| | K9 | | | INIT | T20 | (PNPCS0#) | |
| | K10 | | | INTR | R17 | CPUSTOP# | P19 |
| | K11 | | | NMI | R18 | (PNPCS1#) | |
| | K12 | | | SMI# | M17 | DCSTOP# | L19 |
| | L9 | | | STPCLK# | N17 | (PNPIRQ2) | |
| | L10 | | | | | EXTSMI# | H19 |
| | L11 | | | | | (BMREQ#) | |
| | L12 | | | | | FLAGRD# | M20 |
| | M9 | | | | | (PNPDAK#) | |
| | M10 | | | | | FLAGWR | L20 |
| | M11 | | | | | (PNPDRQ) | |
| | M12 | | | | | INTIRQ8# | V10 |
| | | | | | | (SQWAVE) | |
| GND-USB | F18 | | | | | PCISTOP# | P18 |
| | | | | | | (PNPIRQ0) | |
| $V_{DD3}$ | D4 | | | | | PME# | H20 |
| | D11 | | | | | PWRBTN# | H18 |
| | D17 | | | | | PWRGD | J18 |
| | L4 | | | | | PWRON# | J17 |
| | L17 | | | | | RI# | G18 |
| | U4 | | | | | SERIRQ | P17 |
| | U11 | | | | | (MSIRQ) | |
| | U17 | | | | | SMBUSC | G19 |
| $V_{DD\_REF}$ | N4 | | | | | SMBUSD | G20 |
| | | | | | | SLPBTN# | H17 |
| $V_{DD\_RTC}$ | F19 | | | | | (PMIRQ1) | |
| | | | | | | (EXTIRQ8#) | |
| $V_{DD\_SOFT}$ | J19 | | | | | SUSPEND# | M19 |
| | | | | | | THERM# | U19 |
| $V_{DD\_USB}$ | C20 | | | | | (PMIRQ0) | |

** Multifunction pin*

**Table 109. I/O Cell Types**

| Name | Description |
|---|---|
| Input | Input signal only. |
| Output | Output signal only. This includes outputs that are capable of being in the high-impedance state. |
| OD | Open drain output. These pins are driven low, but pulled up by external circuitry. |
| I/O | Input or output signal. |
| I/OD | Input or open-drain output. |
| w/H | With hysteresis on the input. |
| Analog | Analog pins |

### 10.1.1 State of Pins At Reset

Table 110 shows reference data about each of the AMD-756 peripheral bus controller pins. The Reset column lists the state of the pin while the its power plane is being reset. The Post Reset column lists the state of the pin immediately after that reset. The POS column lists the state of the pin while in the clock-controlled power on suspend state.

**Table 110. State of Pins at Reset**

| Pin Name | Group | Cell | Power Plane | Iol/Ioh | Reset | Post Reset | POS | Notes |
|---|---|---|---|---|---|---|---|---|
| A20M# | CPU | OD | VDD3 | 12 ma | 3-state | 3-state | 3-state | See 8 |
| AD[31:0] | PCI | I/O | VDD3 | 4 / -4 ma | 3-state | 3-state | 3-state | |
| AEN | ISA | Output | VDD3 | 4 / -4 ma | Low | Low | Low | |
| BALE | ISA | Output | VDD3 | see 3 | High | Low | Low | |
| BCLK | ISA | Output | VDD3 | see 3 | Active | Active | Active | |
| C32KHZ | SM | I/O | VDD3 | 4 / -4 ma | Low | Low | See note | Defaults to GPIO output; see 5 |
| CACHE_ZZ | SM | I/O | VDD3 | 4 / -4 ma | Low | Low | See note | Defaults to GPIO output; see 5 |
| C/BE#[3:0] | PCI | I/O | VDD3 | 4 / -4 ma | 3-state | 3-state | 3-state | |
| CPURST# | CPU | OD | VDD3 | 12 ma | Active | Active | Low | See 8 |
| CPUSLEEP# | SM | I/O | VDD3 | 4 / -4 ma | High | High | See note | Defaults to GPIO output; see 5 |
| CPUSTOP# | SM | I/O | VDD3 | 4 / -4 ma | High | High | See note | Defaults to GPIO output; see 5 |
| DACK[7:5,3:0]# | ISA | Output | VDD3 | 4 / -4 ma | High | High | High | |

**Table 110. State of Pins at Reset (continued)**

| Pin Name | Group | Cell | Power Plane | Iol/Ioh | Reset | Post Reset | POS | Notes |
|---|---|---|---|---|---|---|---|---|
| DADDRP[2:0] | IDE | I/O | VDD3 | 4 / -4 ma | Input | Low | Low | See 6 |
| DADDRS[2:0] | IDE | Output | VDD3 | 4 / -4 ma | Low | Low | Low | See 7 |
| DCS1P# | IDE | I/O | VDD3 | 4 / -4 ma | Input | High | High | See 6 |
| DCS1S# | IDE | Output | VDD3 | 4 / -4 ma | High | High | High | See 7 |
| DCS3P# | IDE | Output | VDD3 | 4 / -4 ma | High | High | High | |
| DCS3S# | IDE | Output | VDD3 | 4 / -4 ma | High | High | High | See 7 |
| DCSTOP# | SM | I/O | VDD3 | 4 / -4 ma | High | High | See note | Defaults to GPIO output; see 5 |
| DDACKP# | IDE | Output | VDD3 | 4 / -4 ma | High | High | High | |
| DDACKS# | IDE | Output | VDD3 | 4 / -4 ma | High | High | High | See 7 |
| DDATAP[15:0] | IDE | I/O | VDD3 | 4 / -4 ma | 3-state | 3-state | Low | |
| DDATAS[15:0] | IDE | I/O | VDD3 | 4 / -4 ma | 3-state | 3-state | Low | See 7 |
| DDRQP | IDE | Input | VDD3 | - | - | - | - | |
| DDRQS | IDE | Input | VDD3 | - | - | - | - | See 7 |
| DEVSEL# | PCI | I/O | VDD3 | 4 / -4 ma | 3-state | 3-state | 3-state | |
| DIORP# | IDE | Output | VDD3 | 4 / -4 ma | High | High | High | |
| DIORS# | IDE | Output | VDD3 | 4 / -4 ma | High | High | High | See 7 |
| DIOWP# | IDE | Output | VDD3 | 4 / -4 ma | High | High | High | |
| DIOWS# | IDE | Output | VDD3 | 4 / -4 ma | High | High | High | See 7 |
| DRDYP# | IDE | Input | VDD3 | | - | - | - | |
| DRDYS# | IDE | Input | VDD3 | | - | - | - | See 7 |
| DRQ[7:5,3:0] | ISA | Input | VDD3 | | - | - | - | |
| EXTSMI# | SM | I/O | VDD_SOFT | 4 / -4 ma | Input | Input | Active | Muxed with GPIO12 and BMREQ# |
| FRAME# | PCI | I/O | VDD3 | 4 / -4 ma | 3-state | 3-state | 3-state | |
| FERR# | CPU | Input w/H | VDD3 | | - | - | - | |
| FLAGWR | SM | I/O | VDD3 | 4 / -4 ma | Low | Low | See note | Defaults to GPIO output; see 5 |
| FLAGRD# | SM | I/O | VDD3 | 4 / -4 ma | High | High | See note | Defaults to GPIO output; see 5 |
| GND[15:0] | P&G | Analog | | | Low | Low | Low | |
| GND_USB | P&G | Analog | | | Low | Low | Low | |
| IDSEL | PCI | Input | VDD3 | | - | - | - | |
| IGNNE# | CPU | OD | VDD3 | 12 ma | 3-state | 3-state | 3-state | See 8 |
| INIT# | CPU | OD | VDD3 | 12 ma | low | low | low | See 8 |

**Table 110. State of Pins at Reset (continued)**

| Pin Name | Group | Cell | Power Plane | Iol/Ioh | Reset | Post Reset | POS | Notes |
|---|---|---|---|---|---|---|---|---|
| INTIRQ8# | SM | I/O | VDD3 | 4 / -4 ma | High | High | See note | Defaults to GPIO output; see 5 |
| INTR | CPU | OD | VDD3 | 12 ma | low | low | low | See 8 |
| IOCHK# | ISA | Input | VDD3 | | - | - | - | |
| IOCHRDY | ISA | I/OD | VDD3 | see 3 | 3-state | 3-state | 3-state | |
| IOCS16# | ISA | Input | VDD3 | | - | - | - | |
| IOR# | ISA | I/O | VDD3 | see 3 | High | High | High | |
| IOW# | ISA | I/O | VDD3 | see 3 | High | High | High | |
| IRDY# | PCI | I/O | VDD3 | 4 / -4 ma | 3-state | 3-state | 3-state | |
| IRQ12 | ISA | Input | VDD3 | | - | - | - | Muxed with SMBALERT#, USBOC1# |
| IRQ [15,14,11:9,7:3] | ISA | Input | VDD3 | | - | - | - | |
| KBDT | KBDC | I/O | VDD3 | 4 / -4 ma | See note | See note | Active | See 1 |
| KBCK | KBDC | I/O | VDD3 | 4 / -4 ma | See note | See note | Active | See 1 |
| KEYLOCK | KBDC | I/O | VDD3 | 4 / -4 ma | Input | Input | Active | Muxed with GPIO13, PRDY |
| LA[23:17] | ISA | I/O | VDD3 | see 3 | 3-state | Low | Last state | |
| MASTER# | ISA | Input | VDD3 | | - | - | - | |
| MEMCS16# | ISA | I/OD | VDD3 | see 3 | 3-state | 3-state | 3-state | |
| MEMR# | ISA | I/O | VDD3 | see 3 | High | High | High | |
| MEMW# | ISA | I/O | VDD3 | see 3 | High | High | High | |
| MSDT | KBDC | I/O | VDD3 | 4 / -4 ma | See note | See note | Active | See 1 |
| MSCK | KBDC | I/O | VDD3 | 4 / -4 ma | See note | See note | Active | See 1 |
| NMI | CPU | OD | VDD3 | 12 ma | low | low | low | See 8 |
| OSC | ISA | Input | VDD3 | | - | - | - | |
| PAR | PCI | I/O | VDD3 | 4 / -4 ma | 3-state | 3-state | 3-state | |
| PCIRST# | PCI | Output | VDD3 | 4 / -4 ma | Active | Active | High | |
| PCISTOP# | SM | I/O | VDD3 | 4 / -4 ma | High | High | See note | Defaults to GPIO output; see 5 |
| PCLK | PCI | Input | VDD3 | | - | - | - | Must be active during reset |
| PGNT# | PCI | Input | VDD3 | | - | - | - | |
| PIRQ[A,B,C,D]# | PCI | Input | VDD3 | | - | - | - | |

**Table 110. State of Pins at Reset (continued)**

| Pin Name | Group | Cell | Power Plane | Iol/Ioh | Reset | Post Reset | POS | Notes |
|---|---|---|---|---|---|---|---|---|
| PME# | SM | Input | VDD_SOFT | | - | - | - | |
| PREQ# | PCI | Output | VDD3 | 4 / -4 ma | High | High | High | |
| PWRBTN# | SM | Input | VDD_SOFT | | - | - | Active | |
| PWRGD | SM | Input | VDD_SOFT | | - | - | - | |
| PWRON# | SM | OD | VDD_SOFT | 4 ma | Low | Low | Active | |
| REFRESH# | ISA | I/O | VDD3 | see 3 | High | High | Active | |
| RI# | SM | I/O | VDD_SOFT | 4 / -4 ma | Input | Input | Active | Muxed with GPIO14 |
| ROM_KBCS# | ISA | I/O | VDD3 | 4 / -4 ma | Input | High | High | See 1 |
| RSTDRV | ISA | Output | VDD3 | see 3 | Active | Active | Low | |
| RTCX_IN | RTC | Analog | VDD_RTC | | Active | Active | Active | |
| RTCX_OUT | RTC | Analog | VDD_RTC | | Active | Active | Active | |
| SA[16:0] | ISA | I/O | VDD3 | see 3 | 3-state | Active | Last state | See 4 |
| SBHE# | ISA | I/O | VDD3 | see 3 | 3-state | Active | Last state | |
| SD[15:0] | ISA | I/O | VDD3 | see 3 | 3-state | Active | 3-state | |
| SERIRQ | SM | I/O | VDD3 | 4 / -4 ma | Input | Input | See note | Defaults to GPIO input; see 5 |
| SERR# | PCI | Input | VDD3 | | - | - | - | |
| SMBUSC | SM | I w/H / OD | VDD_SOFT | 4 / -4 ma | 3-state | 3-state | Active | Muxed with GPIO0 |
| SMBUSD | SM | I w/H / OD | VDD_SOFT | 4 / -4 ma | 3-state | 3-state | Active | Muxed with GPIO1 |
| SMEMR# | ISA | Output | VDD3 | see 3 | High | High | High | |
| SMEMW# | ISA | Output | VDD3 | see 3 | High | High | High | |
| SMI# | CPU | OD | VDD3 | 12/-12 ma | 3-state | 3-state | 3-state | See 8 |
| SPARE1 | MISC | - | VDD3 | - | - | - | - | |
| SPKR | ISA | I/O | VDD3 | 12 / -12 ma | Input | Low | Last state | See 2 |
| SLPBTN# | SM | I/O | VDD_SOFT | 4 / -4 ma | - | - | Active | Muxed with GPIO3, PMIRQ1, and EXTIRQ8# |
| STOP# | PCI | I/O | VDD3 | 4 / -4 ma | 3-state | 3-state | 3-state | |
| STPCLK# | CPU | OD | VDD3 | 12/-12 ma | 3-state | 3-state | Active | See 8 |
| SUSPEND# | SM | I/O | VDD3 | 4 / -4 ma | High | High | Active | Muxed with GPIO4 |
| TC | ISA | Output | VDD3 | see 3 | Low | Low | Low | |
| TEST# | MISC | Input | VDD3 | | - | - | - | |
| THERM# | SM | I/O | VDD3 | 4 / -4 ma | Input | Input | Active | Muxed with GPIO2, PMIRQ0 |

**Table 110. State of Pins at Reset (continued)**

| Pin Name | Group | Cell | Power Plane | Iol/Ioh | Reset | Post Reset | POS | Notes |
|---|---|---|---|---|---|---|---|---|
| TRDY# | PCI | I/O | VDD3 | 4 / -4 ma | 3-state | 3-state | 3-state | |
| USBCLK | USB | Input | VDD3 | | - | - | - | |
| USBN[3:0] | USB | Analog | VDD_USB | | 3-state | 3-state | Active | |
| USBOC0# | USB | Input | VDD3 | | - | - | Active | |
| USBP[3:0] | USB | Analog | VDD_USB | | 3-state | 3-state | Active | |
| VDD3[7:0] | P&G | Analog | | | - | - | - | |
| VDD_REF | P&G | Analog | | | - | - | - | |
| VDD_RTC | P&G | Analog | | | - | - | - | |
| VDD_SOFT | P&G | Analog | | | - | - | - | |
| VDD_USB | P&G | Analog | | | - | - | - | |

*Notes:*

1. *During reset, the KBCK, KBDT, MSCK, and MSDT pins' default is determined by the state of the ROM_KBCS# signal which selects between an internal or external keyboard controller. If the internal keyboard controller is selected, then KBCK and MSCK are low outputs and KBDT and MSDT are inputs. If the external keyboard controller is selected, then KBCK (KA20G), MSCK (IRQ1), and MSDT (IRQ12) are inputs and KBDT (KBRC#) is a high output. After reset, all the outputs are functional. During POS, these pins remain functional.*
2. *SPKR is only an input during PWRGD reset. The state of SPKR is latched in function 3 offset 48 (but does not control anything).*
3. *These I/O current for these pins is can be selected to be either 24 or 12 milliamps (source and sink), by function 0, offset 49, bit 3, ISA12MA.*
4. *SA[15:0] can be multiplexed with the muxed IDE controller's DDATAM[15:0] pins during operation.*
5. *These pins can be individually configured to be either PNP controls, GPIOs, or power management pins. As PNP pins, these are held in the inactive state during POS. As GPIO outputs, these just stay in their last state during POS. As power management pins, these pins perform specific functions during C2, C3, and POS as shown in the system management section of this document.*
6. *DADDRP[2:0] and DCS1P# are only inputs during PWRGD reset. Their states are latched in function 3 offset 48.*
8. *The I/O cell for the output CPU signals, A20M#, CPURST, IGNNE#, INTR, INIT, NMI, SMI#, and STPCLK#, can be selected by the state of the SPKR pin at reset to be either open-drain or normal. Also, PICD0#, PICD1#, WSC#, and PICCLK are all muxed with internal AMD-756 peripheral bus controller interrupts, as selected by function 0, offset 49, bit 6, RVLINT. As AMD-756 peripheral bus controller interrupts, they are normal outputs that are driven both high and low.*

## 10.1.2 AMD-756™ Peripheral Bus Controller Pin Diagram

Figure 53 shows the pin diagram for the AMD-756 peripheral bus controller.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | SA1 | SA2 | SD7 | IOCHK# | RSTDRV | AD3 | AD2 | C/BE#0 | AD12 | AD14 | SERR# | TRDY# | C/BE#2 | AD21 | AD22 | AD25 | AD29 | AD31 | PGNT# | PCLK |
| B | SBHE# | SA0 | SD5 | SD6 | BALE | AD1 | AD6 | AD8 | AD10 | AD13 | AD15 | STOP# | IDSEL | AD18 | AD20 | AD23 | AD27 | AD30 | PREQ# | USBOC0# |
| C | LA23 | MEM-CS16# | SD4 | IRQ9 | SPARE1 | AD0 | AD4 | AD9 | C/BE#1 | DEVSEL# | FRAME# | AD17 | AD19 | AD24 | AD28 | PIRQA# | PIRQD# | PIRQB# | USBCLK | VDD_USB |
| D | LA22 | IOCS16# | SD3 | VDD3 | TEST# | AD5 | AD7 | AD11 | PAR | IRDY# | VDD3 | AD16 | C/BE#3 | AD26 | PCIRST# | PIRQC# | VDD3 | USBP3 | USBN3 | USBP2 |
| E | LA21 | IRQ10 | SD2 | DRQ2 | | | | | | | | | | | | | USBN2 | USBP1 | USBN1 | USBP0 |
| F | LA20 | IRQ11 | SD0 | SD1 | | | | | | | | | | | | | USBN0 | GND_USB | VDD_RTC | RTCX_IN |
| G | LA19 | IRQ12 | AEN | IOCHRDY | | | | | | | | | | | | | RTCX_OUT | RI# | SMBUSC | SMBUSD |
| H | LA18 | IRQ15 | SA16 | SMEMW# | | | | | | | | | | | | | SLPBTN# | PWR-BTN# | EXTSMI# | PME# |
| J | LA17 | IRQ14 | IOW# | SMEMR# | | | | | GND | GND | GND | GND | | | | | PWRON# | PWRGD | VDD_SOFT | KBDT |
| K | MEMR# | DACK0# | DACK3# | IOR# | | | | | GND | GND | GND | GND | | | | | KBCK | MSCK | KEYLOCK | MSDT |
| L | MEMW# | DRQ0 | DRQ3 | VDD3 | | | | | GND | GND | GND | GND | | | | | VDD3 | CPU-SLEEP# | DCSTOP# | FLAGWR |
| M | SD8 | DACK5# | SA14 | SA15 | | | | | GND | GND | GND | GND | | | | | SMI# | A20M# | SUS-PEND# | FLAGRD# |
| N | SD9 | DRQ5 | DACK1# | VDD_REF | | | | | | | | | | | | | STPCLK# | WSC# | CPURST# | ROM_KBCS# |
| P | SD10 | DACK6# | DRQ1 | SA13 | | | | | | | | | | | | | SERIRQ | PCISTOP# | CPU-STOP# | CACHE_ZZ |
| R | SD11 | DRQ6 | REFRESH# | SA12 | | | | | | | | | | | | | INTR | NMI | PICD1# | PICD0# |
| T | SD12 | DACK7# | SA10 | SA11 | | | | | | | | | | | | | PICCLK | IGNNE# | FERR# | INIT# |
| U | SD13 | DRQ7 | SA9 | VDD3 | IRQ7 | SA7 | SA6 | SA5 | SA4 | SA3 | VDD3 | DDATAS7 | DDATAS5 | DDATAS12 | DDATAS13 | DDATAS14 | VDD3 | DDATAS0 | THERM# | OSC |
| V | SD14 | BCLK | SA8 | IRQ6 | IRQ5 | IRQ4 | IRQ3 | DACK2# | TC | INTIRQ8# | SPKR | DDATAS6 | DDATAS4 | DDATAS3 | DDATAS2 | DDATAS1 | DDATAS15 | DADDRP2 | DCS1S# | C32KHZ |
| W | MASTER# | DDATAP8 | DDATAS8 | DDATAP6 | DDATAP10 | DDATAS10 | DDATAP4 | DDATAP12 | DDATAP13 | DDATAP14 | DDATAP15 | DDRQS | DIOWS# | DIORS# | DRDYS# | DDACKS# | DADDRS1 | DADDRS0 | DCS1P# | DCS3S# |
| Y | SD15 | DDATAP7 | DDATAP9 | DDATAS9 | DDATAP5 | DDATAP11 | DDATAS11 | DDATAP3 | DDATAP2 | DDATAP1 | DDATAP0 | DDRQP | DIOWP# | DIORP# | DRDYP# | DDACKP# | DADDRP1 | DADDRP0 | DADDRS2 | DCS3P# |

**Figure 53. AMD-756™ Peripheral Bus Controller Pin Diagram**

### 10.1.3 Multiplexed Pins

Table 111 shows pins that are multiplexed with other functions that can be selected by the systemboard designer. Shaded areas are the power-up defaults from the PWRGD reset. These can be individually changed after power up through the selection register shown.

**Table 111. Multiplexed Pins Power Up Defaults**

| Pin name | Alternate | GPIO Pin | Selection register |
|---|---|---|---|
| SMBUSC | | GPIO0 | PM00 +C0h |
| SMBUSD | | GPIO1 | PM00 +C1h |
| THERM# | PMIRQ0 | GPIO2 | PM00 +C2h |
| SLPBTN# | PMIRQ1 | GPIO3 | PM00 +C3h (this pin can also be specified to be EXTIRQ8#) |
| SUSPEND# | | GPIO4 | PM00 +C4h |
| CPUSLEEP# | PNPCS0# | GPIO5 | PM00 +C5h |
| CPUSTOP# | PNPCS1# | GPIO6 | PM00 +C6h |
| PCISTOP# | PNPIRQ0 | GPIO7 | PM00 +C7h |
| CACHE_ZZ | PNPIRQ1 | GPIO8 | PM00 +C8h |
| DCSTOP# | PNPIRQ2 | GPIO9 | PM00 +C9h |
| FLAGWR | PNPDRQ | GPIO10 | PM00 +CAh |
| FLAGRD# | PNPDAK# | GPIO11 | PM00 +CBh |
| EXTSMI# | BMREQ# | GPIO12 | PM00 +CCh |
| KEYLOCK | DBRDY | GPIO13 | PM00 +CDh |
| RI# | | GPIO14 | PM00 +CEh |
| C32KHZ | | GPIO15 | PM00 +CFh |
| INTIRQ8# | SQWAVE | GPIO16 | PM00 +D0h |
| SERIRQ | MSIRQ | GPIO17 | PM00 +D1h |
| IRQ12 | SMBALERT# | | Function 3 Offset 46 bits[10:9] (this can also be a second USB over-current indicator, USBOC1#) |

Table 112 shows the pin defaults that are determined by ROM_KBCS# at the trailing edge of PWRGD reset.

**Table 112. Multiplexed Pins determined by ROM_KBCS#**

| Pin name | Alternate | Selection register |
|---|---|---|
| KBCK | KA20G | Function 3 offset 48 bit[INTKBC] |
| KBDT | KBRC# | Function 3 offset 48 bit[INTKBC] |
| MSCK | EKIRQ1 | Function 3 offset 48 bit[INTKBC] |
| MSDT | EKIRQ12 | Function 3 offset 48 bit[INTKBC] |

Table 113 show the pins that are multiplexed between the control signals for the second processor and AMD-756 peripheral bus controller internal interrupts. At power up, these signals all default to the processor interrupt signals.

**Table 113. Multiplexed Pins selected by Interrupt Sources**

| Pin name | Alternate | Selection register |
|---|---|---|
| PICD1# | PITIRQ | Function 0 offset 49 bit[RVLINT] |
| PICD0# | KBCIRQ | Function 0 offset 49 bit[RVLINT] |
| WSC# | SCIIRQ | Function 0 offset 49 bit[RVLINT] |
| PICCLK | USBIRQ | Function 0 offset 49 bit[RVLINT] |

# 11 Package Specifications

The AMD-756 peripheral bus controller comes in a 272-ball Plastic Ball Grid Array (PBGA). The dimensions and thermal specifications are shown below.

$\theta_{JA}$ <= 25 °C/W

$\theta_{JC}$ = Junction to case thermal resistance (not available)

$\theta_{CA}$ = Case to ambient thermal resistance (not available)

**Table 114. 272-Pin PBGA Package Preliminary Specification**

| Symbol | Millimeters | | | Inches | | | Notes |
|---|---|---|---|---|---|---|---|
| | Min | Typ | Max | Min | Typ | Max | |
| A | 26.80 | 27.00 | 27.20 | 1.06 | 1.063 | 1.07 | |
| B | | 24.13 | | | 0.95 | | |
| C | | 24.00 | | | 0.945 | | |
| D | | 24.00 | | | 0.945 | | |
| E | 2.20 | 2.33 | 2.46 | 0.087 | 0.092 | 0.097 | |
| F | | 1.17 | | | 0.046 | | |
| G | 0.51 | 0.56 | 0.61 | 0.020 | 0.022 | 0.024 | |
| H | 0.50 | 0.60 | 0.70 | 0.020 | 0.024 | 0.028 | |
| M | | 1.27 | | | 0,05 | | |

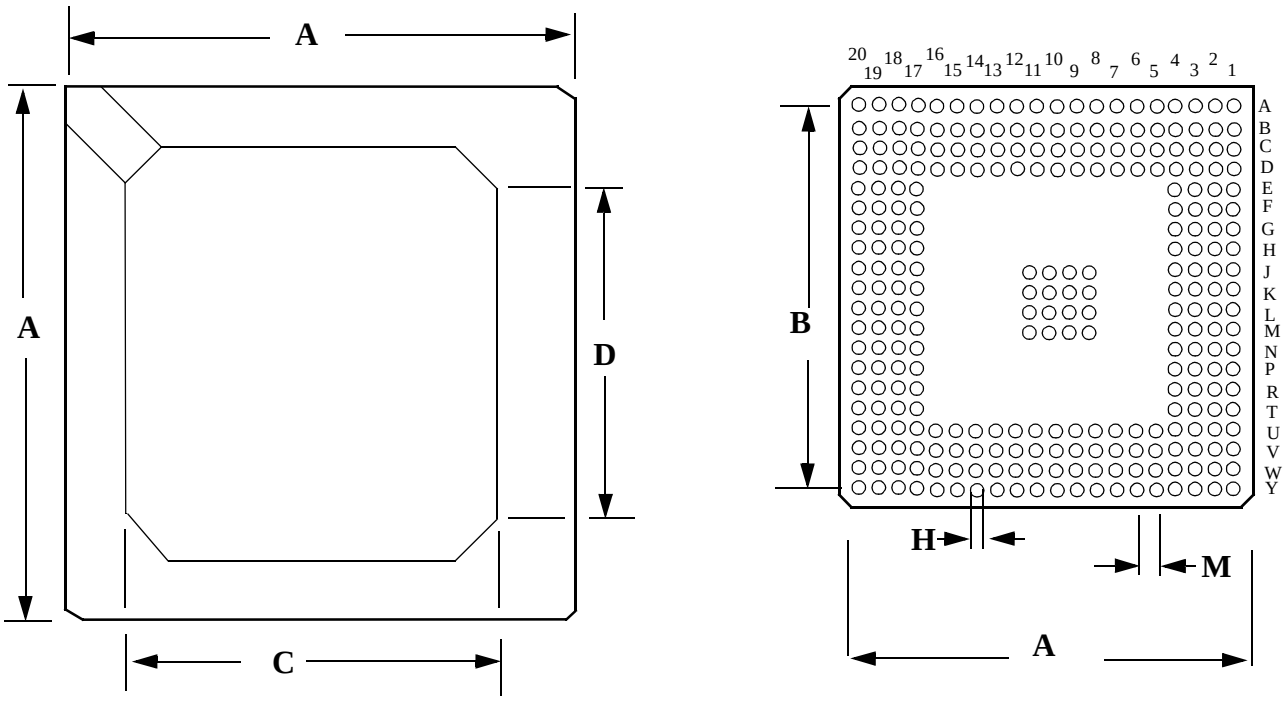


**Figure 54. 272 - Pin PBGA Package Preliminary Specification**

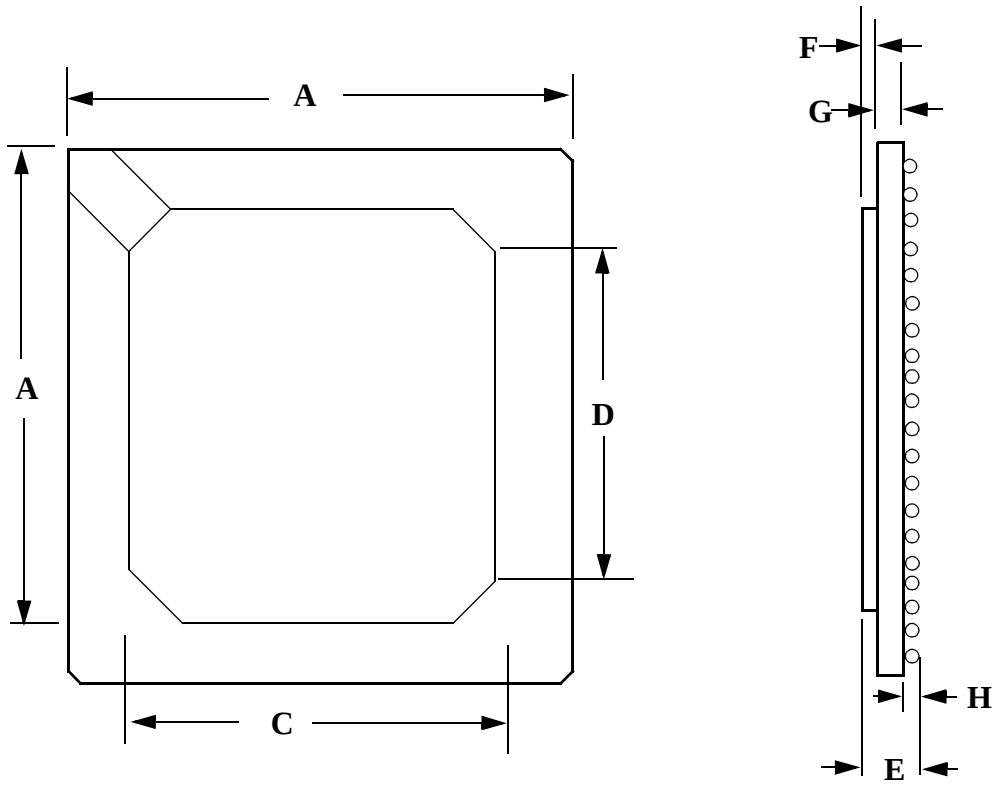


**Figure 55. 272 - Pin PBGA Package Preliminary Specification, continued**

## A



## B



## C



## D

## E



## F



## G



## H



## I

## K



## L



## M



## N



## O



## P

## R



## S

## T



## U



## V



## W



## X