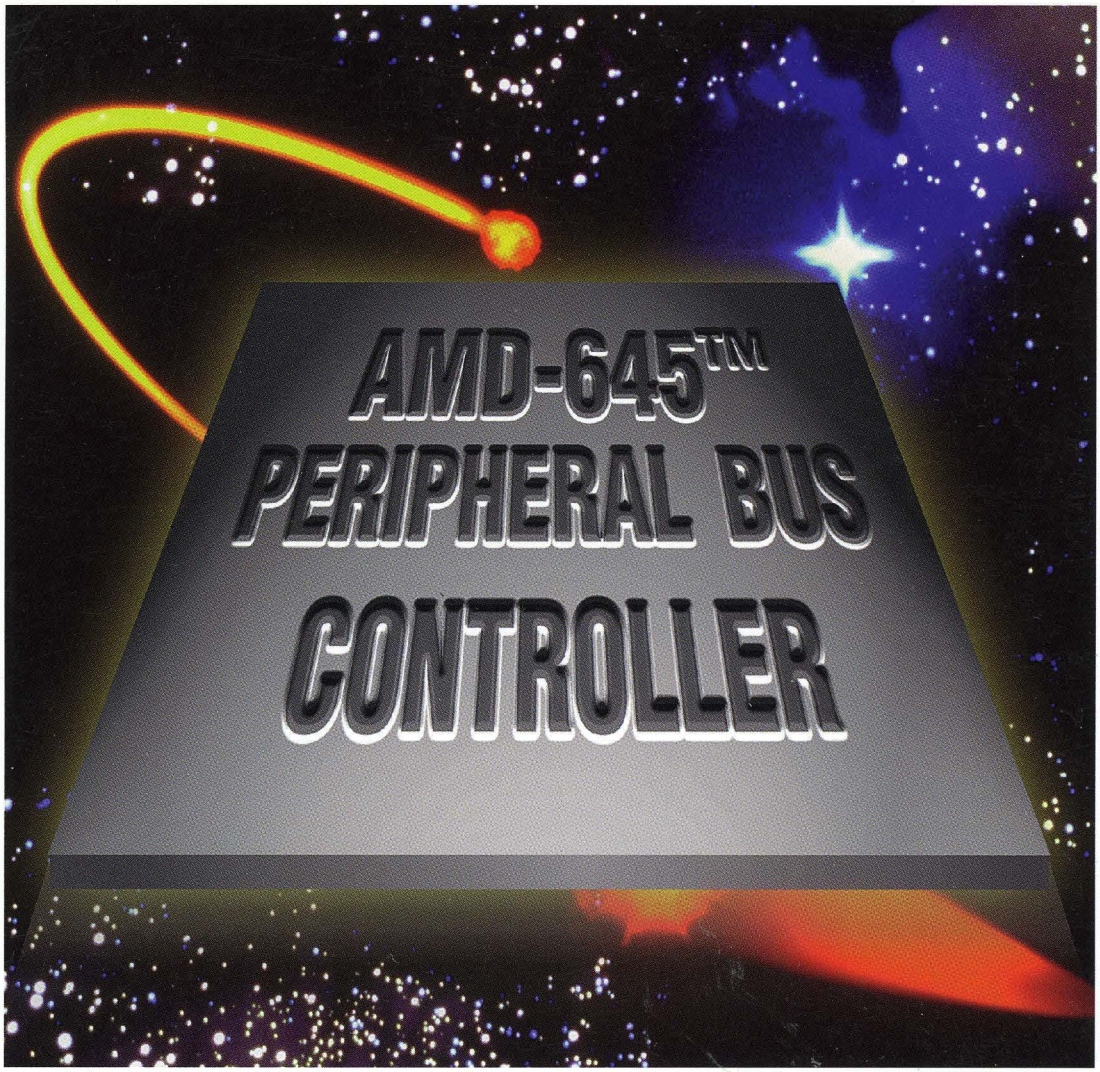
AMD

AMD-645™ Peripheral Bus Controller

Data Sheet



Preliminary Information

AMD-645™

Peripheral Bus Controller

Data Sheet

AMD

Contents

List of Figures

List of Tables

1 Features

The AMD-640™ chipset is a highly integrated system solution designed to deliver superior performance for the AMD-K5™ processor, AMD-K6™ MMX processor, and other Socket 7-compatible processors. The AMD-640 chipset consists of the AMD-640 System Controller in a 328-pin BGA package and the AMD-645™ Peripheral Bus Controller in a 208-pin PQFP package. The AMD-645 Peripheral Bus Controller features an integrated ISA bus controller, enhanced master mode PCI IDE controller with ultra DMA-33 support, USB controller, keyboard/mouse controller, and real-time clock.

This document describes the features and operation of the AMD-645 Peripheral Bus Controller. For a description of the AMD-640 System Controller, see the *AMD-640 System Controller Data Sheet*, order# 21090. Key features of the AMD-645 Peripheral Bus Controller are provided in this section.

1.1 Enhanced IDE Controller

- Enhanced master mode PCI IDE controller with Ultra DMA-33 support
- Dual channel master mode PCI supporting four enhanced IDE devices
- Transfer rate up to 33 Mbytes per second to cover PIO mode 4 and multi-word DMA mode 2 drivers, and Ultra DMA-33/ ATA-33 interface
- Sixteen levels (doublewords) of prefetch and write buffers
- Interlaced commands between the two channels
- Bus master programming interface for SFF-8038i, rev. 1.0 and Microsoft Windows 95® compliance
- Full scatter-gather capability
- Supports ATAPI-compliant devices
- Supports PCI native and ATA compatibility modes
- Complete software driver support

1.2 Universal Serial Bus Controller

- USB v. 1.0 and Intel Universal HCI v. 1.1-compatible
- Eighteen-level (doubleword) data FIFOs
- Root hub and two function ports with built-in physical layer transceivers
- Legacy keyboard and PS/2 mouse support

1.3 Plug-N-Play Support

- PCI interrupts steerable to any interrupt channel
- Three steerable interrupt channels and DMA signal steering with Plug-N-Play control
- Microsoft Windows 95 and Plug-N-Play BIOS compliant

1.4 Sophisticated Power Management

- Supports both ACPI (Advanced Configuration and Power Interface) and legacy (APM) power management
- ACPI v.0.9 Compliant
- APM v.1.2 Compliant
- Supports soft-off and power-on suspend with hardware automatic wakeup
- One idle timer, one peripheral timer, and one general purpose timer, plus 24- and 32-bit APCI-compliant timer
- Dedicated input pin for external modem ring indicator for system wakeup
- Normal, doze, sleep, suspend, and conserve modes
- System event monitoring with two event classes
- Five multipurpose I/O pins plus support for up to 16 general purpose input ports and 16 output ports
- Primary and secondary interrupt differentiation for individual channels
- Clock throttling control
- Multiple internal and external $\overline{\text{SMI}}$ sources for flexible power management

1.5 PC97-Compliant PCI-to-ISA Bridge

- Dual cascaded AT-compatible 8259 interrupt controllers
- AT-compatible 8255 programmable interval timer
- Dual AT-compatible 8237 DMA controllers
- Distributed DMA support for ISA legacy DMA across the PCI bus
- Integrated keyboard controller with PS/2 mouse support
- Integrated real-time clock with extended 256-byte CMOS RAM
- PCI v. 2.1-compliant interface
- Eight double-word line buffer between PCI and ISA bus
- Supports type F DMA transfers
- Fast reset and gate A20 operation
- Edge-triggered or level-sensitive interrupts
- Flash, 2-Mbyte EPROM, and combined BIOS support
- Programmable ISA bus clock
- Supports external IOAPIC interface with symmetrical multiprocessor configurations

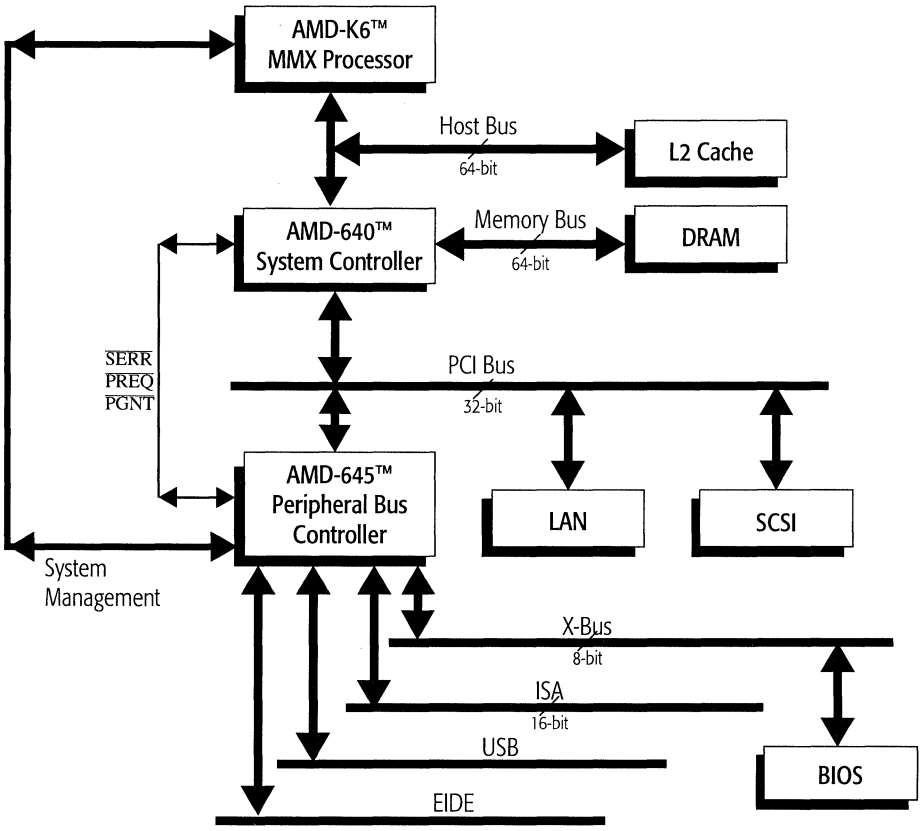


Figure 1-1. AMD-640 Chipset System Block Diagram

2 Overview

The AMD-645 Peripheral Bus Controller is responsible for converting between PCI and ISA bus cycles. The AMD-645 Peripheral Bus Controller PCI-to-ISA bridge contains eight double-word line buffers and supports Type F DMA transfers and delay transactions to streamline PCI bus operation and comply with PCI Specification version 2.1.

The AMD-645 Peripheral Bus Controller also integrates many AT-compatible and system control functions, including a keyboard controller with PS/2 mouse support, real-time clock with extended 256-byte CMOS RAM, master mode EIDE controller with full scatter and gather capability, and a USB interface with root hub and two function ports with built-in physical layer transceiver.

2.1 PCI-to-ISA Bridge

The AMD-645 Peripheral Bus Controller offers both a PCI-compatible bus interface and an ISA-compatible interface. These interfaces, which are fully compliant with the PCI 2.1 specification, control PCI/ISA bus communication. Two main blocks, the PCI bus master and slave blocks, make up the PCI interface control.

To become PCI bus master, the AMD-645 Peripheral Bus Controller must arbitrate for control of the bus with the AMD-640 System Controller. Once bus ownership has been granted, the AMD-645 Peripheral Bus Controller assumes PCI bus master responsibility. The AMD-645 Peripheral Bus Controller is in slave mode whenever it does not own the PCI bus.

2.1.1 PCI Bus Master Mode

The AMD-645 Peripheral Bus 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 mode, the data transfer takes place either between two ISA bus resources or between an ISA and a PCI bus resource.

To determine the destination of the bus master request, the AMD-645 Peripheral Bus Controller can sample an active $\overline{\text{DEVSEL}}$ input, which indicates that a particular target on the PCI bus is responding to the current request. The destination can also be determined by a positive decoding of the master-driven address.

A third alternative for determining the destination is subtractive decode. If the destination is not identified by either positive address decoding or an active $\overline{\text{DEVSEL}}$ input, the AMD-645 Peripheral Bus Controller assumes the access is occurring only between two ISA bus resources.

The AMD-645 Peripheral Bus Controller PCI interface translates all non-positive decoded ISA master requests to the PCI bus. In situations in which the request is forwarded to the PCI bus, the AMD-645 Peripheral Bus Controller ensures ISA and PCI bus synchronization by controlling the ISA-based IOCHRDY signal. If an active $\overline{\text{DEVSEL}}$ response is not received within the specified time, the AMD-645 Peripheral Bus Controller master interface assumes the requested cycle was between ISA resources and executes a PCI master abort cycle. In the event the $\overline{\text{DEVSEL}}$ signal is sampled active within the specified time, the AMD-645 Peripheral Bus Controller master interface executes a data transfer between the ISA and PCI buses.

2.1.2 PCI Bus Slave Mode

The AMD-645 Peripheral Bus Controller stays in PCI slave mode when it does not own the bus. The slave interface responds to any request from a PCI resource by asserting $\overline{\text{DEVSEL}}$ if it has positively decoded the current address as a destination for either the ISA bus or for on-chip I/O.

When the current address is not positively decoded, the AMD-645 Peripheral Bus Controller target interface is de-selected by an active $\overline{\text{DEVSEL}}$ input driven by another PCI resource.

If no active DEVSEL signal is received within a specified time, the AMD-645 Peripheral Bus Controller 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-645 Peripheral Bus Controller responds to requests destined for the ISA bus or on-chip I/O by executing a single data transfer and signalling a target disconnect. If the AMD-645 Peripheral Bus Controller samples an active DEVSEL input within a specified time, it is de-selected, allowing the transfer to take place between the two PCI resources.

The AMD-645 Peripheral Bus Controller is capable of posting PCI-to-ISA memory write cycles. When posting is enabled, the PCI request is acknowledged immediately and the write data is latched to allow the ISA cycle to proceed independently from the PCI transaction.

2.2 ISA Controller

The integrated ISA system address latches and control logic allow the AMD-645 Peripheral Bus Controller user to design an extremely cost-effective system. In addition, the AMD-645 Peripheral Bus Controller contains the decode logic to select an external keyboard controller. This keyboard controller can be programmed for attachment on either the XD or SD bus.

The AMD-645 Peripheral Bus Controller controls accesses to 8-bit BIOS ROM and to 8-bit or 16-bit ISA bus ROM. The BIOS ROM must be 8 bits and is accessed via an external XD bus. All other ROM is accessed as either 8-bit or 16-bit ROM residing on the ISA SD bus, either on-board or off-board via the slots. Accesses in the C0000h–CFFFFh and E0000h–EFFFFh ranges are optionally definable as on-board system ROM or off-board memory via the ROM relocation register. A special mode is supported for erasing and programming flash memories in areas where such devices are used as the BIOS ROMs.

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 over a 24-bit (16- Mbyte) address range. Internal latches latch the middle address bits output by the 8237A megacells. A memory mapper generates the upper address bits.

As specified by the industry standard, distributed DMA offers support for seven DMA channels. The distributed DMA logic remaps I/O cycles from the distributed I/O target locations to the applicable DMA controller. When this remapping is enabled, accesses to the legacy DMA I/O addresses are disabled and ISA cycles are generated instead. DMA requests from the ISA bus that address PCI memory cause PCI master requests and cycles to be generated by the AMD-645 Peripheral Bus Controller.

The AMD-645 Peripheral Bus Controller generates synchronous ISA bus timing and synchronous IDE interface timing from the 33-MHz PCI bus clock.

The AMD-645 Peripheral Bus Controller performs all the data steering functions between the ISA bus and the PCI bus. 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 into several ISA bus reads, the data returned by the ISA devices is assembled by latches before being returned to the PCI bus. The AMD-645 Peripheral Bus Controller also performs low-to-high and high-to-low byte swaps on the 16-bit SD bus.

As a PCI slave, the AMD-645 Peripheral Bus Controller is capable of expanding PCI accesses with non-contiguous byte enables into the appropriate discrete ISA cycles.

The AMD-645 Peripheral Bus Controller functions are programmable via a set of internal device-specific configuration registers. The state of various interface pins on reset is used to determine the default configuration.

2.3 EIDE Controller

The AMD-645 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 supports 32-bit PCI data transfers. The IDE-PCI interface operates at PCI speed and allows concurrent IDE and PCI operations to maximize system performance.

Logically, the IDE drive interface can be viewed as being composed of six controller blocks.

CPU Command Processor

The CPU command processor receives input commands from the CPU, FIFO full/FIFO empty signals from the write-FIFO, and read-ahead full signals from the read-ahead buffer.

I/O Processor

The I/O processor is the IDE control signal block, containing all of the IDE bus control logic. It receives inputs from the IDE bus, command processor, and write FIFO. The I/O processor issues the $\overline{\text{IOR}}$/$\overline{\text{IOW}}$ signals to the IDE bus, based on programmed address setup time, $\overline{\text{IOR}}$/$\overline{\text{IOW}}$ precharge time, and $\overline{\text{IOR}}$/$\overline{\text{IOW}}$ active duration. It also translates 16-bit cycles to two 8-bit cycles when necessary.

Write Buffer

The write buffer takes 32-bit CPU data and converts it to the proper 16-bit or 8-bit data format.

Read-Ahead/Posted-Write FIFO

This block functions as a read-ahead buffer during read accesses from I/O address 1F0h. During writes, this block stores 16-bit data in the 16-doubleword FIFO and passes control to the I/O processor or DMA state machine. Its direction is determined by commands and register programming.

IDE Arbiter

The arbiter arbitrates between IDE channels and multiplexes the IDE data bus, IDE address, and IDE chip selects.

DMA State Machine

The DMA bus mastering state machine controls $\overline{\text{IOW}}$ and $\overline{\text{IOR}}$ pulses for each IDE channel during DMA accesses.

The AMD-645 Peripheral Bus Controller's enhanced IDE controller provides a data path and control interface to standard IDE drives. The block is fully 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 channels are supported with the ability to connect to both with no external logic. Data is transferred over a shared 16-bit IDE data bus.

The AMD-645 Peripheral Bus Controller contains two IDE interfaces. Channel 0 is the primary interface, with target I/O addresses at 1F0h–1F7h and 3F6h. Its IRQ pin is mapped to IRQ14. Channel 1 is the secondary IDE interface, with target I/O addresses at 170h–177h and 376h. Its IRQ pin is mapped to IRQ15. Unless otherwise noted, discussions in this document referring to channel 0 resources apply equally to the respective channel 1 resources.

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 Function 1, 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. Independent configuration registers exist in PCI configuration space for each channel.

2.4 Universal Serial Bus

The AMD-645 Peripheral Bus Controller USB host controller interface is fully compatible with both the USB specification v.1.0 and the Intel Universal HCI specification v.1.1. There are two sets of software-accessible registers, the PCI configuration registers and the USB I/O registers.

The interface supports eighteen levels (doublewords) of data FIFOs, and a root hub and two function ports with built-in physical layer transceivers. The USB controller allows hot Plug-N-Play and isochronous peripherals to be inserted into the system with universal driver support.

In addition, the AMD-645 Peripheral Bus Controller offers legacy (X-bus) keyboard and PS/2 mouse support.

2.5 Power Management

The AMD-645 Peripheral Bus Controller supports Advanced Configuration and Power Interface (ACPI) as well as legacy Advanced Power Management (APM). It complies with both ACPI specification v.0.9 and APM specification v.1.2. In addition, AMD-645 Peripheral Bus Controller power management is compatible with PC97 and OnNow.

The real-time clock with 256-byte extended CMOS includes a data alarm and other enhancements for compatibility with the ACPI standard. Two types of sleep states are provided, soft-off and power-on suspend, along with hardware automatic wake-up. Additional power management features includes event monitoring, CPU clock throttling, hardware and software-based event handling, general purpose IO, and external SMI.

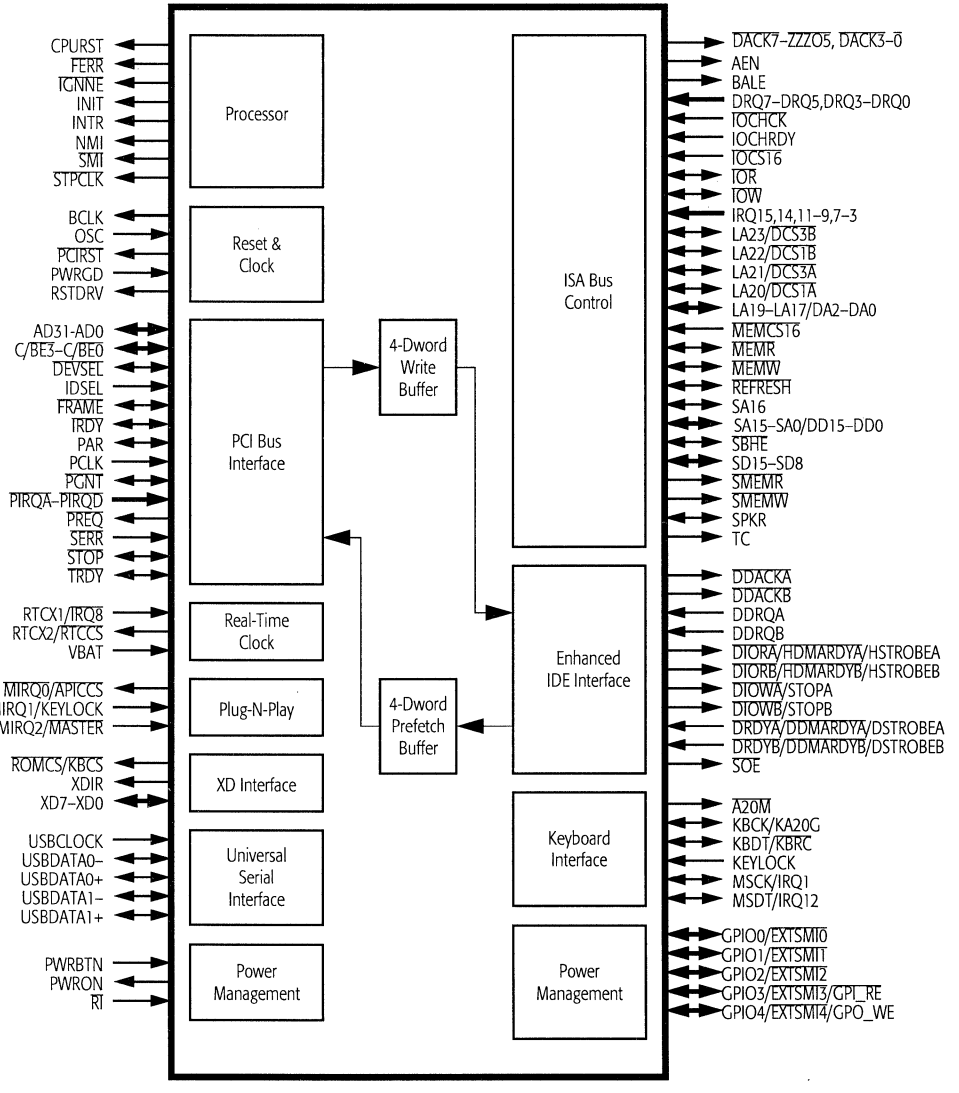


Figure 2-1. AMD-645 Peripheral Bus Controller Block Diagram

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 3-1. Valid Combinations

OPN	Package Type	Operating Voltage	Case Temperature
AMD-645	208-pin PQFP	4.75 V–5.25 V	70°C

Note:
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 PCI Bus Interface

AD31–AD0 — ***PCI Address/Data Bus*** — ***Bidirectional***

AD31–AD0 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.

When the AMD-645 Peripheral Bus Controller is PCI master, these lines are outputs during the address and write data phases of a transaction, and inputs during the read data phase.

When the AMD-645 Peripheral Bus Controller is PCI slave, these lines are inputs during the address and write data phases of a transaction, and outputs during the read data phase.

C3/BE3–C0/BE0 — ***PCI Command/Byte Enable*** — ***Bidirectional***

During the first clock of a PCI transaction, when FRAME is asserted, these lines contain the PCI bus command (C3–C0). On subsequent clocks, these lines contain PCI byte enables (BE3–BE0) corresponding to supplied or requested data.

C3/BE3–C0/BE0 are outputs when the AMD-645 Peripheral Bus Controller is the PCI bus master. They are inputs when it is the slave.

DEVSEL — ***PCI Bus Device Select*** — ***Bidirectional***

When the AMD-645 Peripheral Bus Controller is PCI bus master, DEVSEL is an input that determines 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-645 Peripheral Bus Controller aborts the PCI bus cycle.

When the AMD-645 Peripheral Bus Controller is not PCI bus master it defaults to target mode, and DEVSEL is an output indicating that it claims a PCI transaction through either positive or subtractive decoding. In a positive decode, the AMD-645 Peripheral Bus Controller asserts DEVSEL one PCLK cycle after FRAME is sampled active and holds it Low through the end of the transaction. In a subtractive decode, DEVSEL is driven Low three PCLK cycles after FRAME is asserted. Positive and negative decoding are explained in Section 5.1 on page 5-1.

FRAME ***PCI Bus Cycle Frame*** ***Bidirectional***

The assertion of FRAME indicates the address phase of a PCI transfer, while its negation indicates that one more data transfer is desired by the cycle initiator. While FRAME is asserted, data transactions can continue. When FRAME is deasserted, data transactions are in the final phase.

When the AMD-645 Peripheral Bus Controller is PCI bus master, FRAME is driven active for one clock cycle to start the current bus cycle. When the AMD-645 Peripheral Bus Controller is the slave, FRAME is an input indicating the beginning and duration of the current bus cycle.

IDSEL ***PCI Initialization Device Select*** ***Input***

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

IRDY ***PCI Bus Initiator Ready*** ***Bidirectional***

IRDY is asserted by a PCI initiator from the first clock cycle after FRAME to the last clock of the transaction to indicate it is ready for data transfer.

When the AMD-645 Peripheral Bus Controller is PCI master, IRDY is an output that indicates the ability of the chip to complete the current data phase of the transaction. When the AMD-645 Peripheral Bus Controller is PCI slave, a read cycle cannot end and the write cycle cannot start until the IRDY input is sampled active.

PAR ***PCI Bus Parity*** ***Bidirectional***

This signal provides even parity for AD31–AD0 and C3/$\overline{BE3}$–C0/$\overline{BE0}$. When the AMD-645 Peripheral Bus Controller is PCI bus master, it drives PAR one PCLK after the address and write data phases.

When the AMD-645 Peripheral Bus Controller is PCI slave, it samples the PAR input one clock after a read is completed.

PCIRST ***PCI Reset*** ***Output***

$\overline{PCIRST}$ is an active Low reset signal for the PCI bus. The AMD-645 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.

PCLK ***PCI Bus Clock*** ***Input***

PCLK provides timing for all transactions on the PCI bus. It runs at half the CPU frequency, up to 33 MHz. PCLK can also be divided down to generate the ISA bus clock.

PGNT ***PCI Grant*** ***Input***

The AMD-640 System Controller drives $\overline{PGNT}$ to grant PCI bus access to the AMD-645 Peripheral Bus Controller.

$\overline{PIRQA}$–$\overline{PIRQD}$ ***PCI Interrupt Requests*** ***Input***

These pins are typically connected to the PCI bus INT lines as shown in Table 4-1.

Table 4-1. Connecting PIRQ Lines to PCI INT Lines

	$\overline{PIRQA}$	$\overline{PIRQB}$	$\overline{PIRQC}$	$\overline{PIRQD}$
PCI Slot 1	$\overline{INTA}$	$\overline{INTB}$	$\overline{INTC}$	$\overline{INTD}$
PCI Slot 2	$\overline{INTB}$	$\overline{INTC}$	$\overline{INTD}$	$\overline{INTA}$
PCI Slot 3	$\overline{INTC}$	$\overline{INTD}$	$\overline{INTA}$	$\overline{INTB}$
PCI Slot 4	$\overline{INTD}$	$\overline{INTA}$	$\overline{INTB}$	$\overline{INTC}$

$\overline{PREQ}$ ***PCI Request*** ***Output***

The AMD-645 Peripheral Bus Controller asserts $\overline{PREQ}$ to request control of the PCI bus.

SERR — ***System Error*** — ***Input***

Any PCI device that detects a system error condition can alert the system by asserting SERR for one PCI clock. The AMD-645 Peripheral Bus Controller can be programmed to generate an NMI to the CPU if it samples SERR active.

STOP — ***Stop*** — ***Bidirectional***

A PCI target asserts STOP to request that the master stop the current transaction. When the AMD-645 Peripheral Bus Controller is PCI master, STOP is an input that causes the AMD-645 Peripheral Bus Controller to terminate the transfer and abort or retry it depending on the state of DEVSEL and TRDY.

When the AMD-645 Peripheral Bus Controller is PCI slave, it asserts STOP and TRDY simultaneously to indicate a target disconnect following the data transfer or burst. It does not assert STOP if the transfer is a single, non-bursted transfer.

TRDY — ***PCI Target Ready*** — ***Bidirectional***

A PCI target asserts TRDY when it is ready for data transfer. When the AMD-645 Peripheral Bus Controller is the PCI master, TRDY is an input that indicates the ability of the target device to complete the data phase of the transaction. Once a PCI bus transaction is initiated, the AMD-645 Peripheral Bus Controller inserts wait cycles until TRDY is sampled active.

As the PCI slave, the AMD-645 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.

4.2 ISA Bus Interface

AEN — ***Address Enable*** — ***Output***

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. It is asserted only when the DMA controller is the bus owner.

BALE — ***Bus Address Latch Enable*** — ***Output***

BALE is asserted for a bus clock at the beginning of any bus cycle initiated by a PCI master. It is asserted by the AMD-645 Peripheral Bus Controller to indicate that the address signal lines (SA19–SA0, LA23–LA17, and $\overline{\text{SBHE}}$) are valid.

BCLK — ***Bus Clock*** — ***Output***

BCLK is the ISA bus clock.

$\overline{\text{DACK7}}$–$\overline{\text{DACK5}}$, $\overline{\text{DACK3}}$–$\overline{\text{DACK0}}$ — ***DMA Acknowledge*** — ***Output***

These lines indicate that the corresponding request for DMA service has been accomplished.

DRQ7–DRQ5, DRQ3–DRQ0 — ***DMA Request*** — ***Input***

These asynchronous DMA request lines are used by external devices to request services from the AMD-645 Peripheral Bus Controller DMA controller. DRQ3–DRQ0 are used for transfers between 8-bit I/O adapters and system memory. DRQ7–DRQ5 are used for transfers between 16-bit I/O adapters and system memory. DRQ4 is not available externally.

$\overline{\text{IOCHCK}}$ — ***I/O Channel Check*** — ***Input***

$\overline{\text{IOCHCK}}$ 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, the AMD-645 Peripheral Bus Controller generates an NMI to the processor when it samples $\overline{\text{IOCHCK}}$ asserted.

IOCHRDY — ***I/O Channel Ready*** — ***Input***

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 CPU, DMA controllers, or refresh controller. The AMD-645 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 CPU is as follows:

- 8-bit peripherals 4 wait states
- 16-bit peripherals 1 wait state

- ROM cycles 3 wait states

One DMA wait state is inserted as the default for all DMA cycles. 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-645 Peripheral Bus Controller always drives IOCHRDY Low in either DMA or Master Mode to allow for PCI bus latency.

$\overline{\text{IOCS16}}$ ***16-Bit I/O Chip Select*** ***Input***

$\overline{\text{IOCS16}}$ is driven by I/O devices on the ISA bus to indicate that they support 16-bit I/O bus cycles.

The AMD-645 Peripheral Bus Controller samples $\overline{\text{IOCS16}}$ to determine when a CPU access requires a 16-bit to 8-bit conversion. It also performs a conversion if it requests a 16-bit I/O cycle and samples $\overline{\text{IOCS16}}$ High. In a conversion, the AMD-645 Peripheral Bus Controller inserts a command delay of one bus cycle and the cycle becomes four wait states long. If $\overline{\text{IOCS16}}$ is sampled Low, the AMD-645 Peripheral Bus Controller performs an I/O access in one wait state, inserting one command delay.

$\overline{\text{IOR}}$ ***I/O Read*** ***Bidirectional***

$\overline{\text{IOR}}$ is the command to an ISA I/O slave device indicating the slave can drive data onto the ISA data bus.

$\overline{\text{IOR}}$ is an input when the AMD-645 Peripheral Bus Controller is bus master and an output at all other times. When the AMD-645 Peripheral Bus Controller is a PCI slave, $\overline{\text{IOR}}$ is driven by the internal ISA bus controller.

During DMA transfers, $\overline{\text{IOR}}$ is driven by the DMA controller. It is inactive during a refresh cycle.

$\overline{\text{IOW}}$ ***I/O Write*** ***Bidirectional***

$\overline{\text{IOW}}$ is the command to an ISA I/O slave device indicating the slave can latch data from the ISA data bus.

IOW is an input when the AMD-645 Peripheral Bus Controller is bus master and an output at all other times. When the AMD-645 Peripheral Bus Controller is a PCI slave, IOW is driven by the internal ISA bus controller.

During DMA transfers, IOW is driven by the DMA controller. It is inactive during a refresh cycle.

IRQ15, IRQ14, IRQ11–IRQ9, IRQ7–IRQ3

Interrupt Request ***Input***

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

LA23/DCS3B, LA22/DCS1B, LA21/DCS3A, LA20/DCS1A, LA19–LA17/ DA2–DA0

Multifunctional Pins ***Bidirectional***

ISA Bus Cycles—*Unlatched Address*
The LA23–LA17 address lines are bidirectional and allow accesses to physical memory on the ISA bus up to 16 Mbytes.

PCI IDE Cycles—*Chip Select*
DCS1A, DCS3A, DCS1B and DCS3B are for the ATA command register block and correspond directly to CS1FX, CSF3X, CS17X, and CS37X on the primary IDE connector, respectively.

PCI IDE Cycles—*Disk Address*
DA2–DA0 are used to indicate which byte in either the ATA command or control block is being accessed.

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. The LA pins are outputs when MASTER is High and are inputs when it is Low.

MASTER / MIRQ2

Multifunctional Pin ***Input***

ISA Master Cycle Indicator
An external bus master device asserts MASTER to indicate that it has control of the bus.

Plug and Play Interrupt Request 2
MIRQ2 is a steerable interrupt request for on-board devices.

MEMCS16 — ***16-Bit Memory Chip Select*** — ***Input***

ISA 16-bit slave memory devices drive this line Low to indicate support for 16-bit memory bus cycles. This line is sampled to determine when a 16-bit to 8-bit conversion is needed for CPU accesses. Conversion is performed when the AMD-645 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.

MEMR — ***Memory Read*** — ***Bidirectional***

MEMR is the command to a memory slave that permits it to drive data onto the ISA data bus. This signal is an input when an external bus master is in control and an output at all other times.

MEMW — ***Memory Write*** — ***Bidirectional***

MEMW is the command to a memory slave that permits it to latch data from the ISA data bus. This signal is an input when an external bus master is in control and an output at all other times.

REFRESH — ***Refresh*** — ***Bidirectional***

As an output, REFRESH indicates a refresh cycle is in progress. It is asserted by the AMD-645 Peripheral Bus Controller whenever a refresh cycle is initiated. As an input, REFRESH is driven by 16-bit ISA bus masters to indicate a refresh cycle.

RSTDRV — ***Reset Drive*** — ***Output***

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.

SA15–SA0/ DD15–DD0 — ***System Address Bus/IDE Data Bus*** — ***Bidirectional***

These pins serve as the address bus in ISA operation and the data bus in IDE operation.

SA16 — ***System Address Bus*** — ***Bidirectional***

This signal is ISA address bit 16.

SBHE — ***System Byte High Enable*** — ***Bidirectional***

When asserted, $\overline{\text{SBHE}}$ indicates that a byte is being transferred on the upper byte of the ISA data bus (SD15–SD8). $\overline{\text{SBHE}}$ is negated during refresh cycles.

SD15–SD8/ GPI15–GPI8 GPO15–GPO8 — ***Multifunctional Pins*** — ***Bidirectional***

ISA System Data
SD15–SD8 provide the high order data path for devices residing on the ISA bus.

General-Purpose Inputs
If the GPIO3_CFG bit is cleared (Function 3, offset 40h, bit 6), these pins function as GPI15–GPI8 and pin 92 serves as read enable $\overline{\text{GPI\_RE}}$.

General-Purpose Outputs
If the GPIO4_CFG bit is cleared (Function 3, offset 40h, bit 7), these pins function as GPO15–GPO8 and pin 92 serves as write enable GPI_WE.

$\overline{\text{SMEMR}}$ — ***Standard Memory Read*** — ***Output***

$\overline{\text{SMEMR}}$ is the command that permits a slave to drive data residing below the 1 MByte region onto the ISA data bus.

$\overline{\text{SMEMW}}$ — ***Standard Memory Write*** — ***Output***

$\overline{\text{SMEMW}}$ is the command that permits a slave to latch data residing below the 1 MByte region from the ISA data bus.

TC — ***Terminal Count*** — ***Output***

The AMD-645 Peripheral Bus Controller asserts TC to DMA slaves to indicate that one of the DMA channels has transferred all data.

SPKR/ Power-Up Strap — ***Multifunction Pin*** — ***Bidirectional***

Speaker Drive
After reset, this pin functions as the SPKR signal, which is the output of counter 2.

Power-Up Strapping
At reset, if this pin is strapped Low, the IDE I/O base is fixed. If the pin is strapped High, the IDE I/O base is flexible.

4.3 Ultra DMA-33 Enhanced IDE Interface

Note: The IDE address, data, and drive select pins are multiplexed with the ISA bus LA and SA pins and are described in Section 4.2.

DDACKA — ***Disk DMA Acknowledge A*** — ***Output s***

DDACKA is the primary IDE channel DMA acknowledge. The AMD-645 Peripheral Bus Controller responds to DDRQA either to acknowledge that data has been accepted or to inform that data is available.

DDACKB — ***Disk DMA Acknowledge B*** — ***Output***

DDACKB is the secondary IDE channel DMA acknowledge. The AMD-645 Peripheral Bus Controller responds to DDRQB either to acknowledge that data has been accepted or to inform that data is available.

DDRQA — ***Device DMA Request A*** — ***Input***

DDRQA is the primary IDE channel DMA request. A device asserts DDRQA when it is ready to read or write DMA data.

DDRQB — ***Device DMA Request B*** — ***Input***

DDRQB is the secondary IDE channel DMA request. A device asserts DDRQB when it is ready to read or write DMA data.

DIORA/ HDMARDYA/ HSTROBEA — ***Multifunction Pin*** — ***Output***

EIDE Mode—*Device I/O Ready A*
DIORA is the primary IDE channel drive write strobe. The falling edge of DIORA enables the transfer of data from a register or data port of the drive onto the IDE data bus, DD15–DD0. The rising edge of DIORA latches the data.

Ultra DMA Mode—*Host DMA Ready*
HDMARDYA functions as the primary channel input flow control. The host can assert HDMARDYA to pause input data transfers.

Ultra DMA Mode—*Host Strobe A*
HSTROBEA functions as the primary channel output strobe. The host can stop HSTROBEA to pause output data transfers.

DIORB/ HDMARDYB/ HSTROBEB

Multifunction Pin ***Output***

EIDE Mode—*Device I/O Ready B*
DIORB is the secondary IDE channel drive write strobe. The falling edge of DIORB enables the transfer of data from a register or data port of the drive onto the IDE data bus, DD15–DD0. The rising edge of DIORB latches the data.

Ultra DMA Mode—*Host DMA Ready B*
HDMARDYB functions as the secondary channel input flow control. The host can assert HDMARDYB to pause input data transfers.

Ultra DMA Mode—*Host Strobe B*
HSTROBEB functions as the secondary channel output strobe. The host can stop HSTROBEB to pause output data transfers.

DIOWA/ STOPA

Multifunction Pin ***Output***

EIDE Mode—*Device I/O Write A*
DIOWA is the primary IDE channel drive read strobe. The rising edge of DIOWA clocks data from the IDE data bus (DD15–DD0) into either a register or the data port of the drive.

Ultra DMA Mode—*Stop A*
STOPA halts data transfer in the primary channel. The host asserts STOPA before an Ultra DMA burst is initiated and negates STOPA before an Ultra DMA burst is transferred. The host asserts STOPA during or after data transfer in Ultra DMA mode to signal the termination of the burst.

DIOWB/ STOPB

Disk I/O Write B ***Output***

EIDE Mode—*Device I/O Write B*
DIOWB is the secondary IDE channel drive write strobe. The rising edge of DIOWA clocks data from the IDE data bus (DD15–DD0) into either a register or the data port of the drive.

Ultra DMA Mode—*Stop B*
STOPB halts data transfer in the secondary channel. The host asserts STOPB before an Ultra DMA burst is initiated and negates STOPB before an Ultra DMA burst is transferred. The host asserts STOPB during or after data transfer in Ultra DMA mode to signal the termination of the burst.

DRDYA/ DDMARDYA/ DSTROBEA

Multifunction Pin ***Input***

EIDE Mode—*Device Ready A*
$\overline{\text{DRDYA}}$ is the primary channel device ready indicator. A device negates $\overline{\text{DRDYA}}$ to extend the AMD-645 Peripheral Bus Controller read or write cycle when it is not ready to respond to a data transfer request. When $\overline{\text{DRDYA}}$ is negated, it is in a high impedance state.

Ultra DMA Mode—*Device DMA Ready A*
$\overline{\text{DDMARDYA}}$ is the primary channel output flow control. A device can assert $\overline{\text{DDMARDYA}}$ to pause output transfers.

Ultra DMA Mode—*Device Strobe A*
DSTROBEA is the primary channel input data strobe. A device can stop DSTROBEA to pause input data transfers.

DRDYB/ DDMARDYB/ DSTROBEB

Multifunction Pin ***Input***

EIDE Mode—*Device Ready B*
$\overline{\text{DRDYB}}$ is the secondary channel device ready indicator. A device negates $\overline{\text{DRDYB}}$ to extend the AMD-645 Peripheral Bus Controller read or write cycle when it is not ready to respond to a data transfer request. When $\overline{\text{DRDYB}}$ is negated, it is in a high impedance state.

Ultra DMA Mode—*Device DMA Ready B*
$\overline{\text{DDMARDYB}}$ is the primary channel output flow control. A device can assert $\overline{\text{DDMARDYB}}$ to pause output transfers.

Ultra DMA Mode—*Device Strobe B*
DSTROBEB is the primary channel input data strobe. A device can stop DSTROBEB to pause input data transfers.

SOE — ***System Address Transceiver Output Enable Output***

$\overline{SOE}$ controls the output enables of the 74F245 transceivers that interface the IDE data bus (DD15–DD0) to the system address bus (SA15–SA0). $\overline{MASTER}$ drives the transceiver direction control with DD15–DD0 connected to the "A" side of the transceivers and SA15–SA0 connected to the "B" side.

4.4 XD Bus Interface

$\overline{ROMCS}$/$\overline{KBCS}$ — ***Multifunctional Pin*** ***Output***

ROM Chip Select
In ISA memory cycles, $\overline{ROMCS}$ is the chip select to the ROM BIOS.

Keyboard Chip Select
In ISA I/O cycles, $\overline{KBCS}$ is the chip select to the external keyboard controller.

XD7–XD0/ EXTSMI7–EXTSMI3/ GPI7–GPI0/ GPO7–GPO0/ Power-Up Straps — ***Multifunction Pins*** ***Bidirectional***

XD7–XD0
Connection to external X-bus devices such as BIOS ROM.

EXTSMI7–EXTSMI3
External SCI/SMI ports.

GPI7–GPI0
General-purpose inputs if configuration register Function 3, offSet 40h, bit 6 is cleared.

GPO7–GPO0
General-purpose outputs if configuration register Function 3, offset 40h, bit 7 is cleared.

Power-Up Straps
Pins XD7–XD0 are used as strap options during power-up (see configuration register Function 0, offset 5Ah on page 7-27). Strapping Low disables and strapping High enables the following functions:

- XDO internal KBC
- XD1 internal PS/2 Mouse
- XD2 internal RTC
- XD4–XD7 RP13–RP16 for internal KBC

XDIR ***X-Bus Data Direction*** ***Output***

XDIR is tied directly to the direction control of the 74F245 transceiver that buffers the X-bus data and ISA-bus data. SD0–SD7 connect to the "A" side of the transceiver and XD0–SD7 connect to the "B" side. The output enable of the transceiver should be grounded. A High signal on SDIR indicates that SD0–SD7 drives XD0–XD7.

4.5 Plug-N-Play Support

The AMD-645 Peripheral Bus Controller provides three interrupt request pins to support Plug-n-Play functions from non-PnP devices. These asynchronous interrupt requests are mappable to any of the interrupt channels. Each pin has an alternate function which is selected in configuration register Function 0, offset 59h (see page 7-26).

MIRQ2/MASTER ***Multifunction Pin*** ***Input***

Plug-n-Play—*Interrupt Request 2*

ISA—*Master Cycle Indicator* (see page 4-7)

MIRQ1/KEYLOCK ***Multifunction Pin*** ***Input***

Plug-n-Play—*Interrupt Request 1*

KEYLOCK—*Keyboard Lock Input*

MIRQQ0/APICCS ***Multifunction Pin*** ***Input***

Plug-n-Play—*Interrupt Request 0*

APICCS—*APIC Chip Select*
This signal is provided for external IO APIC devices in symmetric multiprocessor implementations.

4.6 Universal Serial Bus Interface

USBCLK	***Universal Serial Bus Clock***	***Input***
USBDATA0+	***USB Port 0 Data +***	***Bidirectional***
USBDATA0–	***USB Port 0 Data –***	***Bidirectional***
USBDATA1+	***USB Port 1 Data +***	***Bidirectional***
USBDATA1–	***USB Port 1 Data –***	***Bidirectional***

4.7 Power Management

PWRBTN — ***Power Button*** — ***Input***

Referenced to V_{DD}-5VSB.

PWRGD — ***Power Good*** — ***Input***

PWRGD is connected to the POWERGOOD signal on the power supply.

PWRON — ***Power Supply Control*** — ***Output***

Powered by V_{DD}-5VSB.

RI — ***Ring Indicator*** — ***Input***

This signal can be connected to external modem circuitry to allow the system to be reactivated by a received phone call. Input referenced to V_{DD}-5VSB.

4.8 Power and Ground

AGND — ***USB Differential Output Ground*** — ***Power***

A_{VDD} — ***USB Differential Output Power Source***

$V_{DD}3$ — ***Power Supply for the CPU I/O VoltagePower***

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

V_{DD} ***Power Supply of 4.75 V to 5.25 V*** ***Power***

This supply is turned on only when the mechanical switch on the power supply is turned on and the PWRON signal is conditioned high.

V_{DD}-5SB ***Power Supply*** ***Power***

V_{DD}-5SB is always available unless the mechanical switch of the power supply is turned off. If the "soft-off" state is not implemented, then this pin can be connected to V_{DD}.

V_{DD}-PCI ***PCI Voltage, 3.3 V or 5 V*** ***Power***

GND ***Ground*** ***Power***

4.9 Internal Real-Time Clock

OSC ***Oscillator*** ***Input***

OSC is a 14.31818-MHz clock used by the internal Real-Time Clock (RTC).

RTCX1/$\overline{\text{IRQ8}}$ ***Multifunctional Pin*** ***Input***

RTCX1
When the internal RTC is enabled, this signal is the RTC crystal or oscillator input (32.768 KHz.)

IRQ8
When the internal RTC is disabled, $\overline{\text{IRQ8}}$ is an input from an external keyboard controller.

RTCX2/RTCCS ***Multifunctional Pin*** ***Output***

RTCX2
When the internal RTC is enabled, this signal is the RTC crystal or oscillator output (32.768 KHz.)

RTCCS
When the internal RTC is disabled, this signal is the External RTC chip select.

VBAT ***RTC Battery*** ***Input***

This signal is the battery input for internal RTC.

4.10 Keyboard Interface

$\overline{\text{A20M}}$ — ***A20 Mask*** — ***Output***

The AMD-645 Peripheral Bus Controller $\overline{\text{A20M}}$ is a direct connection to $\overline{\text{A20M}}$ on the CPU.

KBCK/KA20G — ***Multifunctional Pin*** — ***Bidirectional***

Keyboard Clock
When the internal keyboard controller is enabled, KBCK is the clock to the keyboard interface.

Keyboard Gate A20
When the internal keyboard controller is disabled, KA20G is the Gate A20 output from the external keyboard controller.

KBDT/$\overline{\text{KBRC}}$ — ***Multifunctional Pin*** — ***Bidirectional***

Keyboard Data
When the internal keyboard controller is enabled, KBDT is the data line to the keyboard interface.

Keyboard Reset
When the internal keyboard controller is disabled, $\overline{\text{KBRC}}$ is a reset input from the external keyboard controller.

KEYLOCK — ***Keyboard Lock*** — ***Input***

KEYLOCK is the keyboard lock signal for the internal keyboard controller.

MSCK/IRQ1 — ***Multifunctional Pin*** — ***Bidirectional***

Mouse Clock
When the PS/2 mouse is enabled, MSCK functions as the clock to the PS/2 mouse interface.

IRQ1
When both the PS/2 mouse and the internal KBC are disabled, IRQ1 functions as interrupt request 1 from the external KBC.

MSDT/IRQ12 — ***Multifunctional Pin*** — ***Bidirectional***

Mouse Data
When the PS/2 mouse is enabled, MSDT functions as data to the PS/2 mouse interface.

IRQ12
When the PS/2 mouse is disabled, IRQ12 functions as interrupt request 12 from the external KBC.

4.11 CPU Interface

CPURST — ***CPU Reset*** — ***Output***

The AMD-645 Peripheral Bus Controller asserts CPURST to reset the CPU during power-up.

FERR — ***Numerical Coprocessor Error*** — ***Output***

FERR is tied to the coprocessor error signal on the CPU.

IGNNE — ***Ignore Error*** — ***Output***

IGNNE is connected to the ignore error pin on the CPU.

INIT — ***Initialization*** — ***Output***

The AMD-645 Peripheral Bus Controller asserts INIT if it detects a shut-down special cycle on the PCI bus, or if a soft reset is initiated by the register.

INTR — ***CPU Interrupt*** — ***Output***

INTR is driven by the AMD-645 Peripheral Bus Controller to signal the CPU that an interrupt request is pending and needs service.

NMI — ***Non-Maskable Interrupt*** — ***Output***

NMI is used to force a non-maskable interrupt to the CPU. The AMD-645 Peripheral Bus Controller generates an NMI when either SERR or IOCHK is asserted.

SMI — ***System Management Interrupt*** — ***Output***

SMI is asserted by the AMD-645 Peripheral Bus Controller to alert the CPU in response to selected power management events.

STPCLK — ***Stop Clock*** — ***Output***

STPCLK is asserted by the AMD-645 Peripheral Bus Controller to the CPU in response to selected power management events.

4.12 General-Purpose I/O

GPIO0/ EXTSMI0

Multifunction Pin ***Bidirectional***

GPIO0
General-purpose I/O. This pin sits on the V_{DD}-5VSB power plane and is available in the soft-off state as well as regular operation.

EXTSMI0
An external input signal to trigger an SMI/SCI to the CPU.

GPIO1/ EXTSMI1/ I²CD1(Clock)

Multifunction Pin ***Bidirectional***

GPIO1
General-purpose I/O.

EXTSMI1
An external input signal to trigger an SMI/SCI to the CPU.

I²CD1
This pin can be used along with GPIO2 as an I^2C pair (software convention defines this pin as clock).

GPIO2/ EXTSMI2/ I²CD2(Data)

Multifunction Pin ***Bidirectional***

GPIO2
General-purpose I/O.

EXTSMI2
An external input signal to trigger an SMI/SCI to the CPU.

I²CD1
This pin can be used along with GPIO1 as an I^2C pair (software convention defines this pin as data).

GPIO3/ EXTSMI3/ GPI_RE

Multifunction Pin ***Bidirectional***

GPIO3
General-purpose I/O (if configuration register Function 3, offset 40h, bit 6 is set)

EXTSMI3
An external input signal to trigger an SMI/SCI to the CPU.

GPI_RE
Read enable for general-purpose inputs (if configuration register Function 3, offset 40h, bit 6 is cleared). This pin connects to the output enable pin ($\overline{OE}$) of the external FS244 buffers connecting SD15–8 and XD7–0 for GPI15–0.

GPIO4/
EXTSMI4/
$\overline{GPI\_WE}$

Multifunction Pin ***Bidirectional***

GPIO4
General-purpose I/O (if configuration register Function 3, offset 40h, bit 7 is set).

EXTSMI4
An external input signal to trigger an SMI/SCI to the CPU.

GPI_WE
Write enable for general-purpose inputs (if configuration register Function 3, offset 40h, bit 7 is cleared). This pin connects to the latch enable pin ($\overline{OE}$) of the external FS244 buffers connecting SD15–8 and XD7–0 for GPI15–0.

5 Functional Operations

5.1 PCI Bus-Initiated Accesses

The AMD-645 Peripheral Bus Controller is responsible for decoding PCI bus requests from PCI bus masters, initiating the requested actions, and responding in the manner required by the PCI bus protocol.

5.1.1 Overview

The AMD-645 Peripheral Bus Controller responds to PCI bus cycles in one of the two following ways.

Positive Decode

If the PCI address matches an address block defined in the AMD-645 Peripheral Bus Controller as positive ISA decode space, the AMD-645 Peripheral Bus Controller claims the cycle and asserts DEVSEL after the first clock following FRAME first sampled asserted. This same DEVSEL assertion time occurs during all configuration cycles when IDSEL is sampled active.

Subtractive Decode

The AMD-645 Peripheral Bus Controller is assumed to be the only agent responsible for any PCI cycles which are not claimed by other PCI targets. It determines if a PCI cycle is unclaimed by the process of subtractive decoding. If a PCI address does not match any address block defined in the AMD-645 Peripheral Bus Controller, and the DEVSEL input is sampled inactive for three clocks after FRAME is first sampled asserted, the AMD-645 Peripheral Bus Controller responds to the cycle. DEVSEL timing for subtractive decoding is fixed at medium time slot.

The AMD-645 Peripheral Bus Controller also generates an ISA bus cycle for any memory or I/O cycle claimed by the ISA function.

5.1.2 Bus Cycle Decoder

Table 5-1 shows how the AMD-645 Peripheral Bus Controller decodes the PCI command signals when an initiator generates a bus cycle.

Table 5-1. PCI Bus Command Encoding and Types

C/$\overline{BE3}$	C/$\overline{BE2}$	C/$\overline{BE1}$	C/$\overline{BE0}$	Command Type
0	0	0	0	Interrupt Acknowledge
0	0	0	1	Special Cycles
0	0	1	0	I/O Read
0	0	1	1	I/O Write
0	1	0	0	Reserved
0	1	0	1	Reserved
0	1	1	0	Memory Read
0	1	1	1	Memory Write
1	0	0	0	Reserved
1	0	0	1	Reserved
1	0	1	0	Configuration Read
1	0	1	1	Configuration Write
1	1	0	0	Memory Read Multiple
1	1	0	1	Dual Address Line
1	1	1	0	Memory Read Line
1	1	1	1	Memory Write and Invalidate

5.2 PCI Bus Commands

The AMD-645 Peripheral Bus Controller responds to the PCI bus commands according to the descriptions in the following sections.

5.2.1 Interrupt Acknowledge

The AMD-645 Peripheral Bus Controller ignores interrupt acknowledge cycles.

5.2.2 Special Bus Cycles

The AMD-645 Peripheral Bus Controller monitors all special bus cycles.

5.2.3 I/O Read/Write

All I/O accesses not claimed by other PCI targets through the assertion of DEVSEL are passed to the ISA bus controller and executed as standard ISA bus cycles. The AMD-645 Peripheral Bus Controller steers the data between the PCI AD bus and the ISA SD bus or the IDE data bus, as required by each cycle type. If the access is to an on-chip I/O location, then the data is steered between the AD bus, the SD bus, and the selected internal location, as required by the cycle type.

The AMD-645 Peripheral Bus Controller asserts TRDY upon completion of all ISA bus accesses. 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 5-3 on page 5-6.

The I/O-related ISA bus signals are IOR, IOW, and IOCS16. IOR is active during an I/O read cycle, while IOW is active during a write cycle. IOCS16 asserted indicates that a 16-bit slave is responding. A High level on IOCS16 indicates that an 8-bit slave is responding.

The AMD-645 Peripheral Bus Controller decodes the PCI commands and issues a command in the middle of TC or at the beginning of TW1, depending on the setting of bit 7 of the ISA Bus Control register, Function 0, offset 40h (see page 7-17). 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 5-1 illustrates I/O accesses for both read and write, including the insertion of wait states.

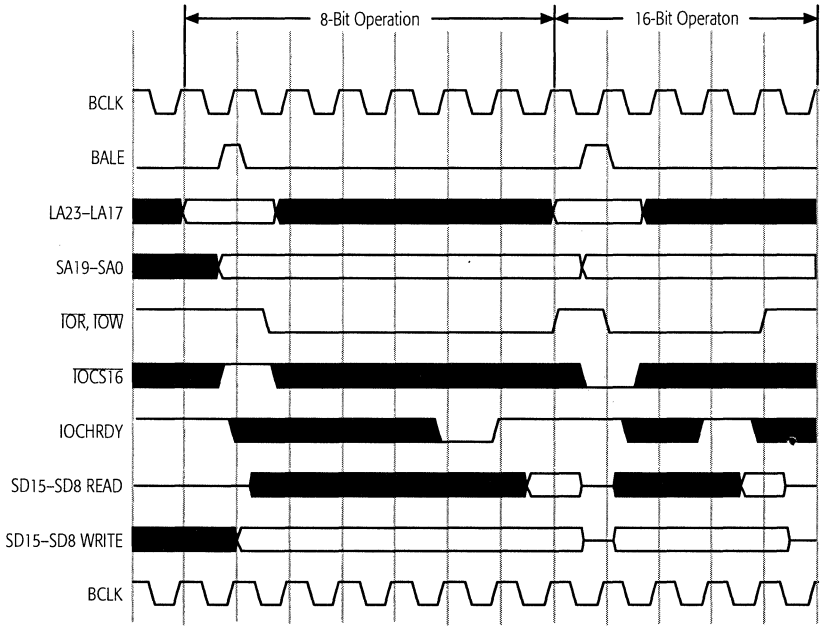


Figure 5-1. I/O Access

For 32-bit or 24-bit accesses to 16-bit ISA bus slaves, or for 32-bit, 24-bit, or 16-bit accesses to 8-bit ISA bus slaves, the AMD-645 Peripheral Bus 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. The conversion of 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-645 Peripheral Bus Controller converts a CPU request for 16-bit data from an 8-bit peripheral into two 8-bit cycles as depicted in Figure 5-2.

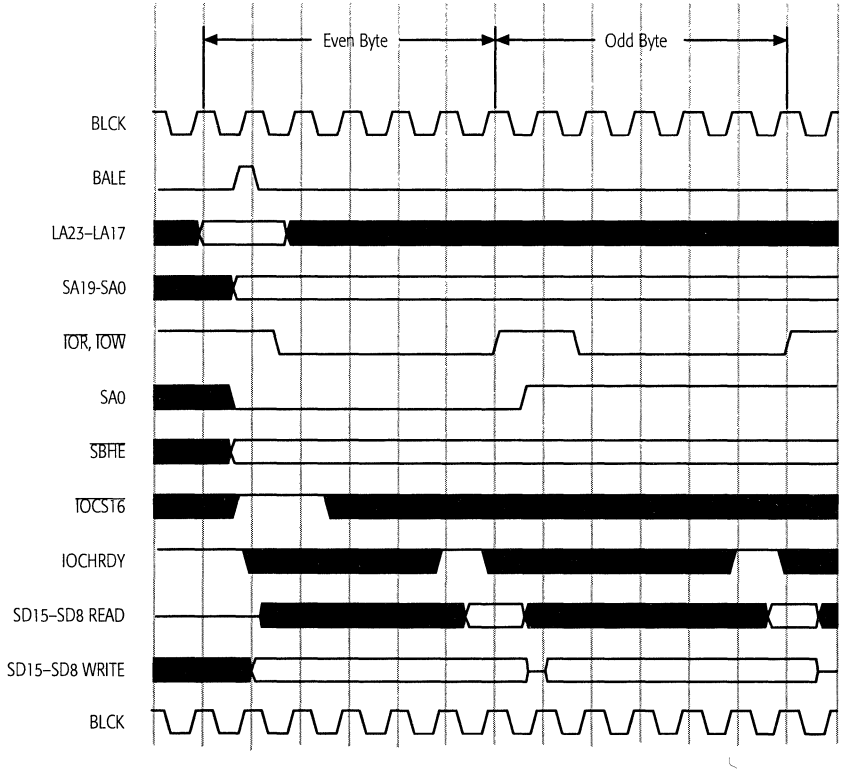


Figure 5-2. I/O Cycle 16-Bit to 8-Bit Conversion

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/Write

The AMD-645 Peripheral Bus Controller directs all memory accesses not claimed by other targets to the ISA bus. The AMD-645 Peripheral Bus Controller steers data between the PCI AD bus and the ISA data bus as required by the requested cycle.

The AMD-645 Peripheral Bus Controller supports bursting (multiple read or write transactions). If FRAME and IRDY are

asserted at the same time, the AMD-645 Peripheral Bus Controller 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 5-3 shows the timing of a non-posted PCI cycle forwarded to the ISA bus.

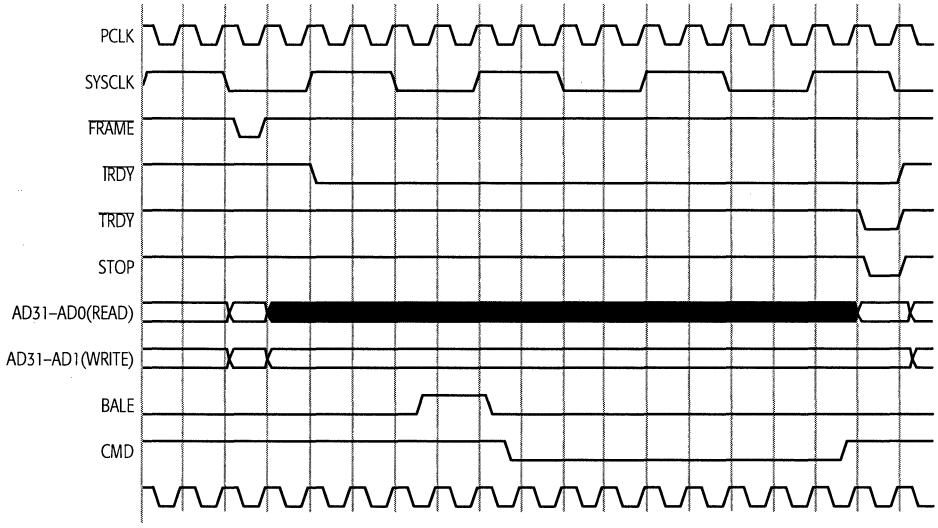


Figure 5-3. Non-Posted PCI-to-ISA Access

If the AMD-645 Peripheral Bus 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 without 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-645 Peripheral Bus Controller disconnects by asserting TRDY and STOP at the end of the current data phase.

Memory write posting in the AMD-645 Peripheral Bus Controller is enabled by setting the Post Memory Write Enable

bit, configuration register Function 0, offset 46h, bit 0 (see page 7-21). When write posting is enabled, TRDY is asserted one clock cycle after both FRAME and IRDY are sampled active. The AMD-645 Peripheral Bus Controller 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 5-4.

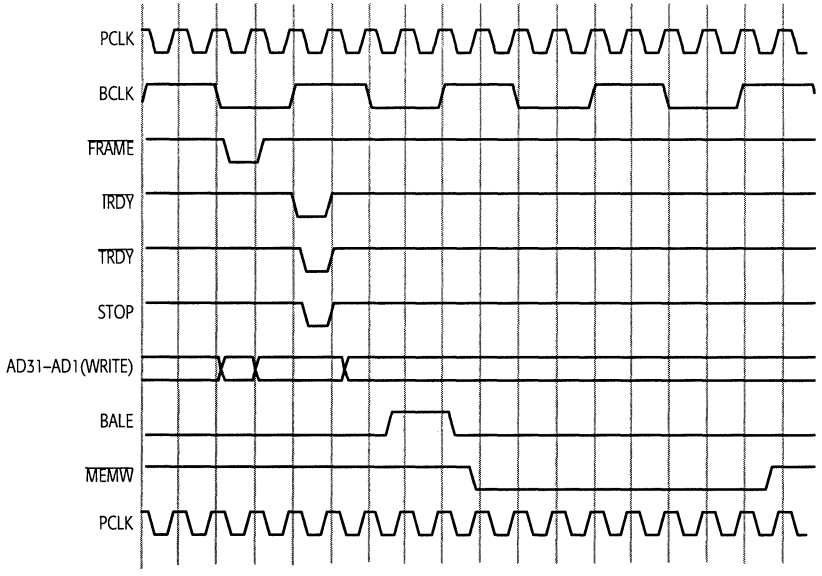


Figure 5-4. Posted PCI-to-Memory Write

The memory-related ISA bus control signals are MEMR, SMEMR, MEMW, SMEMW, and MEMCS16. SMEMR and SMEMW are active 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 5-5.

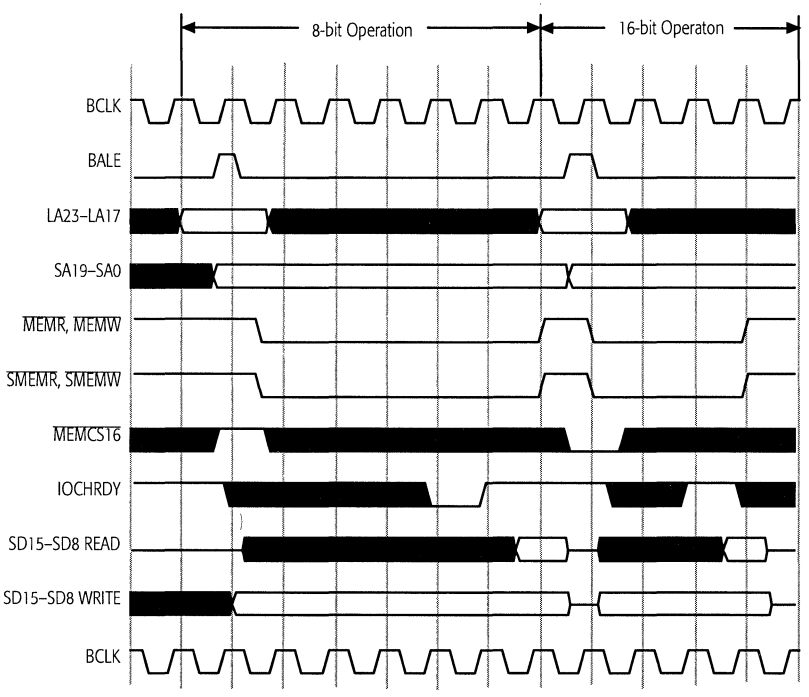


Figure 5-5. ISA Bus Memory Access Cycle

The command signals become active 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 7-17). Under default settings, the command signals are negated at the beginning of TW5 for an 8-bit operation, and at the beginning of TW2 in the case of a 16-bit operation. It may be necessary to delay the rising edge of command signals by one BCLK. This delay can be achieved by setting bit 5 of the ISA Bus Control register. For slow peripherals, wait states may be inserted by pulling IOCHRDY Low by the middle of TW4 for 8-bit cycles and by the beginning of TW2 for 16-bit cycles.

The AMD-645 Peripheral Bus Controller 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 slave results in two 16-bit accesses. The AMD-645 Peripheral Bus Controller also converts requests for non-contiguous bytes by converting the access to the appropriate ISA bus cycles. These conversion cycles are shown in Figures 5-6, 5-7, and 5-8.

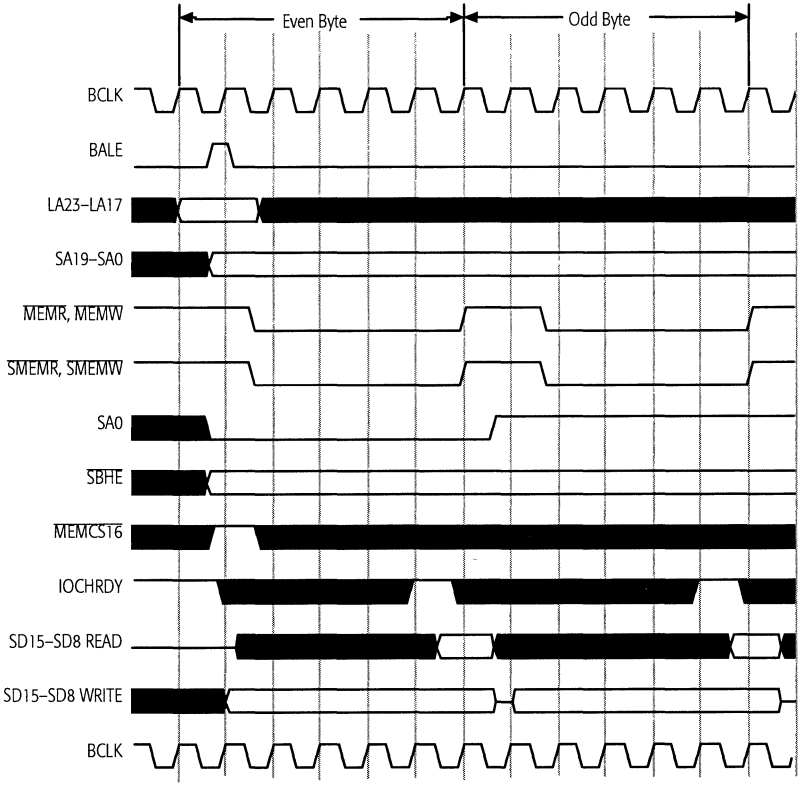


Figure 5-6. ISA Bus Memory Cycle: 16-Bit to 8-Bit Conversion

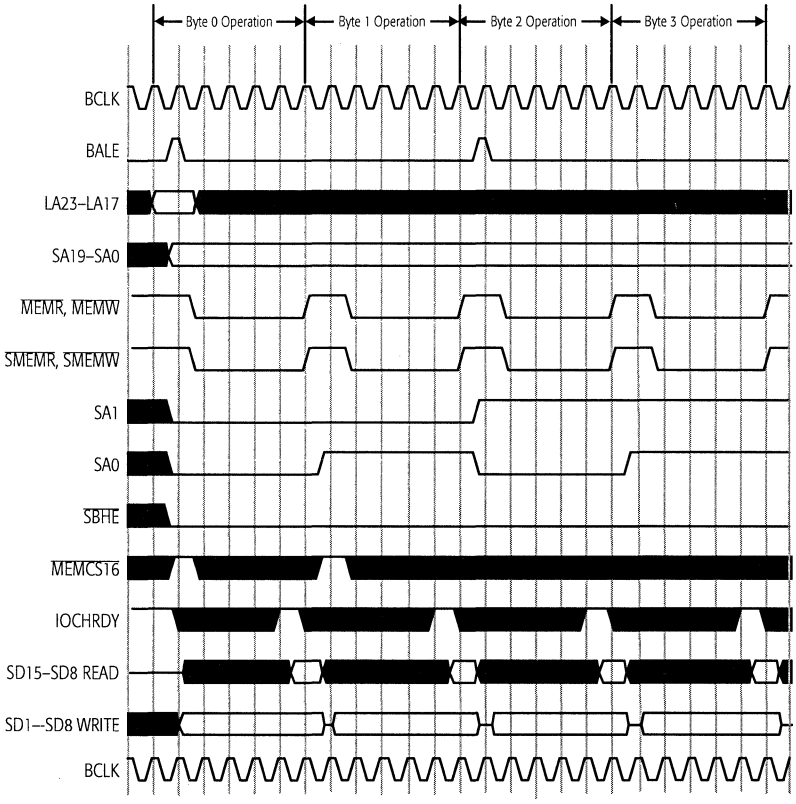


Figure 5-7. Memory Cycle 32-Bit to 8-Bit Conversion

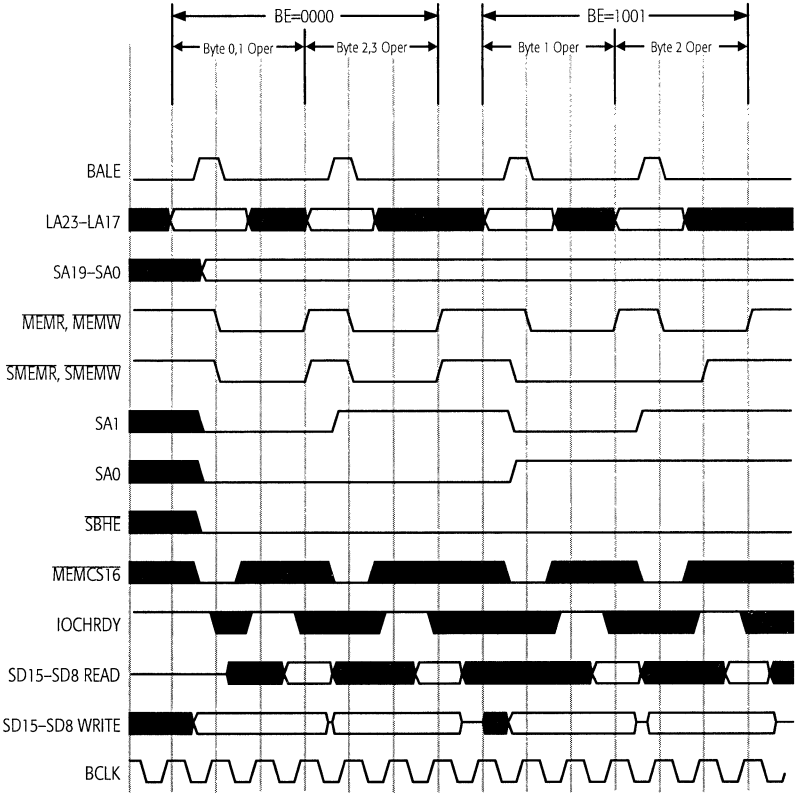


Figure 5-8. Memory Cycle 32-Bit to 16-Bit Conversion

If the memory accessed is ROM, the timing is different for command signals MEMR and SMEMR, which become active 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 7-17). Figure 5-9 shows a ROM access. Figure 5-10 shows requests for 32 bits of data from 8-bit ROMs.

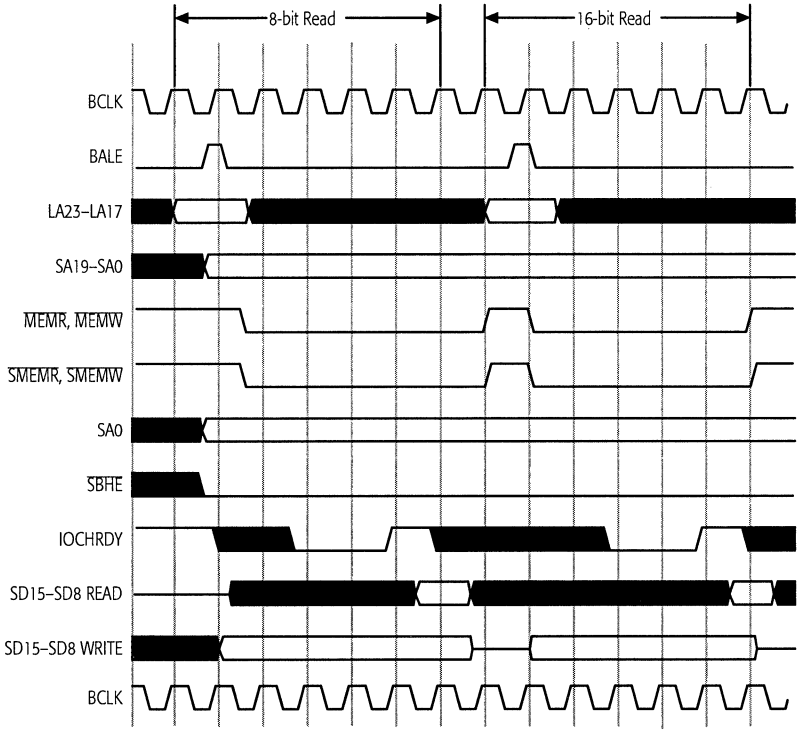


Figure 5-9. ROM Access

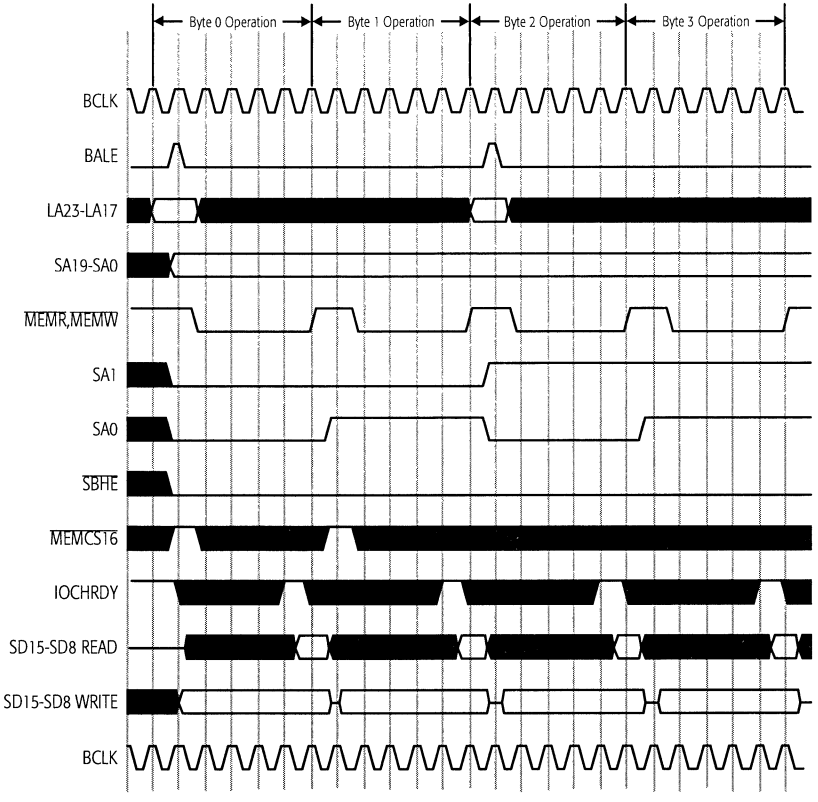


Figure 5-10. ROM Cycle 32-Bit to 8-Bit Conversion

SA1, SBHE, and SA0 are a direct decode of the C/BE3–C/BE0 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 5-2.

Table 5-2. 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/Write

As a target, the AMD-645 Peripheral Bus Controller responds to both read and write configuration cycles. Access to the configuration address space requires device selection decoding to be done externally via the IDSEL pin, 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-645 Peripheral Bus Controller IDSEL is system-specific, but the recommended connection is to AD18.

If the AMD-645 Peripheral Bus Controller 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-645 Peripheral Bus Controller drives the requested configuration register data onto AD31–AD0, 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-645 Peripheral Bus Controller 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 Figures 5-11 and 5-12.

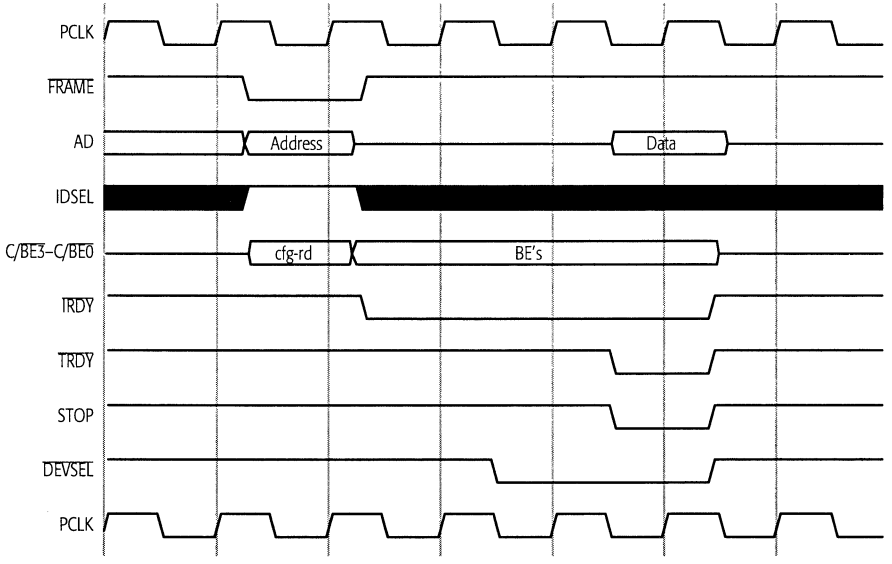


Figure 5-11. Configuration Read Cycle

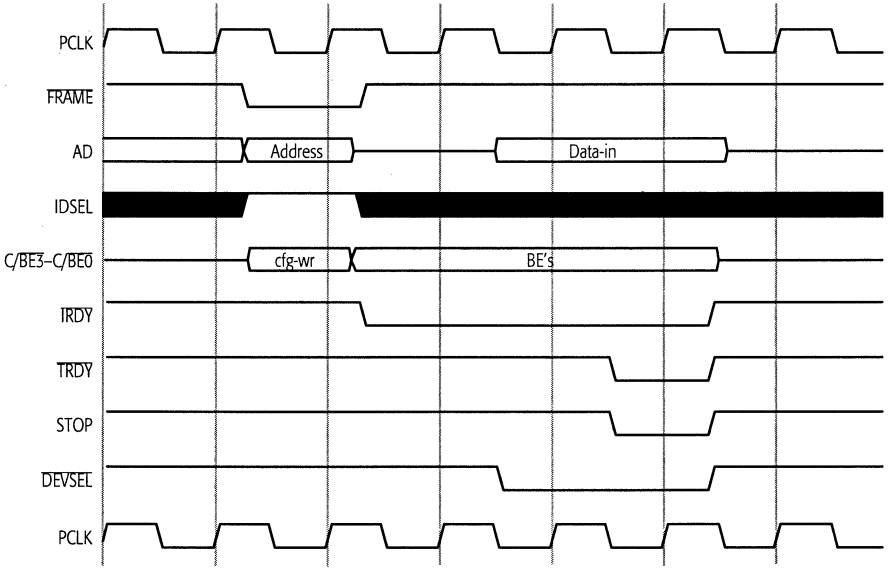


Figure 5-12. Configuration Write Cycle

5.2.6 Memory Read Multiple

The memory read multiple command is treated the same as a memory read command by the AMD-645 Peripheral Bus Controller.

5.2.7 Dual Address Line

The AMD-645 Peripheral Bus Controller supports 32-bit addressing only, so dual address line commands are ignored. There is no response.

5.2.8 Memory Read Line

The AMD-645 Peripheral Bus Controller treats the memory read line command just as it does the memory read command.

5.2.9 Memory Write Invalidate

The AMD-645 Peripheral Bus Controller 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-645 Peripheral Bus 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-645 Peripheral Bus Controller address block is forwarded to the ISA bus. The timing for subtractive decoding is shown in Figure 5-13.

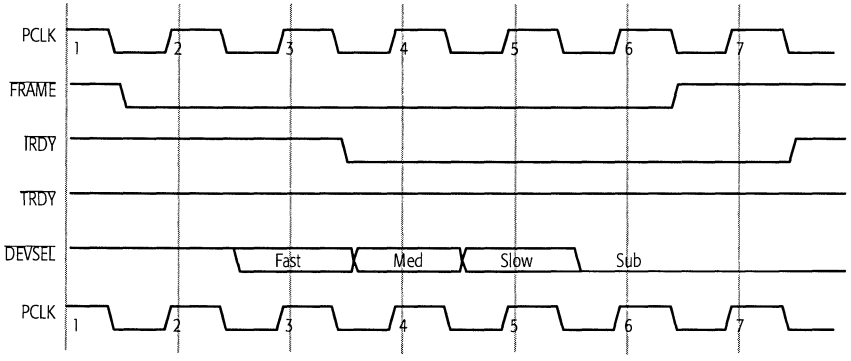


Figure 5-13. Subtractive Decode Timing

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 7-17). 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-645 Peripheral Bus 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 $\overline{\text{IRDY}}$ within eight clocks during all data phases. Ideally, $\overline{\text{IRDY}}$ 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 at a data width of either 8 bits 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 DRQ7–DRQ5 or DRQ3–DRQ0 signals. When the AMD-645 Peripheral Bus Controller 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-645 Peripheral Bus 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-645 Peripheral Bus Controller asserts the corresponding $\overline{\text{DACK}}$ line to indicate a DMA acknowledge. Prefetch data is transferred in response to subsequent DMA requests without further PCI bus accesses.

When the AMD-645 Peripheral Bus Controller detects a memory write request, it asserts the corresponding $\overline{\text{DACK}}$ 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-645 Peripheral Bus Controller 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 go High after the DMA is acknowledged and any pending ISA bus cycle has completed. The DMA address is placed on LA23–LA20 and SA19–SA0. Two DMACLK cycles later, either $\overline{\text{MEMR}}$ and $\overline{\text{IOW}}$ or $\overline{\text{MEMW}}$ and $\overline{\text{IOR}}$ are asserted, depending on the direction of the transfer. If the ISA Command Delay bit of the ISA Bus Control register is set, $\overline{\text{MEMR}}$ is asserted one DMACLK cycle earlier. The command remains active 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 5-14 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.

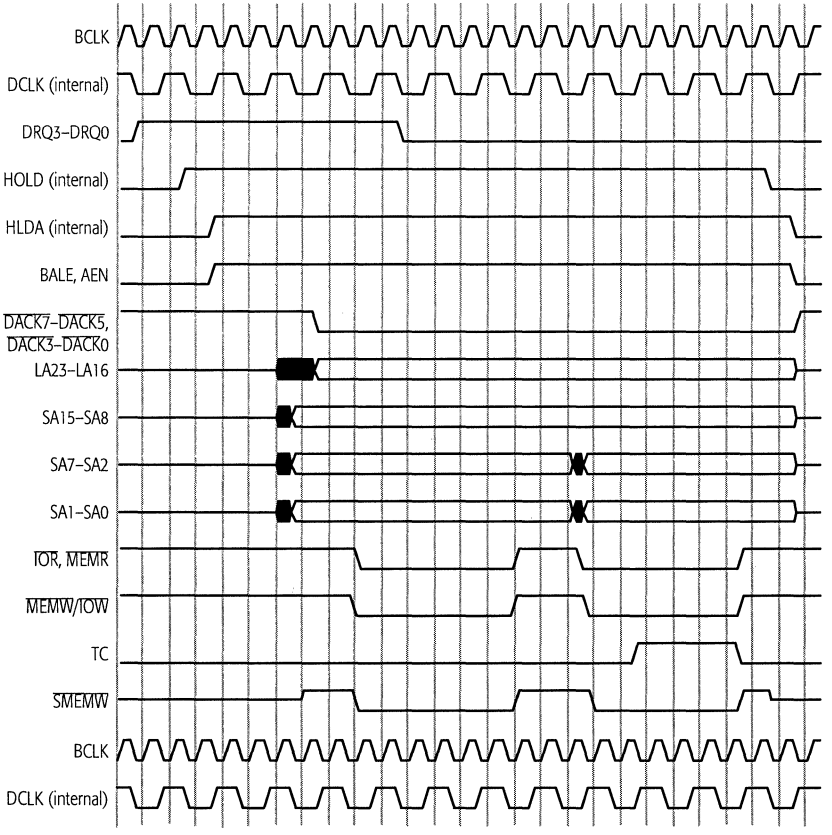


Figure 5-14. DMA Transfer Cycle

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 5-15, using a DMA channel which has been placed in the cascade mode. The AMD-645 Peripheral Bus 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.

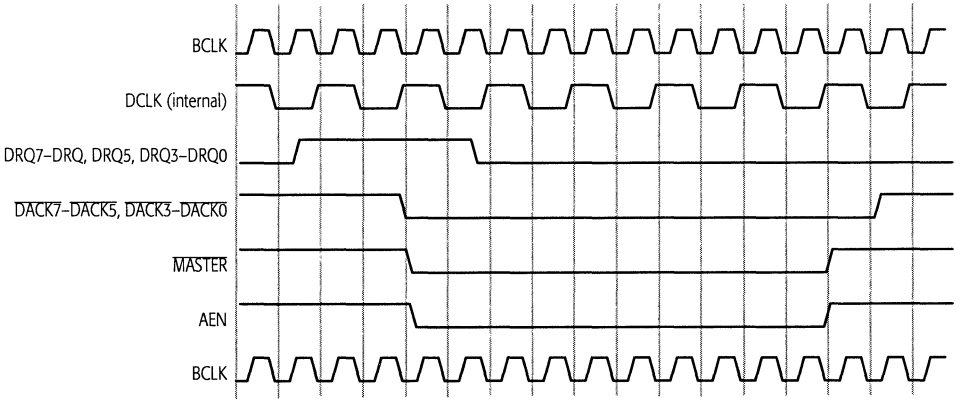


Figure 5-15. ISA Bus Master Arbitration Timing

When the AMD-645 Peripheral Bus Controller detects MEMR or MEMW active, 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 active. Figure 5-16 shows an ISA bus master memory read, and Figure 5-17 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-645 Peripheral Bus Controller 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 Figures 7-16 and 7-17, with the single substitution of IOR and IOW for MEMR and MEMW.

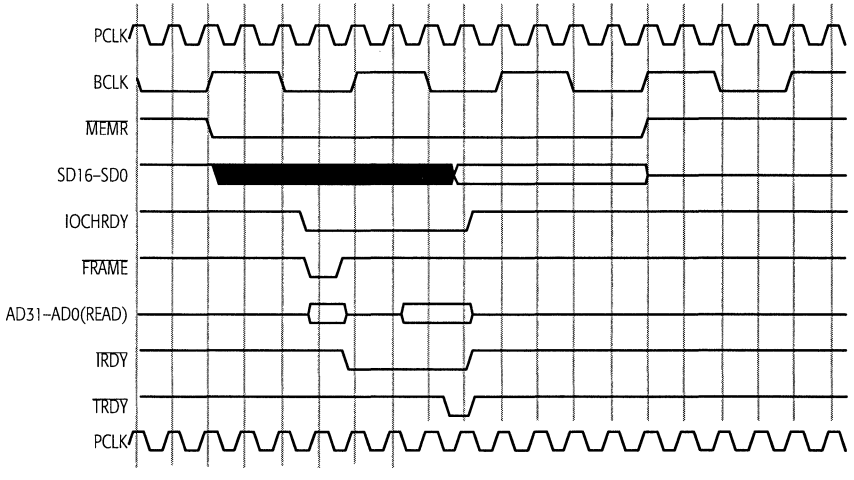


Figure 5-16. ISA Bus Master-to-PCI Memory (Memory Read)

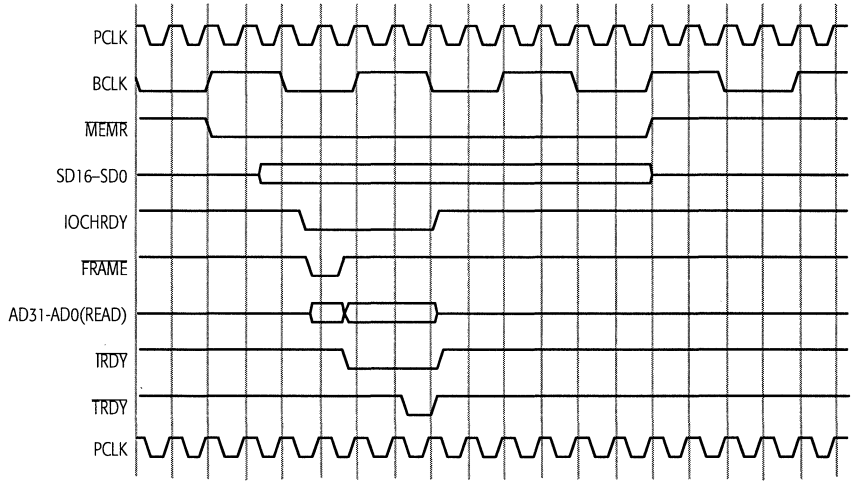


Figure 5-17. ISA Bus Master-to-PCI Memory (Memory Write)

5.5 PCI Bus Arbitration

The signals $\overline{\text{PREQ}}$ and $\overline{\text{PGNT}}$ are used to control requesting and granting of the PCI bus between the AMD-645 Peripheral Bus Controller ISA bridge and the AMD-640 System Controller. The AMD-640 System Controller write buffer control is also implemented in the bi-directional protocol to ensure data coherency during DMA and master mode operations. The AMD-645 Peripheral Bus Controller write poster is also disabled whenever the AMD-640 System Controller write buffers are disabled.

5.6 I/O and Memory Mapping

The AMD-645 Peripheral Bus Controller decodes PCI bus addresses to determine the destination of a PCI memory or I/O request. The AMD-645 Peripheral Bus Controller address decoder distinguishes five general regions for memory or I/O accesses. 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 five general regions are described in the following paragraphs.

IDE Bus I/O Location

The AMD-645 Peripheral Bus 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-645 Peripheral Bus Controller 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)

An ISA bus I/O access cycle is generated via subtractive decoding and is responded to by the AMD-645 Peripheral Bus Controller when it recognizes an on-chip address during the ISA bus cycle.

ISA Bus I/O Location (Off-Chip)

A standard ISA bus I/O access cycle is generated via subtractive decoding when no other PCI slave responds to a PCI I/O cycle. Data is passed between the PCI data bus (AD31–AD0) and the ISA data bus (SD15–SD0). ROMCS/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-645 Peripheral Bus Controller detects a memory access in the ISA slot bus address range. Data is passed between the PCI data bus (AD31–AD0) and the ISA data bus (SD15–SD0). The AMD-645 Peripheral Bus Controller 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, ROMCS/KBCS is asserted.

5.6.1 I/O Mapping

I/O addresses that are 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 5-3, while the distributed DMA I/O locations are at a programmable base address.

Table 5-3. 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 $\overline{\text{IOW}}$ strobe. All I/O read cycles drive data from the SD bus onto the AD bus and generate an $\overline{\text{IOR}}$ strobe. The AMD-645 Peripheral Bus 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 5-4 shows the various memory regions and the destinations (PCI, ROM, or ISA) supported by the AMD-645 Peripheral Bus Controller.

Table 5-4. Memory Address Mapping

Range	Address	Destination	Comments
0 to 786Kbytes	00000h–BFFFFh	PCI bus space ISA bus space	Selected by active DEVSEL (By subtractive decode)
786Kbytes to 960Kbytes	C0000h–EFFFFh	PCI bus space ISA bus space ISA ROM space	Selected by active DEVSEL (By subtractive decode) or selected by ROM decode control
960Kbytes to 1 Mbyte	F0000h–FFFFFh	ISA bus space ISA ROM space	(By subtractive decode)
1 Mbyte to 15.875 Mbytes	100000h–FDFFFFh	PCI bus space ISA bus space	Selected by active DEVSEL (By subtractive decode)
15.875 Mbytes to 16 Mbytes	FE0000h–FFFFFFh	PCI bus space ISA bus space	Selected by active DEVSEL (By subtractive decode)
16 Mbytes t0 128 Mbytes	1000000h–7FFFFFFh	PCI bus space Aliased ISA bus space	Selected by active DEVSEL (By subtractive decode)
128 Mbytes to (4Gbytes - 512Kbyte)	80000000h–FFF7FFFFh	PCI bus space Aliased ISA bus space	Selected by active DEVSEL By subtractive decode only
(4Gbytes - 512Kbytes) to 4G	FFF80000h–FFFFFFFFh	ISA ROM space	(By subtractive decode) or 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 active within the fast, medium, or slow sample periods, the AMD-645 Peripheral Bus Controller is deselected and a PCI target device completes the cycle.
- If the DEVSEL input is not sampled active within the fast, medium, or slow sample periods, the AMD-645 Peripheral Bus Controller 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-645 Peripheral Bus Controller initiates a PCI cycle. IF DEVSEL is not asserted within the fast, medium, or slow sample periods, the AMD-645 Peripheral Bus Controller executes a subtractive decode which directs the access to the ISA bus, and IOCHRDY is re-asserted to allow the ISA cycle to complete.

Because the AMD-645 Peripheral Bus Controller subtractive decode method can be used to alias ISA memory space into the upper CPU/PCI address regions, ISA memory that is normally 'hidden' behind DRAM in the lower 16 Mbytes can be accessed via the aliased space. However, DMA and master mode cycles to the ISA memory areas in such configurations are not permitted, because conflicts with PCI bus slaves may arise.

ISA Memory

All memory accesses below 16 Mbytes not accepted by PCI bus devices through the assertion of DEVSEL are directed to the ISA bus. The AMD-645 Peripheral Bus 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 16 Mbytes alias to the 24-bit ISA bus addresses. PCI accesses to these regions should be performed only if no DMA or master mode cycles ever access the referenced locations, because a slot bus memory device may occupy the same aliased address an PCI bus memory and bus contention would occur.

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 of the bits in ROM decode control enable different address ranges to be included in the ROMCS decode. All PCI accesses in the highest 512 Kbytes of each 16 Mbyte memory space (XXF80000h to XXFFFFFFh) are always system ROM accesses. System ROM accesses are a subset of ISA bus accesses. Standard ISA bus accesses are generated on system ROM accesses, with the following differences:

- ROMCS is always asserted on system ROM accesses. XDIR is set to reflect the cycle type, read or write.
- Additional ISA bus wait states can be programmed for system ROM accesses via the ROM Wait States bit of the ISA Bus Control register.

The AMD-645 Peripheral Bus Controller provides the 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 5-5. Setting the indicated bit enables the address range shown to be included in the ROMCS decode.

Table 5-5. 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 that reside on the X-bus. Bit 0 of the ISA Bus Control register (Function 0 offset 40h) is provided to enable write cycle generation.

5.7 Clock Generation

The clocks described in the following paragraphs are used or generated by the AMD-645 Peripheral Bus Controller.

PCLK This input signal is the PCI clock used to synchronize the interface to all PCI bus devices.

OSC This input signal is a 14.318-MHz clock common to the ISA bus signal OSC. It is used by the internal RTC.

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 (see page 7-18). 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 ISA Clock Control register.
2. Program bits 2–0, the ISA Bus Clock Select bits of this register, writing the value selected from Table 5-6.
3. Set bit 3 of ISA Clock Control register.

Table 5-6. 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

5.8 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 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.

5.8.1 DMA Controllers

The AMD-645 Peripheral Bus 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.

5.8.2 DMA Controller Registers

The 8237 megacells can be programmed anytime $\overline{\text{PGNT}}$ is inactive, i.e., when DMA controllers are not in operation. Table 5-7 lists the I/O addresses of all slave and master DMA

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

Slave & Master DMA Controllers Ports C0h–DFh

The slave and master DMA controller ports are listed in Table 5-7.

Table 5-7. Ports 00h–0Fh Master DMA Controller

Slave 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.

When writing to a channel's 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 will toggle back to zero.

The 8237 DMA controller megacells allow the user to program the active level of the DREQ and $\overline{\text{DACK}}$ signals to be Low or High. Because the two megacells are cascaded together internally on the chip, DREQ should always be programmed active High and $\overline{\text{DACK}}$ 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, should be loaded with some valid value, even if the channels are not used.

5.8.3 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 AD16–AD8 pins.

5.8.4 Page Registers

The AMD-645 Peripheral Bus 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 5-8 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 5-8. 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 AD23–AD16 bus lines. In normal operation, zeroes are driven onto PCI address bits AD31–AD24 during DMA cycles, making the AMD-645 Peripheral Bus Controller backward-compatible with the PC/AT standard.

5.8.5 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 AD31–AD16 (8-bit channels) and AD31–AD17 (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 during state S1. The internal middle address bit latches latch this value in. The middle address bit latches are output on PCI address bits AD15–AD8 for 8-bit channels and AD16–AD9 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 AD7–AD0 for 8-bit channels and AD8–AD1 for 16-bit channels.

$\overline{\text{SBHE}}$ 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 5-9 shows the mapping from the DMA subsystem signals to slot bus signals. Table 5-10 shows the mapping of the AMD-645 Peripheral Bus Controller DMA subsystem signals to PCI address bus signals.

Table 5-9. 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]

Table 5-9. 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
	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]
		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]
		$\overline{A[0]}$	$\overline{SBHE}$	-------
		VSS	-------	$\overline{SBHE}$

Table 5-10. 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]

Table 5-10. 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
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]
	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]
		$\overline{A[0]}$	-------	BE[3], BE[2]
		A[1] + A[0]	$\overline{BE[0]}$	-------
		A[1] + $\overline{A[0]}$	$\overline{BE[1]}$	-------
		A[1] + A[0]	$\overline{BE[2]}$	-------
		$\overline{A[1]}$ + $\overline{A[0]}$	$\overline{BE[3]}$	-------

5.8.6 Type F DMA

Type F DMA is supported on all channels. The channels may be individually enabled to provide Type F DMA timing, using the Type F DMA control register (Function 0, offset 45h) as shown in Table 5-11. Therefore, configuration software needs to detect Type F-capable devices and configure their channels only once after reset.

Table 5-11. 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 $\overline{\text{DACK}}$ 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.

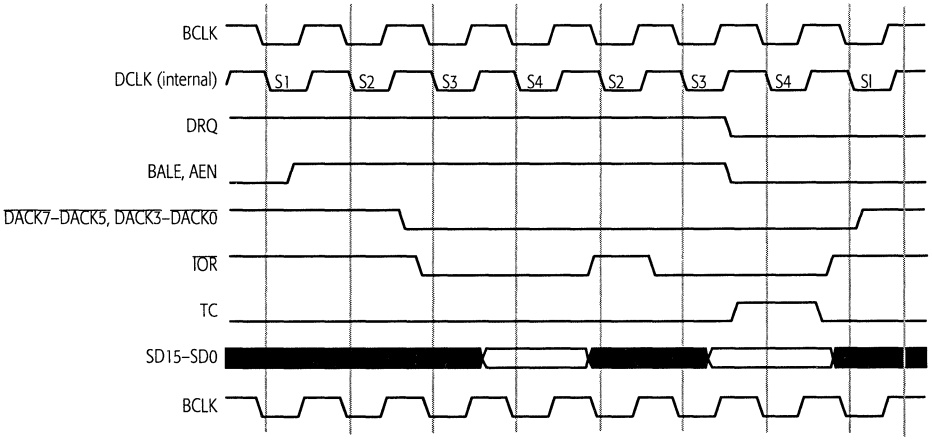


Figure 5-18. Type F DMA Timing

5.8.7 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 $\overline{\text{DACK}}$ is driven High.

5.8.8 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-645 Peripheral Bus 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 further. To add extra wait states, an external device should 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 will be inserted before the DMA controller transitions to state S4.

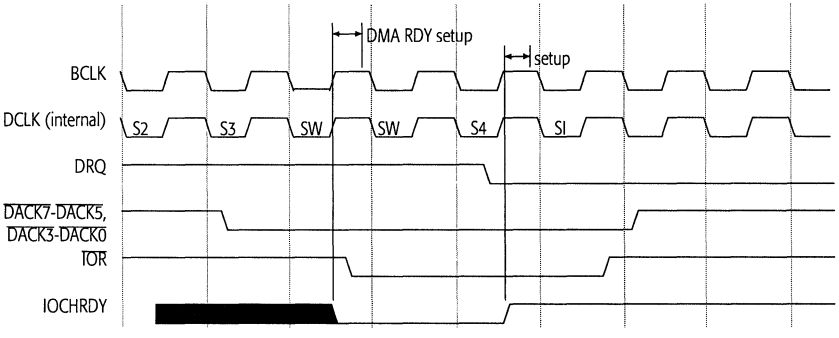


Figure 5-19. DMA Ready Timing

5.8.9 External Cascading

An external DMA controller or bus master can be attached to an AT-compatible design through the AMD-645 Peripheral Bus Controller 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 should then be 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-645 Peripheral Bus Controller will 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 that channel's $\overline{\text{DACK}}$ line goes active, the external device can then pull the $\overline{\text{MASTER}}$ signal Low. As in the DMA controller cascading, the AMD-645 Peripheral Bus Controller does not drive the address, data, and command signals while the cascaded channel's $\overline{\text{DACK}}$ signal is active.

5.8.10 PCI Bus Request Arbiter

The PCI bus request arbiter is used to select between the three possible sources for a PCI bus request to the system controller. A PCI bus request can be generated under the following circumstances:

- The DMA read buffer is empty during DMA memory reads
- The most significant byte (MSB) of the DMA write buffer is full during DMA memory writes
- A DMA acknowledge for a cascaded channel is generated
- When the IDE dual channel controller issues a master bus request

The arbiter has the following four inputs:

- The ISA bus, due to DMA buffer requests or master mode acknowledge
- The primary IDE channel

- The secondary IDE channel
- The IDE master block

At the end of a PCI bus request from any of the sources, the arbiter checks to see if any of the other sources is still requesting the PCI bus. If so, the arbiter sends an acknowledge signal to that source and leaves the $\overline{\text{PREQ}}$ line active. This continues as long as one of the sources is requesting the PCI bus. Only if no source is generating a PCI bus request will the arbiter negate the PCI bus request signal and return control to the system controller. The three IDE bus request signals will go inactive for at least two PCI clocks at the end of each burst transfer.

5.8.11 DMA Read and Write Buffers

Write merging occurs when a sequence of individual DMA memory writes of bytes or words is merged into a single doubleword. DMA memory writes within a singular DMA acknowledge are merged in order to enhance PCI bus performance. Thus DMA writes are merged when the active channel is in demand or block transfer mode but are not merged when the active channel is in the single transfer mode.

Read prefetching occurs when a DMA memory read causes all of the PCI byte enables to be asserted for the request; that is, the entire doubleword for the requested address is fetched. DMA reads are always prefetched to enhance PCI bus performance.

5.8.12 PCI Target Retries

When the ISA interface is busy due to the ISA bus being owned by the DMA controller or an ISA bus master, PCI target requests to the ISA bus or DMA controller are retried.

5.9 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.

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.

5.10 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-645 Peripheral Bus Controller supports Ultra DMA transfer mode 0, 1 and 2. Table 5-12 lists the Ultra DMA interface signals that appear on the IDE drive cable interface.

Table 5-12. Ultra DMA Interface Signals

Signal	Source	Signal	Source
RESET	Host	CSEL	Host
DD15–DD0	Bidirectional	$\overline{\text{DMACK}}$	Host
DMARQ	Device	INTRQ	Device
$\overline{\text{DIOR}}$/HDMARDY/ HSTROBE	Host	DA2–DA0	Host
$\overline{\text{DIOW}}$/STOP	Host	$\overline{\text{PDIAG}}$	Device
IORDY/$\overline{\text{DDMARDY}}$/ DSTROBE	Device	$\overline{\text{CS0}}$, $\overline{\text{CS1}}$	Host
CSEL	Host	$\overline{\text{DASP}}$	Device

HDMARDY is a flow control signal for Ultra DMA input data bursts. It is asserted by the host when it is ready to receive DMA data. The host negates HDMARDY to pause an Ultra DMA data in transfer.

HSTROBE is the strobe signal from the host for an Ultra DMA output data transfer. Both edges of HSTROBE latch data from DD15–DD0 into the device. The host may stop toggling HSTROBE to pause an Ultra DMA output data transfer.

STOP can be asserted by the host during or after data transfer in an Ultra DMA mode to signal the termination of the burst.

DDMARDY is a flow control signal for output data bursts. It is asserted by the device when it is ready to receive DMA data. The device negates DDMARDY to pause an Ultra DMA output data transfer.

DSTROBE is the strobe signal from the device for an Ultra DMA input data transfer. Both edges of DSTROBE latch data from DD15–DD0 into the host. The device may stop toggling DSTROBE to pause an Ultra DMA data in transfer.

The Ultra DMA protocol has three timing modes—mode 0, mode 1, and mode 2. Only one Ultra DMA mode is active at any time. The IDENTIFY DEVICE data specifies the highest timing mode of which a device is capable. Devices reporting support for Ultra DMA transfer mode 2 must also support mode 0 and mode 1. The control signal STROBE that latches data from DD15–DD0 is generated by the same agent, either host or device, which drives the data onto the bus. Several signal lines assume new functions when the Ultra DMA protocol is active. These signal lines revert to the definitions used for multiword DMA transfers upon the termination of the Ultra DMA transfer. All control signals are unidirectional.

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.

5.10.1 Ultra DMA Read Burst Command

Initiating a Read Burst

Figure 5-20 shows the timing for an Ultra DMA read burst. The device asserts DDRQ to initiate a burst. The host asserts DDACK when it is ready to begin the requested burst. 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. The device 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.

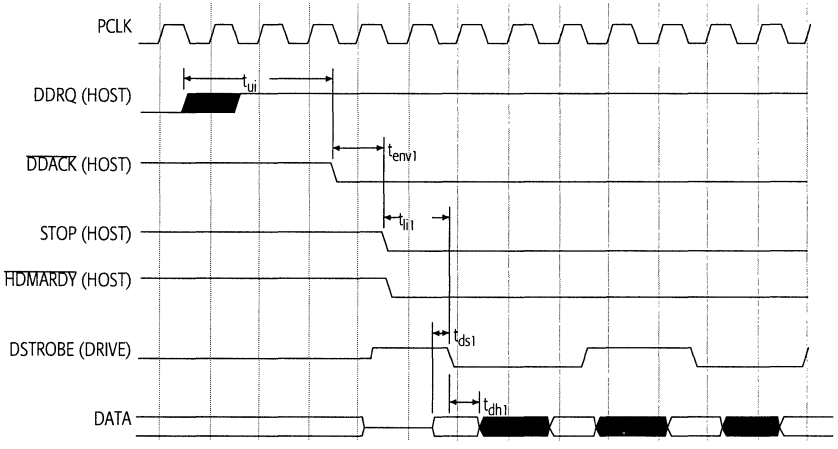


Figure 5-20. Ultra DMA-33 IDE Read Burst

Pausing a Read Burst

Either the device or the host can pause a burst transfer, as shown in Figure 5-21. 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.

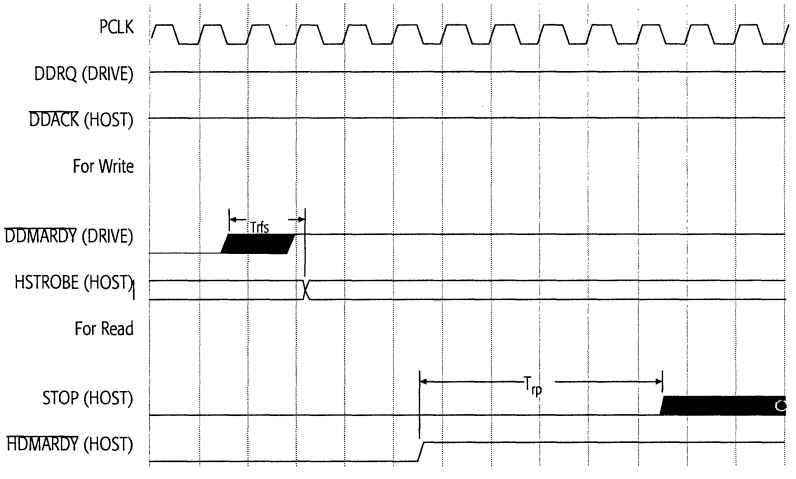


Figure 5-21. Pausing a DMA Burst

Terminating a Read Burst.

Either the device or the host can terminate a burst. The device initiates termination of a read burst by halting DSTROBE toggling and negating DMARQ. The host responds by asserting STOP and negating HDMARDY. 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. Figure 5-22 shows the timing for read burst termination initiated by a device.

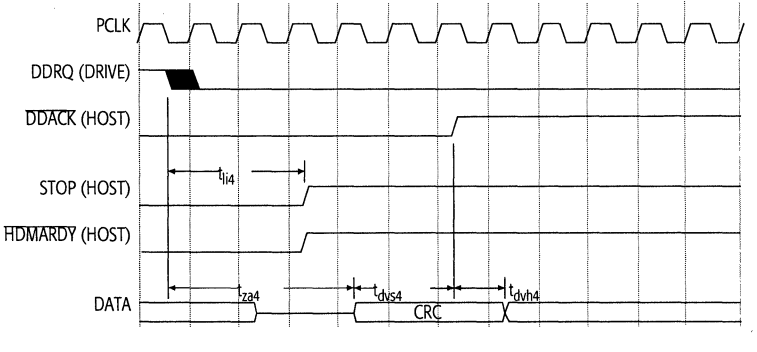


Figure 5-22. Drive Terminating a DMA Read Burst

The host initiates a read burst termination by negating HDMARDY and asserting STOP, as shown in Figure 5-23. The device negates DDRQ. The host then places the result of its CRC (Cyclic Redundancy Check) on DATA and negates DDACK. The CRC is latched in the device at the negating edge of DDACK.

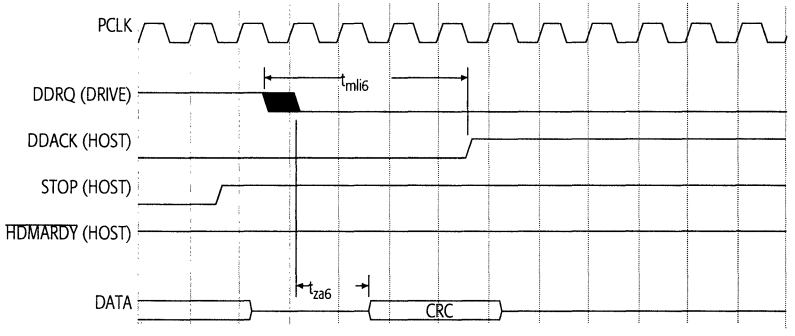


Figure 5-23. Host Terminating DMA Burst During Read Command

5.10.2 Ultra DMA Write Burst Command

Initiating a Write Burst

Figure 5-24 shows the timing for an Ultra DMA write burst. The device asserts DDRQ to initiate a write burst. The host asserts DDACK when it is ready to begin the requested burst. 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. Data is transferred at both edges of HSTROBE until data transfer is complete or the burst is paused.

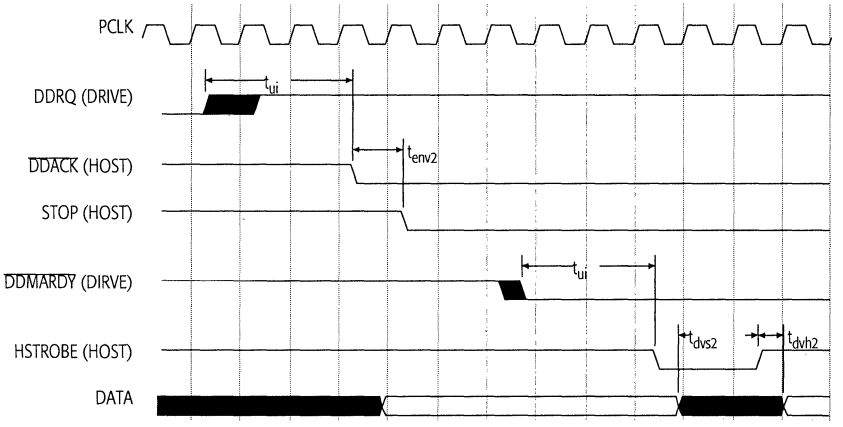


Figure 5-24. Ultra DMA-33 IDE Write Burst

Pausing a Write Burst

Either the device or the host can pause a DMA write burst transfer, as shown in Figure 5-21 on page 5-44. The device pauses a write burst by negating DDMARDY and resumes the burst by reasserting DDMARDY. The host pauses a write burst by halting HSTROBE toggling and resumes the burst by toggling HSTROBE again.

Terminating a Write Burst

Either the device or the host can terminate a write burst. The device initiates burst termination by negating DDMARDY. The host shall halts HSTROBE toggling. The device negates DDRQ, and the host responds by asserting STOP. The host asserts HSTROBE (if it is negated), places the result of its CRC on DATA, and negates DDACK. The CRC is latched in the device at the negating edge of DDACK. Figure 5-25 shows a the timing for a drive terminating a write burst.

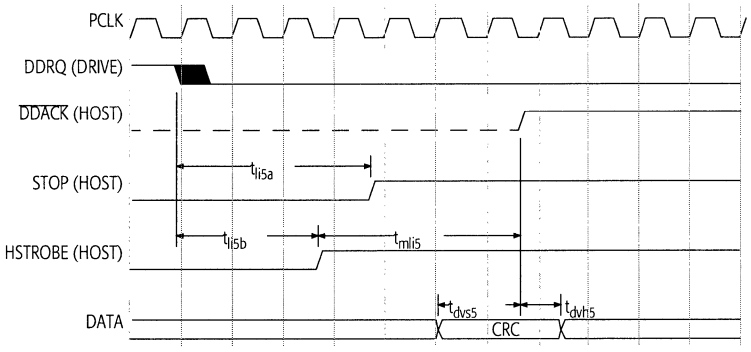


Figure 5-25. Drive Terminating DMA Burst During Write Command

The host initiates burst termination by halting HSTROBE toggling and asserting STOP, as shown in Figure 5-26. The device responds by negating DDRQ and DDMARDY. The host asserts HSTROBE (if it is negated), places the result of its CRC (Cyclic Redundancy Check) on DATA, and negates DDACK. The CRC is latched in the device at the negating edge of DDACK.

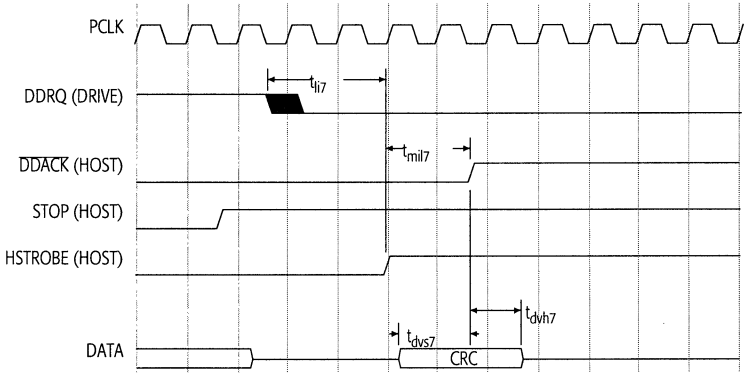


Figure 5-26. Host Terminating DMA Burst During Write Command

5.10.3 Slave DMA Channel

Each slave DMA channel has a block of sixteen 8-bit registers which are defined in Table 5-13. This block is locatable anywhere in the Legacy 64K I/O space by programming the Slave DMA Configuration Register. All slave DMA channels must have an identical programming model. The master DMA is programmed with the base address of each slave DMA by having a matching base address register for each channel.

Table 5-13. Programming Model for Single Slave DMA Channel

Slave Address	Read/ Write	Register Name	Byte DMA Address	Word DMA Address	POR Value
b + 0h	W	Base Address 0–7	CH0 = 0000h CH1 = 0002h Ch2 = 0004h CH3 = 0006h	CH4 = 00C0h CH5 = 00C4h CH6 = 00C8h CH7 = 00CCh	XXh
b + 0h	R	Current Address 0–7	CH0 = 0000h CH1 = 0002h Ch2 = 0004h CH3 = 0006h	CH4 = 00C0h CH5 = 00C4h CH6 = 00C8h CH7 = 00CCh	XXh
b + 1h	W	Base Address 8–15	CH0 = 0000h CH1 = 0002h Ch2 = 0004h CH3 = 0006h	CH4 = 00C0h CH5 = 00C4h CH6 = 00C8h CH7 = 00CCh	XXh
b + 1h	R	Current Address 8–15	CH0 = 0000h CH1 = 0002h Ch2 = 0004h CH3 = 0006h	CH4 = 00C0h CH5 = 00C4h CH6 = 00C8h CH7 = 00CCh	XXh
b + 2h	W	Base Address 16–23	Ch0 = 0087h CH1 = 0083h Ch2 = 0081h CH3 = 0082h	CH4 = N/A CH5 = 008Bh CH6 = 0089h Ch7 = 008Ah	XX
b + 2h	R	Current Address 16–23	Ch0 = 0087h CH1 = 0083h Ch2 = 0081h CH3 = 0082h	CH4 = N/A CH5 = 008Bh CH6 = 0089h 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 Ch1 = 0003h Ch2 = 0005h Ch3 = 0007h	CH4 = 00C2h CH5 = 00C6h Ch6 = 00CAh Ch7 = 00CEh	XXh
b +4h	R	Current Word Count 0–7	Ch0 = 0001h Ch1 = 0003h Ch2 = 0005h Ch3 = 0007h	CH4 = 00C2h CH5 = 00C6h Ch6 = 00CAh Ch7 = 00CEh	XXh

Table 5-13. Programming Model for Single Slave DMA Channel (continued)

Slave Address	Read/ Write	Register Name	Byte DMA Address	Word DMA Address	POR Value
b + 5h	W	Base Word Count 8–15	Ch0 = 0001h Ch1 = 0003h Ch2 = 0005h Ch3 = 0007h	CH4 = 00C2h CH5 = 00C6h Ch6 = 00CAh Ch7 = 00CEh	XXh
b + 5h	R	Current Word Count 8–15	Ch0 = 0001h Ch1 = 0003h Ch2 = 0005h Ch3 = 0007h	CH4 = 00C2h CH5 = 00C6h Ch6 = 00CAh 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)			
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.

5.10.4 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 slave 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 slave 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 will be written 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 slave 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 slave 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 slave channel number is copied to bits 0–3, and the DRQ bit corresponding to the slave channel number is copied to bits 4–7.

5.10.5 DMA Software Commands

Master Clear

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

5.10.6 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 slave DMA does not suffer this problem because it has fully decoded these registers. Table 5-14 shows the relationship between legacy DMA addressing for Base, Count, and Memory Page registers. It also shows where this information is programmed into the slave 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 slave DMA. A slave DMA can be programmed to be in 8-bit/16-bit transfer mode from its PCI configuration space. This mode information defines how the slave DMA treats the data in the registers. Table 5-14 also defines optional non-legacy addressing extensions for the slave.

Table 5-14. 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 Slave 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 Slave 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 slave 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-645 Peripheral Bus Controller implements subtractive decoding for these registers, master DMA blocks that implement them will behave as expected by the distributed DMA specification.*

5.10.7 PCI Slave DMA Configuration Registers

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

The slave DMA is only required to support at least one transfer size. The first four slave DMA channels only support 8-bit

transfers, so bits 2 and 1 always read 00b. The second four slave 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 slave 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.

5.11 ISA Bus Refresh Cycle Types

The AMD-645 Peripheral Bus 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-645 Peripheral Bus Controller 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-645 Peripheral Bus 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 SA8–SA0 by the AMD-645 Peripheral Bus Controller (regardless of which master currently owns the bus) in response to a Low REFRESH signal. The SA16–SA9 addresses are three-stated. SA19–SA17 are driven Low. MEMR is asserted by the AMD-645 Peripheral Bus Controller one BCLK cycle after REFRESH goes active. MEMR remains Low for two BCLK cycles. The REFRESH signal is negated one BCLK period after MEMR negates.

Decoupled Mode Refresh

The decoupled mode refresh enables off-board cycles to run with normal PCI bus accesses, allowing for maximum system performance. In this mode, slot bus addresses are buffered from the PCI address bus, allowing the refresh address to be driven on the ISA bus while PCI bus transactions are occurring. A refresh is requested by an OSC-based counter from the system controller block. It is arbitrated with the DMA request by sampling each line at opposite edges of BCLK and synchronizing to PCLK. When an internal refresh request is generated in decoupled mode, one of two possible sequences occurs:

1. If the AMD-645 Peripheral Bus Controller is currently busy with a pending PCI cycle request, the off-board refresh cycle is postponed until the AMD-645 PCI interface completes its current cycle to the ISA bus.
2. If the PCI interface is idle, the off-board refresh cycle will proceed immediately, and any PCI cycle request for the AMD-645 Peripheral Bus Controller that occurs during the refresh cycle is serviced after the refresh cycle completes.

5.12 ISA Bus Data Steering

Table 5-15 through Table 5-20 describe ISA bus data steering for various data paths.

Table 5-15. Data Steering: PCI Bus Master Writes

Data Type	Device Size	Data Steering	Notes
Byte, Word, Doubleword	ISA (8)	D0 -> SDL or D1 -> SDL or D2 -> SDL or D3 -> SDL	1 slot access generated per byte written (supports non-contiguous byte accesses) 2 slot accesses generated for word writes 4 slot accesses generated for doubleword writes Also, all 8-bit on-chip I/O data bus writes performed in this data steering mode
Byte	ISA (16)	D0 -> SDL or D1 -> SDH or D2 -> SDL or D3 -> SDH	1 slot access generated per byte written (supports non-contiguous byte accesses)
Word, Doubleword	ISA (16)	D0 -> SDL and D1 -> SDH or D2 -> SDL and D3 -> SDH	1 slot access generated for aligned word writes 2 slot accesses generated for aligned doubleword writes 2 slot accesses generated for misaligned word writes
Byte, Word, Doubleword	PCI (32)	N/A	All AMD-645 Peripheral Bus Controller data paths—D0, D1, D2, D3, SDL and SDH—are three-stated during PCI bus device accesses

Table 5-16. Data Steering: PCI Bus Master Reads

Data Type	Device Size	Data Steering	Notes
Byte, Word, Doubleword	ISA (8) ROM (8)	D0 -> SDL or D1 -> SDL or D2 -> SDL or D3 -> SDL On-Chip Bus -> D0, D1, D2, or D3 and On-Chip Bus -> SDL, SDH +	1 slot access generated per byte read (supports non-contiguous byte accesses) 2 slot accesses generated for word reads 4 slot accesses generated for doubleword reads D0, D1, D2, D3 data are latched before re-driving onto the PCI data bus
Byte	ISA (16)	SDL -> D0 or SDH -> D1 or SDL -> D2 or SDH -> D3	1 slot access generated per byte read (supports non-contiguous byte accesses) D0, D1, D2, D3 data are latched before re-driving onto the PCI data bus
Word, Doubleword	ISA (16)	SDL -> D0 and SDH -> D1 or SDL -> D2 and SDH -> D3	1 slot access generated for aligned word reads 2 slot accesses generated for aligned doubleword reads 2 slot accesses generated for misaligned word reads D0, D1, D2, D3 data are latched before re-driving onto the PCI data bus
Byte, Word, Doubleword	PCI (32)	N/A	All AMD-645 Peripheral Bus Controller data paths—D0, D1, D2, D3, SDL and SDH—are three-stated during PCI bus device accesses

Notes:

1. *SDH is ignored during XD bus reads.*
2. *For 8-bit on-chip I/O data bus reads, data is read from on-chip I/O data bus instead of SDL. Data is available on the appropriate PCI data bus byte—D0, D1, D2, D3 and the SDL and SDH bus.*

Table 5-17. Data Steering: DMA/ISA Bus Master Reads PCI

Data Type	Device Size	Data Steering	Notes
Byte, Word	DMA (8)	D0 -> SDL or D1 -> SDL or D2 -> SDL or D3 -> SDL	1 access required for byte reads 2 accesses required for word reads During PCI bus device accesses only, D0, D1, D2, and D3 are latched before data is driven on the slot data bus
Byte	DMA (16) MASTER (16)	D0 -> SDL or D1 -> SDH or D2 -> SDL or D3 -> SDH	1 access required for byte reads During PCI bus device accesses only, D0, D1, D2, and D3 are latched before re-driving onto the slot data bus
Word	DMA (16) MASTER (16)	D0 -> SDL and D1 -> SDH or D2 -> SDL and D3 -> SDH	1 access required for aligned word reads 2 accesses required for misaligned word reads During PCI bus device accesses only, D0, D1, D2, and D3 are latched before data is driven on the slot data bus

Table 5-18. Data Steering: DMA/ISA Bus Master Writes PCI

Data Type	Device Size	Data Steering	Notes
Byte, Word	DMA (8)	SDL -> D0 or SDL -> D1 or SDL -> D2 or SDL -> D3	1 access required for byte reads 2 accesses required for word reads
Byte	DMA (16) MASTER (16)	SDL -> D0 or SDH -> D1 or SDL -> D2 or SDH -> D3	1 access required for byte reads
Word	DMA (16) MASTER (16)	SDL -> D0 and SDH -> D1 or SDH -> D2 and SDL -> D3	1 access required for aligned word reads 2 accesses required for misaligned word reads

Table 5-19. Data Steering: DMA/ISA Bus Master Reads ISA

Data Type	Device Size	Data Steering	Notes
Byte	DMA (8)	SDH -> SDL	Byte swap required for odd byte 8-bit DMA read of 16-bit slot 1 access required for byte reads
Byte	MASTER (16)	SDL -> SDH	Byte swap required for odd byte add-on bus master read of 8-bit slot 1 access required for byte reads
Byte, Word	DMA (8) DMA (16) MASTER (16)	N/A On-Chip Bus -> SDL and SDH	Except for the above two cases and bus master reads of on-chip I/O data bus, the SDL and SDH data pads are three-stated pads for all DMA or master reads of 8-bit/16-bit ISA devices
Note: *Data is available on the SDL and SDH buses for add-on bus master reads of the 8-bit on-chip I/O data bus.*			

Table 5-20. Data Steering: DMA/ISA Bus Master Writes ISA

Data Type	Device Size	Data Steering	Notes
Byte	DMA (8) ISA (16)	SDH -> SDL	Byte swap required for odd byte add-on bus master write to 8-bit slot 1 access required for byte reads
Byte	MASTER (16) ISA (8)	SDL -> SDH	Byte swap required for odd byte 8-bit DMA write to 16-bit slot 1 access required for byte reads
Byte, Word	DMA (8) DMA (16) MASTER (16)	N/A	Except for the above two cases, the SDL and SDH data pads are input pads for all DMA or master writes to 8-bit/16-bit ISA devices All bus master writes to 8-bit on-chip I/O data bus are performed in this data steering mode

5.13 Fast IDE/EIDE Interface

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-645 Peripheral Bus Controller, 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 5-21.

Table 5-21. IDE Register Map

Channel 0	Channel 1	Type	Description
1F0h	170h	Read/Write	Data register (16-bit)
1F1h	171h	Read-Only Write-Only	Error Register (8-bit) 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 Write-Only	Status Register (8-bit) Command Register (8-bit)
3F6h	376h	Read-Only 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 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.

5.13.2 PCI Cycles

The IDE controller supports 8-bit, 16-bit, and 32-bit PCI cycles with the appropriate conversions to the 8-bit or 16-bit IDE register, as shown in Table 5-22. The IDE data register is a 16-bit register located at 1F0h or 170h. The IDE control registers are 8-bit registers located at 1F1h–1F7h and 3F6h, or at 171h–177h and 376h.

Table 5-22. PCI Cycles

PCI Cycle	IDE Register	IDE Cycle	Comments
Byte	Data	Word	The upper byte is always transferred
Byte	Control	Byte	
Word	Data	Word	
Word	Control	2 Byte	Two sequential IDE accesses are generated
Doubleword	Data	2 Word	Two IDE accesses to the Data Register are generated
Doubleword	Control	4 Byte	Four sequential IDE accesses are generated

Non-FIFO IDE Writes

When the CPU issues a write access to the IDE, the command process issues the command to the I/O process. The I/O process then waits for the address setup time to satisfy the $\overline{IOR}$/$\overline{IOW}$ precharge of the previous operation. $\overline{IOW}$ becomes active for the pre-set duration.

FIFO IDE Writes

In FIFO IDE writes, the IDE interface simply latches the data and decodes the address into the FIFO. If the FIFO is full, the IDE interface waits until the FIFO is empty due to the completion of one IDE write transfer. The IDE interface signals the PCI slave to disconnect and retry the IDE write.

Non-Read-Ahead IDE Reads

Read accesses to the IDE interface must wait until the write-FIFO is empty to ensure the proper execution order. If the write-FIFO is not empty, the read access is retried at the PCI interface and the write-FIFO is flushed. When the write-FIFO is empty, the $\overline{IOR}$ pre-charge and address setup time are satisfied and $\overline{IOR}$ becomes active for the programmed duration. Accesses to the control registers are not buffered, and any access to these addresses will invalidate data in the read-ahead buffer.

Read-Ahead IDE Reads

Read accesses to the IDE interface must wait until the write-FIFO is empty to ensure the proper execution order. If the write-FIFO is not empty, the read access is retried at the PCI interface and the write-FIFO is flushed. When the write-FIFO is empty, it issues the $\overline{IOR}$ command to the IDE, as in the case of a non-read-ahead read transfer.

If the read is not to the data register, the cycle behaves as if it is a normal non-read-ahead operation. If the read is to the data register, then the read-ahead cycle begins operating. The I/O process block issues the $\overline{IOR}$ to the IDE until the read-ahead buffer is full, without CPU intervention. If the IDE is slow enough to let the CPU catch up, the PCI $\overline{TRDY}$ is returned after the $\overline{IOR}$. In this case, read-ahead still helps since $\overline{IOR}$ starts before the CPU cycle.

Read-ahead is intended for data register reads. It counts the number of words to be transferred from the data register. However, there might be applications that transfer control data from the data port, which might not work with the prediction. The IDE interface is designed to terminate the read-ahead cycle if it senses any of the following:

- Read or write accesses to IDE control registers (any register other than the data register)
- Write access to the data register
- Read-ahead count expires (normal read-ahead termination)

In addition, a read of any I/O space register (bus mastering configuration registers) causes a retry to the PCI and a flush of the FIFO. A 32-bit pipeline register on the PCI interface buffers data to the PCI bus. A 16-bit pipeline register buffers data to the IDE interface.

5.13.3 DMA Bus Mastering

IDE DMA is supported through the PCI IDE bus mastering logic. In a typical bus master command sequence, the bus master registers are initialized with the transfer address and count. The registers then they are started, causing the PCI interface to transfer long words to or from the FIFO in 32-byte bursts. A command to the drive is then issued, which causes the drive to transfer words to or from the internal FIFO using a DRQ/$\overline{\text{DACK}}$ handshake and $\overline{\text{IOR}}$ or $\overline{\text{IOW}}$ strobes. The transfer continues until the transfer count is exhausted or until the drive generates an interrupt.

Subject to arbitration, the DMA state machine asserts $\overline{\text{DACK}}$ upon receiving DRQ. After a threshold of 30 bytes is reached, the PCI bus master requests the PCI bus. During this request, the FIFO is filled with the data from the DMA transfer. The FIFO is flushed on any access to the channel's I/O registers, or when the channel interrupt occurs. A flush in process causes PCI accesses to that channel's I/O registers (including the I/O that instigated the flush) to be retried and the IDE DMA acknowledge to be removed until the flush is completed. An enabled interrupt from the IDE device (hard drive or CD ROM drive) is routed to the FIFO and causes the flush at the end of a DMA access. The interrupt is routed to the status register after the flush of the FIFO occurs. The DMA state machine must also determine whether an access is single-word or multi-word DMA. It does this by the programmed pulse timing.

The FIFO must be emptied before accepting any I/O access to the bus mastering IDE controller. Any PCI access to the IDE channel's I/O registers causes a flush of the channel's DMA

FIFO. PCI accesses to the channel's I/O registers are retried until the channel FIFO is flushed. PCI retries cause the IDE to re-issue the request with the retried address and data. PCI disconnects cause the IDE to re-issue the request with the next address and data.

An IDE arbiter state machine is required to handle all channel activity. The arbiter must arbitrate between the two channels. The PIO requests have priority over any DMA requests. Also, the primary channel has a higher priority than the secondary channel. The PCI arbiter must respond to the IDE arbiter for its arbitration scheme. The PCI arbiter must arbitrate between ISA DMA and IDE DMA requests. The ISA DMA has priority over IDE.

Each IDE channel has a bi-directional 32-byte FIFO. Only DMA accesses are put in this FIFO. The direction of the FIFO is controlled by registers. For PCI bus mastering DMA accesses, the bus master command and status registers determine the direction of the FIFO. Both channels cannot operate over the IDE interface simultaneously due to the 16-bit IDE data bus shared between two channels. Note, however, that a channel's FIFO may be connected to the PCI data bus while the other channel's FIFO is connected to the IDE data bus.

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

1. Software prepares a physical region descriptor (PRD) table in system memory. Each PRD is 8 bytes long and consists of an address pointer to the starting address and the transfer count of the memory buffer to be transferred. In any given PRD table, two consecutive PRDs are offset by eight bytes and are aligned on a 4-byte boundary.
2. Software provides the starting address of the PRD table by loading the PRD table pointer register. The direction of the data transfer is specified by setting the Read/Write Control bit. Clear the Interrupt bit and the Error bit in the Status register.
3. Software issues the appropriate DMA transfer command to the disk device.
4. Engage the bus master function by writing a "1" to the Start bit in the Bus Master IDE Command Register for the appropriate channel.

5. The controller transfers data to or from memory, responding to DMA requests from the IDE device.
6. At the end of the transfer the IDE device signals an interrupt. This interrupt is generated as ISA interrupt 14 for the primary channel or as ISA interrupt 15 for the secondary channel.
7. In response to the interrupt, software resets the Start/Stop bit in the Master Command register, then reads the controller status and drive status to determine whether the transfer completed successfully.

The physical memory transfer region is described by a physical region descriptor (PRD). The data transfer proceeds until all regions described by the PRDs in the table are transferred. Each PRD entry is eight bytes long. The first four bytes specify the byte address of a physical memory region. The next two bytes specify the count of the region in bytes, with a 64-Kbyte limit per region. A value of zero in these two bytes indicates 64 Kbytes. Bit 7 of the last byte indicates the end of the table.

Bus master operation terminates when the last descriptor has been retired.

5.13.4 IDE Channel Arbitration

The IDE channel arbiter controls the IDE data and address paths between the two IDE channels. The arbiter must determine which channel already has access to the bus and what type of access is occurring. On DMA accesses, the data bus is controlled and the address bus is not. For PIO accesses, both the data and address buses are controlled.

The arbiter grants data bus accesses to the first IDE channel to access the bus. Once the IDE bus has been granted, only a PIO request from the other channel for the IDE bus will cause a removal of the grant. Otherwise, the bus will not be re-arbitrated until the access is finished. For large data transfers, this procedure allows the data transfer to complete without an interruption.

PIO Accesses

The IDE arbiter monitors the address decode logic of each channel to determine when there is an access. On access, the data and address buses will be steered to the channel where

the access occurred. The PIO access will be retried if the DMA FIFO is not empty, or if a $\overline{\text{DACK}}$ is active. The PIO access causes a flush of the DMA FIFO if it is not empty.

DMA Accesses

The IDE arbiter monitors the DMA request from the drives. When the DRQ from a drive is detected, its channel receives the data bus. When the DRQ is de-asserted, the IDE arbiter re-arbitrates for the IDE data bus. If a DMA access from a channel is in process during a PIO request from the same channel, the PCI bus access to the IDE will end in a retry. If a DMA access from a channel is in process during a PIO request from the other channel, the DMA grant is removed and the PCI bus access to the IDE ends in a retry with a delayed transaction implemented internally. The IDE arbiter notifies the PCI bus to retry the cycle.

The PCI arbiter follows a fixed priority:

1. PIO access to the primary channel
2. PIO access to the secondary channel
3. DMA request from the primary channel
4. DMA request from the secondary channel
5. Bus master controller accesses to memory.

5.13.5 Additional Features

Some of the additional IDE/EIDE features of the AMD-645 Peripheral Bus Controller are discussed in the following paragraphs.

IDE Reset Buffering

The AMD-645 Peripheral Bus Controller provides the buffered reset signal to the IDE drive, removing the need for motherboard logic to perform this function. The signal is driven Low during reset to the AMD-645 Peripheral Bus Controller, or when the proper bit of the Primary Channel Configuration register is set.

Interrupt Routing

The interrupt from the IDE drive is routed to the AMD-645 Peripheral Bus Controller. Two potential interrupt sources are made available to each IDE channel. One is a Plug-N-Play (PNP) interrupt and the other is the ISA IRQ input. The interrupt source is selected with the IDE Configuration register, Function 1, offset 9h, such that ISA Compatibility

Mode or Native PCI Mode can be selected. If the ISA IRQ is selected, the Interrupt Routing register in Function 0, offset 4Ah can be used to select the IDE interrupt source. The primary channel uses IRQ14 and can be set to IRQ15, IRQ10, or IRQ11, while the secondary channel uses IRQ15 and can be set to IRQ14, IRQ10, or IRQ11.

If the ISA IRQ is selected, the PNP IRQ has no effect on the IDE IRQ output. If the PNP IRQ is selected, the IDE interrupt output is ANDed with the ISA IRQ. If the IDE interrupt is disabled, the ISA IRQ is passed through with no change. This configuration allows the option of interrupt sharing on the IDE channel's interrupt level.

5.13.6 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.

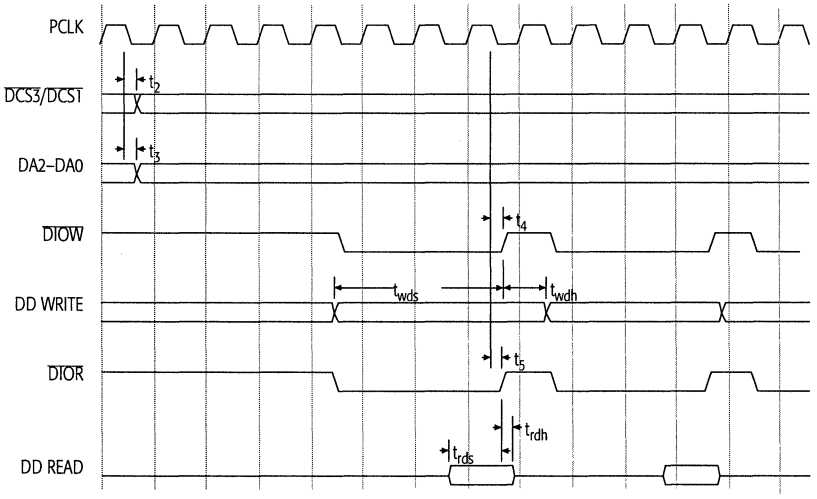


Figure 5-27. PIO Cycle

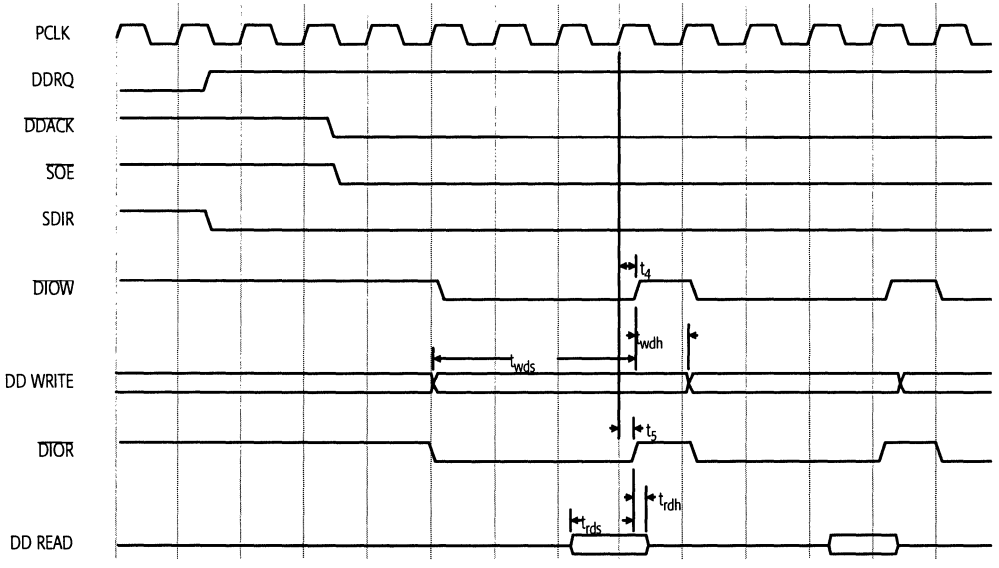
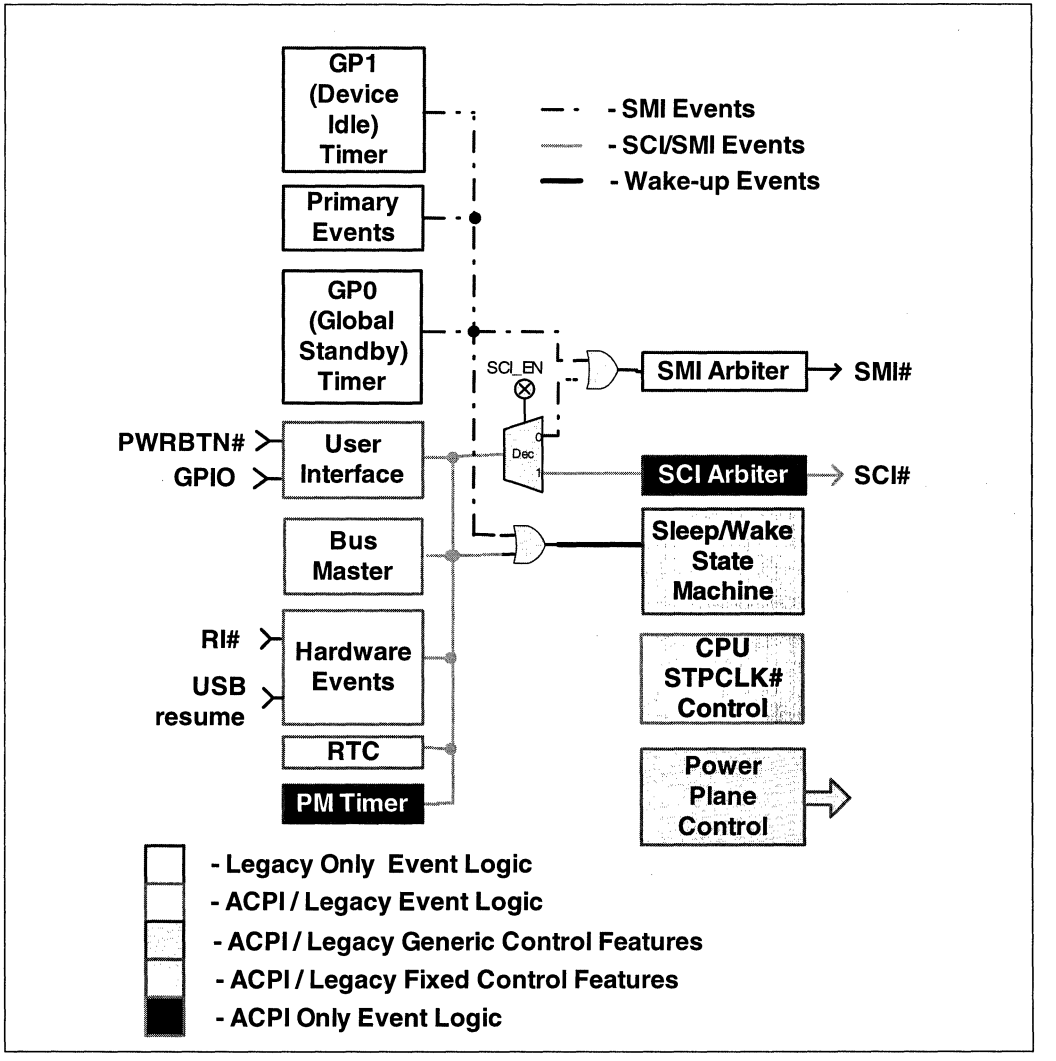


Figure 5-28. IDE Multiword DMA Cycle

5.14 Power Management Support

5.14.1 Power Management Subsystem

The power management function of the AMD-645 Peripheral Bus Controller is indicated in the following block diagram.



5.14.2 Power Plane Management

There are three power planes inside the AMD-645 Peripheral Bus Controller. This scheme is optimal for systems with ATX power supplies, although it also works using non-ATX power supplies. The key feature of the ATX power supply is the

availability of two sets of power sources. The first set is always on unless turned off by the mechanical switch. Only one voltage (5 V) is available for this set. The second set includes the normal 5 V and 12 V power supplies and is controlled by the input signal PWRON as well as a mechanical switch. This set of voltages is available only when both the mechanical switch is on and the PWRON signal is High. The power planes powered by the above two sets of supplies are referred to as V_{DD}-5VSB and V_{DD}, respectively. In addition to the two power planes, a third plane is powered by the combination of 5VSB and VBAT for the integrated real time clock. Most of the circuitry inside the AMD-645 Peripheral Bus Controller is powered by V_{DD}. Very little logic is powered by V_{DD}-5VSB and it remains functional as long as the mechanical switch of the power supply is turned on. The main function of this logic is to control the power supply of the V_{DD} plane.

General Purpose I/O Ports

As ACPI-compliant hardware, the AMD-645 Peripheral Bus Controller includes $\overline{\text{PWRBTN}}$ (pin 91) and $\overline{\text{RI}}$ (pin 93) pins to implement power button and ring indicator functionality. In addition, a PWRON pin (pin 107) is also available to control the V_{DD} power plane by V_{DD}-5VSB-powered logic. Furthermore, the AMD-645 Peripheral Bus Controller offers many general purpose I/O ports with the following capabilities:

- I^2C support
- Three GPIO ports without external logic in addition to the I^2C port. Five GPIO ports are available if I2C functionality is not used. Every port can be used as inputs, outputs or I/O with external SCI/SMI capabilities.
- Sixteen GPI and sixteen GPO pins using external buffers (244 buffers for input and 373 latches for output)

Pins 87, 88, and 94 of the AMD-645 Peripheral Bus Controller are dedicated general purpose I/O pins that can be used as inputs, outputs, or I/O with external SMI capability. In particular, pins 87 and 88 can be used to implement a software-implemented I^2C port for system configuration and general purpose peripheral communication. Pins 92 and 136 can be configured either as dedicated general purpose I/O pins or as control signals for external buffers for implementing up to sixteen GPI and sixteen GPO ports. The GPI and GPO ports are connected to SD15–SD8 and XD7–0SD0. The configuration is

determined in the GPIO4_CFG and CPIO3_CFG bits of the PIN_CFG register.

GPIO4_CFG defaults to 1 to define pin 136 as GPIO4. Clear GPIO4_CFG to redefine the pin as GPO_WE latch enable.

GPIO3_CFG defaults to 1 to define pin 92 as GPIO3. Clear GPIO3_CFG to redefine the pin as $\overline{\text{GPI\_RE}}$ buffer enable.

5.14.3 Power Management Events

Three types of power management events are supported:

1. ACPI-required fixed events defined in the PM1a_STS and PM1a_EN registers. These events can trigger the following SCI or SMI events depending on the SCI_EN bit:
   - $\overline{\text{PWRBTN}}$ Triggering
   - RTC alarm
   - ACPI power management timer carry (always SCI)
   - BIOS release (always SCI)
2. ACPI-aware general purpose function events defined in GP_STS and GP_SCI_EN, and GP_SMI_EN registers. These events can trigger the following SCI or SMI events depending on the setting of individual SMI and SCI enable bits:
   - EXTSMI triggering
   - USB resume
   - $\overline{\text{RI}}$ indicator
3. Generic global events defined in the GBL_STS and GBL_EN registers. These registers are used primarily for the following SMI events:
   - GP0 and GP1 timer time out
   - Secondary event timer time out
   - Occurrence of primary events (defined in register PACT_STS and PACT_EN)
   - Legacy USB accesses (keyboard and mouse)

Once enabled, each of the EXTSMI inputs triggers an SCI or SMI at either the rising or falling transition of the corresponding input pin signal. Software can check the status

of the input pins via register EXTSMI_VAL and take proper actions.

Among many possible actions, the SCI and SMI routine can change the processor state by programming the P_BLK registers. The routine can also set the SLP_EN bit to put the system into one of the following two suspend states:

1. Suspend to Disk (or Soft-Off)—The V_{DD} power plane is turned off while V_{DD}-5VSB and V_{DD}-RTC planes remain on.
2. Power-On-Suspend—All power planes remain on but the processor is put in the C3 state.

In either suspend state, there is minimal interface between powered and non-powered planes.

The AMD-645 Peripheral Bus Controller allows the following events to wake up the system from the two suspend states and from the C2 state to the normal working state (processor in C0 state):

- Activation of External Inputs—$\overline{\text{PWRBTN}}$, $\overline{\text{RI}}$, GPIO0 and other EXTSMI pins (see table below)
- RTC Alarm and ACPI Power Management Timer—(see table below)
- USB Resume Event—(see Table 5-23)
- Interrupt Events—Always resume independent of any register setting
- ISA Master or DMA Events—Always resume independent of any register setting

The AMD-645 Peripheral Bus Controller also provides flexible SCI/SMI steering and PWRON control for the events listed in Table 5-23.

Table 5-23. SCI/SMI/Resume Control for PM Events

Event	Global SCI/SMI Control	Individual Enable Bits for SCI & SMI	Separate Control for PWRON Resume
PWRBTN	SCI_EN bit	N	Y
RI	N	Y	Y
RTC Alarm	N	Y	N
GP1O0 (EXTSMI0)	N	Y	Y
External SCI/SMI (non-GPIO0)	N	Y	Y
ACPI PM Timer	Always SCI	N	N
USB Resume	N	N	Y

Table 5-24 shows the availability of resume events in each type of suspend state.

Table 5-24. Suspend Resume Events and Conditions

Input Trigger	Power Plane	Soft-Off	Power-On Suspend
$\overline{\text{PWRBTN}}$	V_{DD}-5VSB	yes	yes
RI	V_{DD}-5VSB	yes	yes
RTC alarm	VBAT	yes	yes
GP1O0 ($\overline{\text{EXTSMI0}}$)	V_{DD}-5VSB	yes	yes
External SCI/SMI (non-GPIO0)	V_{DD}-5V	no	yes
ACPI PM timer	V_{DD}-5V	no	yes
USB resume	V_{DD}-5V	no	yes
PCI/ISA interrupts	V_{DD}-5V	no	yes
PCI/ISA master/DMA	V_{DD}-5V	no	yes

5.14.4 Legacy Management Timers

In addition to the ACPI power management timer, the AMD-645 Peripheral Bus Controller includes the following four legacy power management timers:

- GP0 Timer—General purpose timer with primary event

- GP1 Timer—General purpose timer with peripheral event reload
- Secondary Event Timer—To monitor secondary events
- Conserve Mode Timer—Not used in desktop applications

The normal sequence of operations for a general purpose timer (GP0 or GP1) is as follows:

1. Program the time base and timer value of the initial count (register GP_TIM_CNT).
2. Activate counting by setting the GP0_START or GP1_START bit to one: the timer will start with the initial count and count down towards 0.
3. When the timer counts down to zero, an SMI will be generated if enabled (GP0TO_EN and GP1TO_EN in the GBL_EN register) with status recorded (GP0TO_STS and GP1TO_STS in the GBL_STS register).
4. Each timer can also be programmed to reload the initial count and restart counting automatically after counting down to 0. This feature is not used in standard BIOS.

The GP0 and GP1 timers can be used just as the general purpose timers described above. However, they can also be programmed to reload the initial count by system primary events or peripheral events thus used as the primary event (global standby) timer and peripheral timer, respectively. The secondary event timer is solely used to monitor secondary events.

5.14.5 System Primary and Secondary Events

Primary system events are distinguished in the PRI_ACT_STS and PRI_ACT_EN registers. The bit controls in these registers are summarized in Table 5-25.

Table 5-25. PRI_ACT_STS and PRI_ACT_EN Register Bits

Bit	Event	Trigger
7	Keyboard Access	I/O port 60h
6	Serial Port Access	I/O ports 3F8h-3FFh, 2F8h-2FFh, 3E8h-3EFh, or 2E8h-2EFh
5	Parallel Port Access	I/O ports 378h-37Fh or 278h-27Fh
4	Video Access	I/O ports 3B0h-3DFh or memory A/B segments
3	IDE/Floppy Access	I/O ports 1F0h-1F7h, 170h-177h, or 3F5h
2	Reserved	
1	Primary Interrupts	Each channel of the interrupt controller can be programmed as a primary or secondary interrupt
0	ISA Master/DMA Activity	

Each category can be enabled as a primary event by setting the corresponding bit of the PRI_ACT_EN register. If enabled, the occurrence of the primary event reloads the GP0 timer if the PACT_GP0_EN bit is also set. The cause of the timer reload is recorded in the corresponding bit of the PRI_ACT_STS register while the timer is reloaded. If no enabled primary event occurs during the count down, the GP0 timer will time out (count down to 0) and the system can be programmed (setting the GP0TO_EN bit in the GBL_EN register to one) to trigger an SMI to switch the system to a power down mode.

The AMD-645 Peripheral Bus Controller distinguishes two kinds of power management interrupt requests, primary and secondary interrupts. Like other primary events, the occurrence of a primary interrupt demands that the system be restored to full processing capability. Secondary interrupts are typically used for background housekeeping tasks that are unnoticeable to the user. The AMD-645 Peripheral Bus Controller allows each channel of interrupt request to be declared as either primary, secondary, or ignorable in the PIRQ_CH and SIRQ_CH registers. Secondary interrupts are the only system secondary events defined in the AMD-645 Peripheral Bus Controller.

Like primary events, primary interrupts can be made to reload the GP0 timer by setting the PIRQ_EN bit to 1. Secondary interrupts do not reload the GP0 timer. Therefore, the GP0 timer will time out and the SMI routine can put the system into power down mode if no events other than secondary interrupts occur periodically in the background.

Primary events can be programmed to trigger an SMI (setting of the PACT_EN bit). Typically, this SMI triggering is turned off during normal system operation to avoid degrading system performance. Triggering is turned on by the SMI routine before entering the power down mode so that the system may be returned to normal operation at the occurrence of primary events. At the same time, the GP0 timer is reloaded and the count down process is restarted.

5.14.6 Peripheral Events

Primary and secondary events define system events in general, and the response is typically expressed in terms of system events. Individual peripheral events can also be monitored by the AMD-645 Peripheral Bus Controller through the GP1 timer. The following four categories of peripheral events are distinguished (via register GP_RLD_EN):

- Bit 7—Keyboard access
- Bit 6—Serial Port access
- Bit 4—Video access
- Bit 3—IDE/Floppy access

The four categories are subsets of the primary events as defined in PRI_ACT_EN, and the occurrence of these events can be checked through a common register PRI_ACT_STS. As a peripheral timer, GP1 can be used to monitor one (or more than one) of the above four device types by programming the corresponding bit to one and the other bits to zero. Timeout of the GP1 timer indicates no activity of the corresponding device type and appropriate action can be taken as a result.

6 Initialization

All programmable features in the AMD-645 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 Section 7.

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 6-1. Master 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 6-2. 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 6-3. 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 6-4. 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 6-5. CMOS/RTC/NNI 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 6-6. 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 6-7. System Control Registers

Port	Register Name	Access
92h	System Control	RW

Table 6-8. 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 6-9. Slave 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

6.2 PCI Function 0 Registers—PCI-to-ISA Bridge

Table 6-10. Configuration Space PCI-to-ISA Header Registers

Offset	PCI Header	Default	Access
01h–00h	Vendor ID	1106h	RO
03h–02h	Device ID	0586h	RO
05h–04h	Command	000Fh	RW
07h–06h	Status	0200h	RWC
08h	Revision ID (00h = first silicon)	–	RO
09h	Program Interface	00h	RO
0Ah	Sub Class Code	01h	RO
0Bh	Base Class Code	06h	RO
0Ch	Reserved (Cache Line Size)	00h	–
0Dh	Reserved (Latency Timer)	00h	–
0Eh	Header Type	80h	RO
0Fh	Built-In Self Test (BIST)	00h	RO
10h–3Fh	Reserved	00h	–

Table 6-11. ISA Bus Control Registers

Offset	Register	Default	Recommended		Access
			Setting	Result	
40h	ISA Bus Control	00h	00h	Normal ISA timing	RW
41H	ISA Test Mode	00h	01h	Refresh test mode	RW
42h	ISA Clock Control	00h	00h	ISA clock=PCICLK/4	RW
43h	ROM Decode Control	00h	00h	$\overline{\text{ROMCS}}$ 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	10h	Disable post memory write	RW
47h	Miscellaneous Control 2	00h	C0h	INIT as CPU reset Enable PCI delay transaction	RW
48h	Miscellaneous Control 3	01h	01h	Enable USB, IDE	RW
49h	Reserved	00h	00h		–
4Ah	IDE Interrupt Routing	04h	C4h	Wait for PGNT before grant to ISA master/DMA Access ports 00-FFh via SD IDE primary channel IRQ14 Secondary channel IRQ 15	RW
4Bh	Reserved	00h	00h		–
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
4Fh–4Eh	DMA/Master Mem Access Ctrl 3	0300h	F300h	Top of PCI memory for ISA=16M. Forward 00000h-9FFFFh access to PCI	RW

Table 6-12. Plug-n-Play Control Registers

Offset	Register	Default	Recommended		Access
			Setting	Result	
50h	Reserved (do not program)	24h	24h		RW
53h–51h	Reserved	00h	00h		–
54h	PIC IRQ Edge/Level Selection	00h	00h	PIRQs inverted edge trigger/ Non-inverted level trigger	RW
55h	PnP Routing for External MIRQ0–1	00h	00h	MIRQs disabled	RW
56h	PnP Routing for PCI INTB–A	00h	B0h	INTB routes to IRQ11 INTA disabled	RW
57h	PnP Routing for PIC INTD–C	00h	57h	INTD routes to IRQ5 INTC routes to IRQ7	RW
58h	PnP Routing for External MIRQ2	00h	00h	MIRQ2 disabled	RW
59h	MIRQ Pin Configuration	04h	04h	Configure as $\overline{\text{MASTER}}$	RW
5Ah	XD Power-On Strap Options	Note 1	F7h	Enable Int RTc, PS2 mouse, Int KBC	RW
5Bh	Internal RTC Test Mode	00h	00h	RTC reset enable, SRAM access enable, test enable	RW
5Ch–5Fh	Reserved	00h	00h		–

Notes:
Power-up default value depends on external strapping

Table 6-13. 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	0000h	0000h	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
70h–FFh	Reserved	00h	00h		–

6.3 PCI Function 1 Registers—IDE Control

Table 6-14. Configuration Space IDE Header Registers

Offset	PCI Header	Default	Access
01h–00h	Vendor ID	1106h	RO
03h–02h	Device ID	0571h	RO
05h–04h	Command	0080h	RW
07h–06h	Status	0280h	RW
08h	Revision ID (00h = first silicon)	–	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	20h	RW
0Eh	Header Type	00h	RO
0Fh	Built-In Self Test (BIST)	00h	RO
13h–10h	Base Address–Primary Data/Command	0000_01F0h	RW
17h–14h	Base Address–Primary Control/Status	0000_03F4h	RW
1Bh–18h	Base Address–Secondary Data/Command	0000_0170h	RW
1Fh–1Ch	Base Address–Secondary Control/Status	0000_0374h	RW
23h–20h	Base Address–Bus Master Control	0000C_C01h	RW
24h–2Fh	Reserved (unassigned)	00h	–
30h–33h	Reserved (expansion ROM base address)	00h	–
34h–3Ch	Reserved (unassigned)	00h	–
3Ch	Interrupt Lines	0Eh	RW
3Dh	Interrupt Pin	00h	RO
3Eh	Minimum Grant	00h	RO
3Fh	Maximum Latency	00h	RO

Table 6-15. Configuration Space IDE Registers

Offset	Register	Default	Recommended		Access
			Setting	Result	
40h	Chip Enable	04h	0Bh	Enable pri and sec channel	RW
41h	IDE Configuration	02h	E2h	Enable pri and sec read prefetch buffer Enable pri post write buffer	RW
42h	Reserved (do not program)	09h	09h		RW
43h	FIFO Configuration	3Ah	3Ah	Allocate 8 word buffers in both pri and sec channel Set threshold to 1/2	RW
44h	Miscellaneous Control 1	68h	68h	Master Read/Write cycle $\overline{IRDY}$ 1 wait state FIFO output data 12 clock advance	RW
45h	Miscellaneous Control 2	00h	00h	No channel interrupts swap	RW
46h	Miscellaneous Control 3	C0h	C0h	Pri and Sec Ch Read DMA FIFO flush enabled No limit in DRDY pulse width	RW
4Bh–48h	Drive Timing Control	A8A8A8A8h	A8A8A8A8h	$\overline{DIOR}$ and $\overline{DIOW}$ pulse width set to 11 PCI clocks Recovery time set to 9 clocks	RW
4Ch	Address Setup Time	FFh	FFh	Address setup time 4T	RW
4Dh	Reserved (do not program)	00h	00h		RW
4Eh	Sec Non-1F0h Port Access Timing	FFh	FFh	Sec non-1F0 port access, $\overline{DIOR}$ and $\overline{DIOW}$ pulse width set to 17 PCI clocks	RW
4Fh	Pri Non-1F0h Port Access Timing	FFh	FFh	Pri non-1F0 port access, $\overline{DIOR}$ and $\overline{DIOW}$ pulse width set to 17 PCI clocks	RW
53h–50h	UltraDMA33 Extd Timing Control	03030303h	03030303h	Pri and sec Drive 0 and 1Mode enabled by Set Feature command Disabled UltraDMA33-mode	RW
57h-54h	Reserved	00h	00h		
5Fh–58h	Reserved	A8A8A8A8h	A8A8A8A8h		–
61h–60h	Primary Sector Size	0200h	0200h	200h bytes per sector	RW
67h–62h	Reserved	00h	00h		–
69h–68h	Secondary Sector Size	0200h	0200	200h bytes per sector	RW
6Ah–FFh	Reserved	00h	00		–

Table 6-16. IDE Controller I/O Registers

Offset	Register Name	Default	Access
00h	Primary Channel Command	00h	RW
01h	Reserved	00h	–
02h	Primary Channel Status	00h	RWC
03h	Reserved	00h	–
07h–04h	Primary Channel PRD Table Address	00h	RW
08h	Secondary Channel Command	00h	RW
09h	Reserved	00h	–
0Ah	Secondary Channel Status	00h	RWC
0Bh	Reserved	00h	–
0Fh–0Ch	Secondary Channel PRD Table Address	00h	RW

6.4 PCI Function 2 Registers–USB Controller

Table 6-17. Configuration Space USB Header Registers

Offset	PCI Header	Default	Access
01h–00h	Vendor ID	1106h	RO
03h–02h	Device ID	3038h	RO
05h–04h	Command	0000h	RW
07h–06h	Status	0200h	RWC
08h	Revision ID (00h = first silicon)	–	RO
09h	Program Interface	00h	RO
0Ah	Sub Class Code	03h	RO
0Bh	Base Class Code	0Ch	RO
0Ch	Reserved (Cache Line Size)	00h	RO
0Dh	Latency Timer	16h	RW
0Eh	Header Type	00h	RO
0Fh	Built-In Self Test (BIST)	00h	RO
10h–1Fh	Reserved	00h	–
23h–20h	Base Address	0000301h	RW
24h–3Bh	Reserved	00h	–
3Ch	Interrupt Line	00h	RW
3Dh	Interrupt Pin	04h	RW
3Eh–3Fh	Reserved	00h	–

Table 6-18. Configuration Space USB Registers

Offset	Register	Default	Recommended		Access
			Setting	Result	
40h	Miscellaneous Control 1	00h	00h	Support MRL, MRM, MWI ISB Data length 1280 Disable USB power management DMA 16 DW burst access PCI zero wait state	RW
41h	Miscellaneous Control 1	00h	00h	Always set trap 60/64 status bit A20GATE pass through	RW
42h–43h	Reserved	00h	00h		RO
44h–46h	Reserved (do not program)	00C2h	00C2h		RW
47h	Reserved	0Ch	0Ch		
48h–5Fh	Reserved	00h	00h		
60h	Serial Bus Release Number	10h	10h	Always read 10h	RO
61h–BFh	Reserved	00h	00h		
C1h–C0h	Legacy Support	2000h	2000h	Always read 2000h	RO
C2h–FFh	Reserved	00h	00h		

Table 6-19. USB Controller I/O Registers

Offset	Register Name	Default	Access
01h–00h	USB Command	0000h	RW
03h–02h	USB Status	0000h	RWC
05h–04h	USB Interrupt Enable	0000h	RW
07h–06h	Frame Number	0000h	RW
0Bh–08h	Frame List Base Address	00000000h	RW
0Ch	Start of Frame Modify	40h	RW
11h–10h	Port 1 Status/Control	0080h	RWC
13h–12h	Port 2 Status/control	0080h	RWC

6.5 PCI Function 3 Registers—Power Management

6.5.1 Power Management Configuration Space Registers

Table 6-20. Configuration Space Power Management Header Registers

Offset	PCI Header	Default	Access
01h–00h	Vendor ID	1106h	RO
03h–02h	Device ID	3040h	RO
05h–04h	Command	0000h	RW
07h–06h	Status	0280h	RWC
08h	Revision ID (00h = first silicon)	–	RO
09h	Program Interface	00h	RO
0Ah	Sub Class Code	00h	RO
0Bh	Base Class Code	00h	RO
0Ch	Reserved	00h	RO
0Dh	Latency Timer	16h	RW
0Eh	Header Type	00h	RO
0Fh	Built-In Self Test (BIST)	00h	RO
10h–1Fh	Reserved	00h	–
23h–20h	I/O Register Base Address	00000001h	RW
24h–3Fh	Reserved	00h	–

Table 6-21. Configuration Space Power Management Registers

Offset	Register	Default	Recommended		Access
			Setting	Result	
40h	Pin Configuration	C0h	C0h	Define pin 136 as GPIO4 Define pin 92 as GPIO4	RW
41h	General Configuration	00h	00h	Disable PWRBTN debounce Disable ACPI timer reset ACPI 24-bit timer count 32us clock throttling	RW
42h	SCI Interrupt Configuration	00h	00h	Disable SCI interrupt	RW
43h	Reserved	00h	00h		RW
45h–44h	Primary Interrupt Channel	0000h	0000h	Disable pri interrupt channel	RW
47h–46h	Secondary Interrupt Channel	0000h	0000h	Disable sec interrupt channel	RW
53h–50h	GP Timer Control	00000000h	00000000h	Disable conserve mode Disable sec event time Disable GP1 timer Disable GP0 timer	RW
54h–60h	Reserved	00h	00h		–
61h	Programming Interface Read Value	00h	00h	Value to be returned by register at offset 09h	WO
62h	Sub Class Read Value	00h	00h	Value to be returned by register at offset 0Ah	WO
63h	Base Class Read Value	00h	00h	Value to be returned by register at offset 0Bh	WO
64h–FFh	Reserved	00h	00h		–

6.5.2 Power Management I/O Space Registers

Table 6-22. Basic Power Management Control/Status Registers

Offset	Register Name	Default	Access
01h–00h	Power Management Status	00h	RWC
03h–02h	Power Management Enable	00h	RW
05h–04h	Power Management Control	00h	RW
0Bh–08h	Power Management Timer	00h	RW

Table 6-23. 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

Table 6-24. General Purpose Power Management Registers

Offset	Register Name	Default	Access
21h–20h	General Purpose Status	00h	RWC
23h–22h	General Purpose SCI Enable	00h	RW
25h–24h	General Purpose SMI Enable	00h	RW
27h–26h	Power Supply Control	00h	RW

Table 6-25. Generic Power Management Registers

Offset	Register Name	Default	Access
29h–28h	Global Status	00h	RWC
2Bh–2Ah	Global Enable	00h	RW
2Dh–2Ch	Global Control	00h	RW
2Fh	SMI Command	00h	RW
33h–30h	Primary Activity Detect Status	00h	RWC
37h–34h	Primary Activity Detect Enable	00h	RW
3Bh–38h	GP Timer Reload Enable	00h	RW

Table 6-26. General Purpose I/O

Offset	Register Name	Default	Access
40h	GPIO Direction Control	00h	RW
42h	GPIO Port Output Value	00	RW
44h	GPIO Port Input Value	input	RO
47h–46h	GPO Port Output Value	0000	RW
49h–48h	GPI Port Input Value	input	RO

7 Registers

This section summarizes the AMD-645 Peripheral Bus Controller configuration and I/O registers. Where applicable, they also document the power-on default value and access type for each register.

Access type definitions are as follows:

- RW (Read/Write)
- RO (Read Only)
- WO (Write Only)
- "—" Reserved
- RWC (Read, Write 1's to Clear individual bits)

Registers indicated as RW may have some read-only bits that always read back a fixed value (usually 0 if unused). Registers designated as RWC may have some read-only or read-write bits (see individual register descriptions for details).

7.1 PCI Mechanism #1

The AMD-645 Peripheral Bus Controller uses PCI configuration mechanism #1 to convey and receive configuration data to and from the host processor. This mechanism, described in *PCI Local Bus Specification Revision 2.1*, employs I/O locations 0CF8h to 0CFBh to specify the target address and locations 0CFCh to 0CFFh for data to the target address. The target address includes the specific PCI bus, device, function number, and register number within a PCI device.

Configuration Address Ports 0CFBh–0CF8h

31	bit 30 – bit 24	bit 23 – bit 16	bit 15 – bit 11	10 – 8	bit 7 – bit 2	1	0
En	Reserved	Bus Number	Device Number	Function #	Register Number	0	0
I/O Address OCFBh		I/O Address OCFAh	I/O Address OCF9h		I/O Address OCF8h		

Configuration Address is a read-write port that responds only to doubleword accesses. Byte or word accesses are passed on unchanged.

Bit 31 **Configuration Space Enable**
1 = The targeted PCI device responds.
0 = The I/O access is passed on unchanged.

Bits 30–24 **Reserved** (always reads 0)

Bits 23–16 **PCI Bus Number** - These bits are used to choose a specific system PCI bus.

Bits 15–11 **Device Number** - These bits are used to choose a specific system device.

Bits 10–8 **Function Number** - These bits are used to choose the number of a specific function space in memory.

Bits 7–2 **Register Number** - These bits are used to specify the offset number of a register within the chosen function space. The register number is a doubleword which, in conjunction with the PCI byte enable lines C/$\overline{BE}$3–C/$\overline{BE}$0, specifies the configuration register offset number.

Bits 1–0 **Fixed** (always reads 0)

Configuration Data Ports 0CFCh–0CFFh

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
xxxxxxxxB																															

Configuration Data is a read-write port that responds only to doubleword accesses. Byte or word accesses will be passed on unchanged.

7.2 Legacy I/O Registers

This group of I/O registers includes keyboard and mouse control, 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 a precise manner for backwards compatibility with previous generations of PC hardware.

These registers are listed for reference only. Detailed descriptions of the actions and programming of these registers are included in other industry publications. All of the registers reside in I/O space. They are grouped according to their AMD-645 Peripheral Bus Controller functions. The I/O port address and access type are given for each register.

7.2.1 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.

Keyboard/Mouse Status Port 64h **RO**

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	PE	GRT	MOB	KS	CD	SF	IB	KOB
Reset	0	0	0	0	0	0	0	0

Bit 7 **Parity Error**
1 = A parity error occurred on the last byte received from keyboard/mouse
0 = No parity error (odd parity received = default)

Bit 6 **General Receive/Transmit Timeout**
1 = Error
0 = No error (default)

Bit 5 **Mouse Output Buffer Full**
1 = Mouse output buffer full
0 = Mouse output buffer empty (default)

Bit 4 **Keylock Status**
1 = Free
0 = Locked (default)

Bit 3 **Command/Data**
1 = Last write was command write
0 = Last write was data write (default)

Bit 2 **System Flag**
1 = Self Test Successful
0 = Power-On (default)

Bit 1 **Input Buffer Full**
1 = Keyboard input buffer full
0 = Keyboard input buffer empty (default)

Bit 0 **Keyboard Output Buffer Full**
1 = Keyboard output buffer full
0 = Keyboard output buffer empty (default)

Keyboard/Mouse Command Port 64h WO

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Value of Specific Keyboard Command (See Table 10-1)							
Reset	0	0	0	0	0	0	0	0

Port 64h—Keyboard/Mouse Command—is a write-only I/O register. This register, when written, is used to send commands to the keyboard/mouse controller. Keyboard/mouse command codes recognized by the AMD-645 Peripheral Bus Controller are listed in Table 7-1.

Note that 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 if P13–P16 are set during power-up as strapping options. That is, many of the commands that follow are provided and work, but otherwise perform no useful function (e.g., commands that set P12–P17 high or low). Also note that 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.

Table 7-1. 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 low nibble (bits 0–3) to input ports P10–P13
A1h	Output Keyboard Controller Version #
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) 0 = OK, 1 = Clock stuck Low, 2 = Clock stuck High, 3 = Data stuck Low, 4 = Data stuck High, FF = General error
AAh	KBC Self Test (55h = OK, FCh = Not OK)
ABh	Keyboard Interface Test (results in port 60h) 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 #
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 P11–P13 repeatedly and put results in bits 5–7 of status register)

Table 7-1. Keyboard Controller Command Codes (continued)

Command Code	Keyboard Command Code Description
C2h	Poll Input Port High (read input data on input ports P15–P17 repeatedly and put results in bits 5–7 of status register)
C8h	Unblock P22–P23 (use before command D1 to change the active mode)
C9h	Reblock P22–P23 (protection mechanism for D1 command)
CAh	Read Mode (output KBC mode info to port 60 output buffer) bit 0 = 0 = ISA, bit 0 = 1 = PS/2
D0h	Read Output Port (copy P10–P17 output values to port 60h)
D1h	Write Output Port (data byte following is written to keyboard output port as if it came from the keyboard)
D2h	Write Keyboard Output Buffer & clear status bit 5 (write following byte to keyboard)
D3h	Write Mouse Output Buffer & 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)
E0h	Read Test Inputs (T0–T1 read to bits 0–1 of respective byte)
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:
Codes not listed are undefined or their functions are eliminated by direct control of the keyboard controller logic.

KBC Control Register Port 60h or 64h RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Reserved	PCC	MD	KD	KLD	Flag	MIE	KIE
Reset	0	1	0	0	0	0	0	0

This 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 64h. 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.

Bit 7 **Reserved** (always reads 0)

Bit 6 **PC Compatibility**
1 = Convert scan codes to PC format. Convert 2-byte break sequences to 1-byte PC-compatible break codes (default)
0 = Disable scan conversion

Bit 5 **Mouse Disable**
1 = Disable mouse interface
0 = Enable mouse interface (default)

Bit 4 **Keyboard Disable**
1 = Disable keyboard interface
0 = Enable keyboard interface (default)

Bit 3 **Keyboard Lock Disable**
1 = Disable keyboard inhibit function
0 = Enable keyboard inhibit function (default)

Bit 2 **System Flag** (This bit can be read back as [Status Register] port 64h bit 2)

Bit 1 **Mouse Interrupt Enable**
1 = Generate interrupt on IRQ12 when mouse output buffer has been written
0 = Disable mouse interrupts (default)

Bit 0 **Keyboard Interrupt Enable**
1 = Generate interrupt on IRQ1 when keyboard output buffer has been written
0 = Keyboard output buffer empty (default)

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 7-2.

Table 7-2. 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	-	-
7	P27 - Keyboard data out	-	-
Bit	Test Port	LoCode	HiCode
0	T0 - Keyboard Clock In	-	-
1	T1 - Mouse Clock In	-	-

Keyboard Controller Input Buffer Port 60h **WO**

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Input Buffer							
Reset	0	0	0	0	0	0	0	0

This register should only be written when port 64h, bit 1 is 0. A value of 1 indicates that the input buffer is full.

Keyboard Controller Output Buffer Port 60h **RO**

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Output Buffer							
Reset	0	0	0	0	0	0	0	0

This register should only be read when port 64h, bit 0 is 1. A value of 0 indicates that the output buffer is empty.

7.2.2 DMA Controller I/O Registers

Master DMA Controller Ports 00h–0Fh

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 7-3.

Table 7-3. Ports 00h–0Fh Master DMA Controller

I/O Address Bits 15–0	Register Name	
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
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 Ports C0h–DFh

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 7-4.

Table 7-4. Ports C0h–DFh Slave DMA Controller

I/O Address Bits 15–0	Register Name	
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 Ports 80h–8Fh

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 7-5.

Table 7-5. Ports 80h–8Fh DMA Page Registers

I/O Address Bits 15–0	Register Name	
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

7.2.3 Interrupt Controller Registers

Master Interrupt Controller Ports 20h–21h

The Master Interrupt Controller controls system interrupt channels 0–7. The two registers are shown in Table 7-6.

Table 7-6. Ports 20h–21h Master Interrupt Controller Registers

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 Ports A0h–A1h

The Slave Interrupt Controller controls system interrupt channels 8–15. The slave system interrupt controller also occupies two register locations, as shown in Table 7-7.

Table 7-7. Ports A0h–A1h Slave Interrupt Controller Registers

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.*		

7.2.4 Interrupt Controller Shadow Registers

The following shadow registers are enabled by setting bit 4 of offset 47h to 1. 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.

Master Interrupt Control Shadow Port 20h RO

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Reserved			OCW3–5	OCW2–7	ICW4–4	ICW4–1	ICW1–3
Reset	0	0	0	xxxxxB				

Bits 7–5 **Reserved** (always reads 0)
Bit 4 **OCW3 bit 5**
Bit 3 **OCW2 bit 7**
Bit 2 **ICW4 bit 4**
Bit 1 **ICW4 bit 1**
Bit 0 **ICW1 bit 3**

Master Interrupt Mask Shadow Port 21h RO

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Reserved			T7–T3 of the Interrupt Vector Address				
Reset	0	0	0	xB	xB	xB	xB	xB

Bits 7–5 **Reserved** (always reads 0)
Bits 4–0 **T7–T3 of the Interrupt Vector Address**

Slave Interrupt Control Shadow Port A0h RO

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Reserved			OCW3–5	OCW2–7	ICW4–4	ICW4–1	ICW1–3
Reset	0	0	0	xB	xB	xB	xB	xB

Bits 7–5 **Reserved** (always reads 0)
Bit 4 **OCW3 bit 5**
Bit 3 **OCW2 bit 7**
Bit 2 **ICW4 bit 4**
Bit 1 **ICW4 bit 1**
Bit 0 **ICW1 bit 3**

Slave Interrupt Mask Shadow Port A1h RO

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Reserved			T7–T3 of the Interrupt Vector Address				
Reset	0	0	0	xxxxxB				

Bits 7–5 **Reserved** (always reads 0)
Bits 4–0 **T7–T3 of the Interrupt Vector Address**

7.2.5 Timer/Counter Registers

Timer/Counter Registers Ports 40h–43h

There are four timer/counter registers, as shown in Table 7-8.

Table 7-8. Ports 40h–43h Timer/Counter Registers

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.

7.2.6 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 7-9.

Table 7-9. 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
04h	Hours am 12 hr:	01h–1Ch	01h–12h
	pm 12 hr:	81h–8Ch	81h–92h
	24 hr:	00h–17h	00h–23h
05h	Hours Alarmam 12 hr:	01h–1Ch	01h–12h
	Hours pm 12 hr:	81h–8Ch	81h–92h
	Hours 24 hr:	00h–17h	00h–23h
06h	Day of the WeekSunday = 1:	01h–07h	01h–07h
07h	Day of the Month	01h–1Fh	01h–31h
08h	Month	01h–0Ch	01h–12h
09h	Year	00h–63h	00h–99h
0Ah	Bit 7	Update in progress	
	Bits 6–4	Divide (010 = enable oscillator and keep time)	
	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)		

Ports 70h–71h are compatible with PC industry standards and can be used to access the lower 128 bytes of the 256-byte on-

chip CMOS RAM. Ports 72h–73h can be used to access the full extended 256-byte space. These ports can be accessed only if Function 0, offset 5Ah, bit 2 is set to select the internal RTC. If this bit is cleared, accesses to port 70h–71h or 72h–73h will be directed to an external RTC.

Ports 74h–75h can be used to 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 cleared to enable access to port 74h–75h.

CMOS Address Port 70h — WO

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	NMID	CMOS Address						
Reset	0	0	0	0	0	0	0	0

Bit 7 **NMI Disable**

1 = Disable NMI Generation (default)

0 = Enable NMI Generation. NMI is asserted on encountering $\overline{\text{IOCHCK}}$ on the ISA bus or $\overline{\text{SERR}}$ on the PCI bus.

Bits 6–0 **CMOS Address (128 bytes)**

CMOS Data Port 71h — RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	CMOS Data							
Reset	0	0	0	0	0	0	0	0

Bits 7–0 **CMOS Data (128 bytes)**

CMOS Data Port 72h, Port 73h, Port 74h, and Port 75h — RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	CMOS Data							
Reset	0	0	0	0	0	0	0	0

Bits 7–0 **CMOS Data (256 bytes)**

7.3 Function 0 Registers (PCI-ISA Bridge)

7.3.1 Function 0 PCI Configuration Space Header

Vendor ID Function 0 Offset 01h–00h **RO**

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Vendor ID															
Reset	0	0	0	1	0	0	0	1	0	0	0	0	0	1	1	0

The Vendor ID is a read-only register containing the value 1106h.

Device ID Function 0 Offset 03h–02h **RO**

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Device ID															
Reset	0	0	0	0	0	1	0	1	1	0	0	0	0	1	1	0

The Device ID is a read-only register containing the value 0586h.

Command Function 0 Offset 05h–04h **RW**

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Reserved												SCE	BM	MS	IOS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Bits 15–4 **Reserved** (always reads 0)

Bit 3 **Special Cycle Enable (Normally RW – see note)**
1 = Enabled (default)
0 = Disabled

Bit 2 **Bus Master** (always reads 0)
1 = Enabled (default)
0 = Disabled

Bit 1 **Memory Space (Normally RO, reads 1 – see note)**
1 = Enabled (default)
0 = Disabled

Bit 0 **I/O Space (Normally Ro, reads 1 – see note)**
1 = Enabled (default)
0 = Disabled

Note: *If the test bit at offset 46h, bit 4 is set, access to the bits indicated above is reversed: bit 3 above becomes read only (reading back 1) and bits 0–1 above become read/write (with a default of 1).*

Status Function 0 Offset 07h–06h RWC

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	DPE	SSE	SMA	RTA	STA	DEVSEL		DPD	FBTB	Reserved						
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bit 15 **Detected Parity Error** (write 1 to clear)
Bit 14 **Signalled System Error** (always reads 0)
Bit 13 **Signalled Master Abort** (always reads 0)
Bit 12 **Received Target Abort** (always reads 0 - write 1 to clear)
Bit 11 **Signalled Target Abort** (always reads 0)
Bit 10 – 9 **DEVSEL Timing** (fixed at 01 = medium)
Bit 8 **Data Parity Detected** (always reads 0)
Bit 7 **Fast Back-to-Back** (always reads 0)
Bits 6–0 **Reserved** (always reads 0)

Revision ID Function 0 Offset 08h RO

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Revision number							
Reset	n	n	n	n	n	n	n	n

The Revision ID is a read-only register containing the revision number.

Program Interface Function 0 Offset 09h RO

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	00h							
Reset	0	0	0	0	0	0	0	0

The Program Interface is a read-only register containing the value 00h.

Sub Class Code Function 0 Offset 0Ah RO

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	01h							
Reset	0	0	0	0	0	0	0	1

The Sub Class Code is a read-only register containing the value 01h.

Class Code Function 0 Offset 0Bh **RO**

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	06h							
Reset	0	0	0	0	0	1	1	0

The Class Code is a read-only register containing the value 06h.

Header Type Function 0 Offset 0Eh **RO**

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	80h							
Reset	1	0	0	0	0	0	0	0

The Header Type is a read-only register containing the value 80h.

BIST Function 0 Offset 0Fh **RO**

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	00h							
Reset	0	0	0	0	0	0	0	0

BIST is a read-only register containing the value 00h.

7.3.2 ISA Bus Control

ISA Bus Control Function 0 Offset 40h **RW**

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	CD	BR	SWS	IOWS	IORT	EALE	RWS	ROMW
Reset	0	0	0	0	0	0	0	0

Bit 7 **ISA Command Delay**
1 = Extra delay
0 = Normal delay (default)

Bit 6 **Extended ISA Bus Ready**
1 = Enable
0 = Disable (default)

Bit 5 **ISA Slave Wait States**
1 = 5 Wait states
0 = 4 Wait states (default)

Bit 4 **Chipset I/O Wait States**
1 = 4 Wait states
0 = 2 Wait states (default)

Bit 3 **I/O Recovery Time**
1 = Enable
0 = Disable (default)

Bit 2 **Extend ALE**
1 = Enable
0 = Disable (default)

Bit 1 **ROM Wait States**
1 = 0 Wait states
0 = 1 Wait state (default)

Bit 0 **ROM Write**
1 = Enable
0 = Disable (default)

SA Test Mode Function 0 Offset 41h RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Reserved		P92FR	Reserved	DDMAC	Reserved		
Reset	0	0	0	0	0	0	0	0

Bits 7–6 **Reserved** (always reads 0)

Bit 5 **Port 92 Fast Reset**
1 = Enable
0 = Disable (default)

Bit 4 **Reserved** (always reads 0)

Bit 3 **Double DMA Clock**
1 = Enable (DMA clock = ISA clock)
0 = Disable (DMA clock = ½) (default)

Bits 2–0 **Reserved** (always reads 0)

ISA Clock Control Function 0 Offset 42h RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	LIO16	Reserved			ICSE	ISACS		
Reset	0	0	0	0	0	0	0	0

Bit 7 **Latch $\overline{\text{IO16}}$**
1 = Disable
0 = Enable (recommended) (default)

Bits 6–4 **Reserved** (always reads 0)

Bit 3 **ISA Bus Clock Select Enable**
1 = ISA clock selected per bits 2–0
0 = ISA Clock = PCLK/4 (default)

Bits 2–0 **ISA Bus Clock Select (when bit 3 = 1)**
000 = PCLK/3 (default)
001 = PCLK/2
010 = PCLK/4
011 = PCLK/6
100 = PCLK/5
101 = PCLK/10
110 = PCLK/12
111 = OSC/2

Note: *To switch the ISA Clock, take the following steps:*

1. *Clear bit 3 of this register.*
2. *Change the value of bits 2–0 to reflect the desired clock.*
3. *Set bit 3.*

ROM Decode Control Function 0 Offset 43h RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0
Reset	0	0	0	0	0	0	0	0

Setting the following bits enables the indicated address range to be included in the ROMCS decode:

Bit 7 **FFFE0000h–FFFEFFFFh**
1 = Enable
0 = Disable (default)

Bit 6 **FFF80000h–FFFDFFFFh**
1 = Enable
0 = Disable (default)

Bit 5 **000E8000h–000EFFFFh**
1 = Enable
0 = Disable (default)

Bit 4 **000E0000h–000E7FFFh**
1 = Enable
0 = Disable (default)

Bit 3 **000D8000h–000DFFFFh**
1 = Enable
0 = Disable (default)

Bit 2 **000D0000h–000D7FFFh**
1 = Enable
0 = Disable (default)

Bit 1 **000C8000h–000CFFFFh**
1 = Enable
0 = Disable (default)

Bit 0 **000C0000h–000C7FFFh**
1 = Enable
0 = Disable (default)

Keyboard Controller Control Function 0 Offset 44h RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Reserved				MLE	Reserved		
Reset	0	0	0	0	0	0	0	0

Bits 7–4 **Reserved** (always reads 0)

Bit 3 **Mouse Lock Enable**
1 = Enable
0 = Disable (default)

Bits 2–0 **Reserved**

Type F DMA Control Function 0 Offset 45h RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	LB	CH7	CH6	CH5	CH3	CH2	CH1	CH0
Reset	0	0	0	0	0	0	0	0

Setting the following bits enables DMA type F timing on the indicated DMA channels.

Bit 7 **ISA Master/DMA to PCI Line Buffer**
1 = Enable
0 = Disable (default)

Bit 6 **DMA Type F Timing on Channel 7**
1 = Enable
0 = Disable (default)

Bit 5 **DMA Type F Timing on Channel 6**
1 = Enable
0 = Disable (default)

Bit 4 **DMA Type F Timing on Channel 5**
1 = Enable
0 = Disable (default)

Bit 3 **DMA Type F Timing on Channel 3**
1 = Enable
0 = Disable (default)

Bit 2 **DMA Type F Timing on Channel 2**
1 = Enable
0 = Disable (default)

Bit 1 **DMA Type F Timing on Channel 1**
1 = Enable
0 = Disable (default)

Bit 0 **DMA Type F Timing on Channel 0**
1 = Enable
0 = Disable (default)

Miscellaneous Control 1 Function 0 Offset 46h — RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Reserved			CC04	Reserved		BRI	PMWE
Reset	0	0	0	0	0		0	0

Bits 7–5 **Reserved** (always reads 0)

Bit 4 **Configure Command Register Offset 05h–04h Access (Test Only)**
1 = Test Mode: Command register bits 0–1 are RW, Bit 3 is RO
0 = Normal Mode: Command register bits 0–1 are RO, Bit 3 is RW

Bits 3–2 **Reserved** (always reads 0)

Bit 1 **PCI Burst Read Interruptability**
1 = Disallow PCI burst read interrupting
0 = Allow burst reads to be interrupted (default)

Bit 0 **Post Memory Write Enable**
1 = Enable
0 = Disable (default)

Miscellaneous Control 2 Function 0 Offset 47h — RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	RS	DTE	PE	ICSRE	Reserved	WDTE	RDTE	PCIRST
Reset	0	0	0	0	0	0	0	0

Bit 7 **CPU Reset Source**
1 = Use INIT as CPU reset
0 = Use CPURST (default)

Bit 6 **PCI DelayTransaction Enable**
1 = Enable
0 = Disable (default)

Bit 5 **EISA 4D0/4D1 Port EnableRW**
1 = Enable (ports 4D0h–4D1h per EISA specification)
0 = Disable (ignore ports 4D0–4D1h) (default)

Bit 4 **Interrupt Controller Shadow Register Enable**
1 = Enable
0 = Disable (default)

Bit 3 **Reserved** (always reads 0)

Bit 2 **Write Delay Transaction Time-Out Timer Enable**
1 = Enable
0 = Disable (default)

Bit 1 **Read Delay Transaction Time-Out Timer Enable**
1 = Enable
0 = Disable (default)

Bit 0 **Software PCI Reset**—Setting this bit causes a PCI reset by asserting the PCIRST pin.

Miscellaneous Control 3 Function 0 Offset 48h RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Reserved				Ex74/75	IUSBCD	IIDECD	PCIMD
Reset	0	0	0	0	0	0	0	1

Bits 7–4 **Reserved** (always reads 0)

Bit 3 **Extra RTC Port 74/75 Enable**
1 = Disable
0 = Enable (default)

Bit 2 **Integrated USB Controller Disable**
1 = Disable
0 = Enable (default)

Bit 1 **Integrated IDE Controller Disable**
1 = Disable
0 = Enable (default)

Bit 0 **512K PCI Memory Decode**
1 = Use the contents of bits 15–12 of offset 4Eh plus 512 Kbytes as the top of PCI memory (default)
0 = Use the contents of bits 15–12 of offset 4Eh as the top of PCI memory

IDE Interrupt Routing Function 0 Offset 4Ah RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	WPGNT	BSIO	Reserved		IDESCH		IDEPCH	
Reset	0	0	0	0	0	1	0	0

Bit 7 **Wait for PGNT before Grant to ISA Master/DMA**
1 = Enable (must be set)
0 = Disable (default)

Bit 6 **Bus Select for I/O Devices below 100h**
1 = Access ports 00h–FFh via XD bus
0 = Access ports 00h–FFh via SD bus

Bits 5–4 **Reserved** (always reads 0)

Bits 3–2 **IDE Second Channel IRQ Routing**
00 = IRQ14
01 = IRQ15 (default)
10 = IRQ10
11 = IRQ11

Bits 1–0 **IDE Primary Channel IRQ Routing**
00 = IRQ14 (default)
01 = IRQ15
10 = IRQ10
11 = IRQ11

ISA DMA/Master Memory Access Control 1 Function 0 Offset 4Ch RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Bit Values Correspond to HA23–HA16 (default = 00h)							
Reset	0	0	0	0	0	0	0	0

The bits in this register correspond to HA23–HA16.

Bits 7–0 **PCI Memory Hole Bottom Address**

ISA DMA/Master Memory Access Control 2 Function 0 Offset 4Dh RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Bit Values Correspond to HA23–HA16 (default = 00h)							
Reset	0	0	0	0	0	0	0	0

The bits in this register correspond to HA23–HA16.

Bits 7–0 **PCI Memory Hole Top Address**

Note: *Access to the memory defined in the PCI memory hole will not be forwarded to PCI. This function is disabled if the top address is less than or equal to the bottom address.*

ISA DMA/Master Memory Access Control 3 Function 0 Offset 4Fh–4Eh

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0

ISA DMA/Master Memory Access Control 3 is a RW register.

Bits 15–12 Top of PCI Memory for ISA DMA/Master Accesses
0000 = 1 Mbyte (default)

0001 = 2 Mbytes

...

1111 = 16 Mbytes

Note: ISA DMA/masters that access addresses higher than the top of PCI will not be directed to the PCI bus.

Bit 11 **Forward E0000h–EFFFFh Accesses to PCI** (default = 0)
Bit 10 **Forward A0000h–BFFFFh Accesses to PCI** (default = 0)
Bit 9 **Forward 80000h–9FFFFh Accesses to PCI** (default = 1)
Bit 8 **Forward 00000h–7FFFFh Accesses to PCI** (default = 1)
Bit 7 **Forward DC000h–DFFFFh Accesses to PCI** (default = 0)
Bit 6 **Forward D8000h–DBFFFh Accesses to PCI** (default = 0)
Bit 5 **Forward D4000h–D7FFFh Accesses to PCI** (default = 0)
Bit 4 **Forward D0000h–D3FFFh Accesses to PCI** (default = 0)
Bit 3 **Forward CC000h–CFFFFh Accesses to PCI** (default = 0)
Bit 2 **Forward C8000h–CBFFFh Accesses to PCI** (default = 0)
Bit 1 **Forward C4000h–C7FFFh Accesses to PCI** (default = 0)
Bit 0 **Forward C0000h–C3FFFh Accesses to PCI** (default = 0)

7.3.3 Plug-N-Play Control Registers

PNP DRQ Routing Function 0 Offset 50h **RW**

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Reserved (default = 04h)							
Reset	0	0	0	0	0	1	0	0

Bits 7–0 **Reserved** (always reads 04h)

PCI IRQ Edge/Level Select Function 0 Offset 54h RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Reserved				PIRQA	PIRQB	PIRQC	PIRQD
Reset	0	0	0	0	0	0	0	0

Bits 7–4 **Reserved** (always reads 0)

Bit 3 **$\overline{\text{PIRQA}}$ Invert (edge)/Non-invert (level)**
1 = Edge
0 = Level (default)

Bit 2 **$\overline{\text{PIRQB}}$ Invert (edge)/Non-invert (level)**
1 = Edge
0 = Level (default)

Bit 1 **$\overline{\text{PIRQC}}$ Invert (edge)/Non-invert (level)**
1 = Edge
0 = Level (default)

Bit 0 **$\overline{\text{PIRQD}}$ Invert (edge)/Non-invert (level)**
1 = Edge
0 = Level (default)

PNP IRQ Routing 1 Function 0 Offset 55h RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	MIRQD Routing				MIRQ0 Routing			
Reset	0	0	0	0	0	0	0	0

Bits 7–4 **MIRQ1 Routing**
0000 = Disabled (default)
0001 = IRQ1
0010 = Reserved
0011 = IRQ3
0100 = IRQ4
0101 = IRQ5
0110 = IRQ6
0111 = IRQ7
1000 = Reserved
1001 = IRQ9
1010 = IRQ10
1011 = IRQ11
1100 = IRQ12
1101 = Reserved
1110 = IRQ14
1111 = IRQ15

Bits 3–0 **MIRQ0 Routing** (same as MIRQ1 routing)

PnP IRQ Routing 2 Function 0 Offset 56h RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	PIRQB Routing				PIRQA Routing			
Reset	0	0	0	0	0	0	0	0

Bits 7–4 **PIRQB Routing** (same as MIRQ1 routing)

Bits 3–0 **PIRQA Routing** (same as MIRQ1 routing)

PNP IRQ Routing 3 Function 0 Offset 57h RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	PIRQD Routing				PIRQC Routing			
Reset	0	0	0	0	0	0	0	0

Bits 7–4 **PIRQD Routing** (same as MIRQ1 routing)

Bits 3–0 **PIRQC Routing** (same as MIRQ1 routing)

PNP IRQ Routing 4 Function 0 Offset 58h RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Reserved				MIRQ2 Routing			
Reset	0	0	0	0	0	0	0	0

Bits 7–4 **Reserved** (always reads 0)

Bits 3–0 **MIRQ2 Routing** (same as MIRQ1 routing)

MIRQ Pin Configuration Function 0 Offset 59h RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Reserved					MIRQ/alternate function		
Reset	0	0	0	0	0	0	0	0

Bits 7–3 **Reserved** (always reads 0)

Bits 2 **MIRQ2/MASTER Selection**
0 = MIRQ2 (default)
1 = $\overline{\text{MASTER}}$

Bits 1 **MIRQ1/KEYLOCK Selection**
0 = MIRQ1 (default)
1 = KEYLOCK

Bits 0 **MIRQ0/APICCS Selection**
0 = MIRQ0 (default)
1 = $\overline{\text{APICCS}}$

XD Power-Up Strap Options Function 0 Offset 5Ah RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	KRP16	KRP15	KRP14	KRP13	Reserved	IRTCE	IPS2ME	IKBCE
Reset	XD7	XD6	XD5	XD4	XD3	XD2	XD1	XD0

The values in the bits of this register are latched from pins XD7–XD0 at power-up, but can be accessed after power-up to change the strapped settings.

Bit 7 **Keyboard RP16** (latched from XD7)

Bit 6 **Keyboard RP15** (latched from XD6)

Bit 5 **Keyboard RP14** (latched from XD5)

Bit 4 **Keyboard RP13** (latched from XD4)

Bit 3 **Reserved** (always reads 0)

Bit 2 **Internal RTC Enable** (latched from XD2)
1 = Enable
0 = Disable

Bit 1 **Internal PS2 Mouse Enable** (latched from XD1)
1 = Enable
0 = Disable

Bit 0 **Internal KBC Enable** (latched from XD0)
1 = Enable
0 = Disable

Note: External strap option values can be set by connecting the indicated external pin to ground or through a 4.7-Kohm pullup to V_{CC} (for 1) or driving it Low with a 7407 TTL open-collector buffer (for 0) as shown in Figure 7-1.

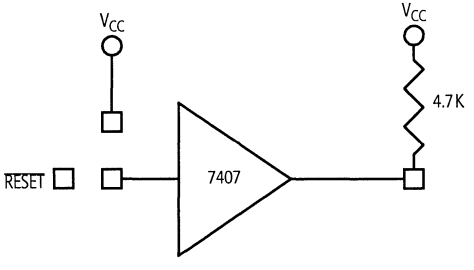


Figure 7-1. Strap Option Circuit

Internal RTC Test Mode Function 0 Offset 5Bh **RW**

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Reserved						RTCSAE	Reserved
Reset	0	0	0	0	0	0	0	0

Bits 7–2 **Reserved** (always reads 0)

Bit 1 **RTC SRAM Access Enable**—This bit is set to access the internal RTC SRAM via ports 74h/75h while the internal RTC is disabled. If the internal RTC is enabled, setting this bit has no effect, and the internal RTC SRAM should be accessed at either ports 70h/71h or ports 72h/73h.
1 = Enable
0 = Disable (default)

Bit 0 **Reserved** (always reads 0)

7.3.4 Distributed DMA Control

Distributed DMA Ch 0 Base/Enable Function 0 Offset 61h–60h **RW**
Distributed DMA Ch 1 Base/Enable Function 0 Offset 63h–62h **RW**
Distributed DMA Ch 2 Base/Enable Function 0 Offset 65h–64h **RW**
Distributed DMA Ch 3 Base/Enable Function 0 Offset 67h–66h **RW**
Distributed DMA Ch 5 Base/Enable Function 0 Offset 6Bh–6Ah **RW**
Distributed DMA Ch 6 Base/Enable Function 0 Offset 6Dh–6Ch **RW**
Distributed DMA Ch 7 Base/Enable Function 0 Offset 6Fh–6Eh **RW**

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Channel n Base Address Bits 15–4												CE	Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits 15–4 **Channel n Base Address bits 15–4**
0000 = default

Bit 3 **Channel n Enable**
1 = Enable
0 = Disable (default)

Bits 2–0 **Reserved** (always reads 0)

7.4 Function 1 Registers (Enhanced IDE Controller)

All EIDE Controller registers are located in Function 1 of the AMD-645 Peripheral Bus Controller PCI configuration space and are accessed through PCI configuration mechanism #1 via address 0CF8h/0CFCh.

The AMD-645 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.

7.4.1 Function 1 PCI Configuration Space Header

Vendor ID Function 1 Offset 01h–00h **RO**

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Vendor ID															
Reset	0	0	0	1	0	0	0	1	0	0	0	0	0	1	1	0

The Vendor ID is a read-only register containing the value 1106h.

Device ID Function 1 Offset 03h–02h **RO**

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Value 0571h															
Reset	0	0	0	0	0	1	0	1	0	1	1	1	0	0	0	1

The Device ID is a read-only register containing the value 0571h.

Command Function 1 Offset 05h–04 **RWh**

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Reserved						FBBC	SE	AS	PER	PS	MWI	SCE	BM	MS	IOS
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bits 15–10 **Reserved** (always reads zero)

Bit 9 **Fast Back-to-Back Cycles** (fixed at 0)
1 = Enabled
0 = Disabled (default)

Bit 8 **$\overline{\text{SERR}}$ Enable** (fixed at 0)
1 = Enabled
0 = Disabled (default)

Bit 7 **Address Stepping** (fixed at 1)
1 = Enabled (default)
0 = Disabled

Bit 6 **Parity Error Response** (fixed at 0)
1 = Enabled
0 = Disabled (default)

Bit 5 **VGA Pallette Snoop** (fixed at 0)
1 = Enabled
0 = Disabled (default)

Bit 4 **Memory Write & Invalidate** (fixed at 0)
1 = Enabled
0 = Disabled (default)

Bit 3 **Special Cycles** (fixed at 0)
1 = Enabled
0 = Disabled (default)

Bit 2 **Bus Master** (SG operation can be issued only when this bit is enabled.)
1 = Enabled
0 = Disabled (default)

Bit 1 **Memory Space** (fixed at 0)
1 = Enabled
0 = Disabled (default)

Bit 0 **I/O Space** (default = 0 = disabled) When this bit is disabled, the device does not respond to any I/O addresses for either compatible or native mode.

Status Function 1 Offset 07h–06h **RWC**

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	DPE	SSE	RMA	RTA	STA	DT		DPD	FBTB	Reserved						
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bit 15 **Detected Parity Error** (default = 0)

Bit 14 **Signalled System Error** (default = 0)

Bit 13 **Signalled Master Abort** (default = 0)

Bit 12 **Received Target Abort** (default = 0)

Bit 11 **Signalled Target Abort** (RO—always reads 0)

Bits 10–9 **DEVSEL Timing**
00 = Fast
01 = Medium (default)
10 = Slow
11 = Reserved

Bit 8 **Data Parity Detected** (default = 0)

Bit 7 **Fast Back-to-Back** (RO—always reads 1)

Bits 6–0 **Reserved** (always reads 0)

Revision ID Function 1 Offset 08 ROh

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Revision Code for IDE Controller Logic Block							
Reset	n	n	n	n	n	n	n	n

The Revision ID is a read-only register containing the revision code for the IDE Controller logic block.

Programming Interface Function 1 Offset 09h RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	MIDEC	Reserved			SPI	SCOM	PPI	PCOM
Reset	1	0	0	0	1	x	1	x

Bit 7 **Master IDE Capability** (fixed at 1 - supported)

Bits 6–4 **Reserved** (always reads 0)

Bit 3 **Secondary Programmable Indicator** (fixed at 1)
1 = Supports both modes (mode is selected by writing bit 2)
0 = Fixed (compatibility or native PCI mode is determined by bit 2)

Bit 2 **Secondary Channel Operating Mode**
1 = Native PCI Mode (default when SPKR=1)
0 = Compatibility Mode (default when SPKR=0)
The default value for this bit is determined at power-up by the strapping at the SPKR pin, pin 134. The strapping determines whether IDE addressing is fixed (1) or flexible (0). (See Figure 7-1 on page 7-27 for a drawing of a strap circuit). After reset, bit 2 can be written to determine the channel operating mode. Table 7-10 summarizes the differences between native PCI and compatibility modes.

Table 7-10. Compatibility Mode vs. Native PCI Mode

Mode		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	
	Secondary	Determined by offset 18h	Determined by offset 1Ch	

Notes:
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).

Bit 1 **Primary Programmable Indicator** (fixed at 1)
1 = Supports both modes (mode is selected by writing bit 0)
0 = Fixed (compatibility or native PCI mode is determined by bit 2)
1 = Native PCI mode (default when SPKR=1)
0 = Compatibility mode (default when SPKR=0)

Bit 0 **Primary Channel Operating Mode**
1 = Native PCI mode (default when SPKR=1)
0 = Compatibility mode (default when SPKR=0)

Sub Class Code Function 1 Offset 0Ah — RO

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	01h							
Reset	0	0	0	0	0	0	0	1

The Sub Class Code is a read-only register containing the value 01h.

Base Class Code Function 1 Offset 0Bh — RO

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	01h							
Reset	0	0	0	0	0	0	0	1

The Base Class Code is a read-only register containing the value 01h.

Latency Timer Function 1 Offset 0Dh — RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	00h							
Reset	0	0	0	0	0	0	0	0

The Latency Timer is a read-write register that defaults to 0.

Header Type Function 1 Offset 0Eh — RO

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	00h							
Reset	1	0	0	0	0	0	0	0

The Header Type is a read-only register containing the value 00h.

BIST Function 1 Offset 0Fh — RO

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	00h							
Reset	0	0	0	0	0	0	0	0

The BIST is a read-only register containing the value 00h.

Primary Data/Command Base Address Function 1 Offset 13h–10h — RW

	Bit 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	Bit 16
	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Port Address													Fixed		
Reset	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	1

The Primary Data/Command Base Address is a read-write register that specifies an 8-byte I/O address space.

Bits 31–16 **Reserved** (always reads 0)

Bits 15–3 **Port address** (default = 01F0h)

Bits 2–0 **Value fixed at 001binary**

Primary Control/Status Base Address Function 1 Offset 17h–14h — RW

	Bit 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	Bit 16
	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Port Address														Fixed	
Reset	0	0	0	0	0	0	1	1	1	1	1	1	0	1	0	1

The Primary Control/Status Base Address is a read-write register that specifies a 4-byte I/O address space, of which only the third byte is active. For example, 3F6h is the active byte for the default base address of 3F4h.

Bits 31–16 **Reserved** (always reads 0)

Bits 15–2 **Port address** (default = 03F4h)

Bits 1–0 **Value fixed at 01binary**

Secondary Data/Command Base Address Function 1 Offset 1Bh–18h RW

	Bit 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	Bit 16
	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Port Address													Fixed		
Reset	0	0	0	0	0	0	0	1	0	1	1	1	0	0	0	1

The Secondary Data/Command Base Address is a read-write register that specifies an 8-byte I/O address space.

Bits 31–16 **Reserved** (always reads 0)

Bits 15–3 **Port address** (default = 0170h)

Bits 2–0 **Value fixed at 001binary**

Secondary Control/Status Base Address Function 1 Offset 1Fh–1Ch RW

	Bit 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	Bit 16
	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Port Address														Fixed	
Reset	0	0	0	0	0	0	1	1	0	1	1	1	0	1	0	1

The Secondary Control/Status Base Address is a read-write register that specifies a 4-byte I/O address space, of which only the third byte is active. For example, 376h is the active byte for the default base address of 374h.

Bits 31–16 **Reserved** (always reads 0)

Bits 15–2 **Port address** (default = 374h)

Bits 1–0 **Value fixed at 01 binary**

Bus Master Control Registers Base Address RW Function 1 Offset 23h–20h

	Bit 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	Bit 16
	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Port Address												Fixed			
Reset	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	1

The Bus Master Control Registers Base Address is a read-write register that specifies a 16-byte I/O address space which is compliant with the SFF 8038i rev. 1.0 specification.

Bits 31–16 **Reserved** (always reads 0)

Bits 15–4 **Port address** (default = CC0h)

Bits 3–0 **Value fixed at 0001 binary**

Interrupt Line Function 1Offset 3Ch — RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	0Eh							
Reset	0	0	0	0	1	1	1	0

The Interrupt Line is a read-write register containing the default value 0Eh.

Interrupt Pin Function 1 Offset 3Dh — RO

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	IRC7	IRC6	IRC5	IRC4	IRC3	IRC2	IRC1	IRC0
Reset	0	0	0	0	0	0	0	0

The Interrupt Pin is a read-only register that defines the interrupt routing mode.

Bits 7–0 **Interrupt Routing Mode**
00h =Legacy mode interrupt routing (default)
01h =Native mode interrupt routing

Min Gnt Function 1 Offset 3Eh — RO

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	00h							
Reset	0	0	0	0	0	0	0	0

Min Gnt is a read-only register containing the value 00h.

Max Latency Function 1 Offset 3Fh — RO

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	00h							
Reset	0	0	0	0	0	0	0	0

Max Latency is a read-only register containing the value 00h.

7.4.2 IDE Controller-Specific Configuration Registers

Chip Enable Function 1 Offset 40h RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Reserved						PCE	SCE
Reset	0	0	0	0	0	1	0	0

Chip Enable is a read-write control register used to enable the primary or secondary channel.

Bits 7–2 **Reserved** (always reads 000001 binary)

Bit 1 **Primary Channel Enable**
1 = Enabled
0 = Disabled (default)

Bit 0 **Secondary Channel Enable**
1 = Enabled
0 = Disabled (default)

IDE Configuration Function 1 Offset 41h RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	PRPB	PPWB	SRPB	SPWB	Reserved			
Reset	0	0	0	0	0	1	1	0

IDE Configuration is a read-write control register that defaults to 06h.

Bit 7 **Primary IDE Read Prefetch Buffer**
1 = Enabled
0 = Disabled (default)

Bit 6 **Primary IDE Post Write Buffer**
1 = Enabled
0 = Disabled (default)

Bit 5 **Secondary IDE Read Prefetch Buffer**
1 = Enabled
0= Disabled (default)

Bit 4 **Secondary IDE Post Write Buffer**
1 = Enabled
0 = Disabled (default)

Bits 3–0 **Reserved** (always reads 0110 binary). Although they are read-write, the value of these bits should never be changed.)

Reserved (Do Not Program) Function 1 Offset 42h RW

The reserved register at Function 1, offset 42h is a read-write register that should not be programmed.

FIFO Configuration Function 1 Offset 43h RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Reserved.	FIFO Configuration		Reserved	TPC		TSC	
Reset	0	0	1	1	1	0	1	0

First-In-First-Out (FIF0) Configuration is a read-write control register.

Bit 7 **Reserved** (always reads 0)

Bits 6–5 **FIFO Configuration**—These bits determine FIFO distribution as shown in Table 7-11.

Table 7-11.FIFO Distribution

Bits 6, 5	Primary Channel	Secondary Channel
00	16	0
01(default)	8	8
10	8	8
11	0	16

Bit 4 **Reserved** (always reads 1)

Bits 3–2 **Threshhold for Primary Channel**
00 = 1
01 = 3/4
10 = 1/2 (default)
11 = 1/4

Bits 1–0 **Threshhold for Secondary Channel**
00 = 1
01 = 3/4
10 = 1/2 (default)
11 = 1/4

Miscellaneous Control 1 Function 1 Offset 44h RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Rsvd.	MRCWS	MWCWS	CA	SRRR	Reserved		
Reset	0	1	1	0	1	0	0	0

Bit 7 **Reserved** (always reads 0)

Bit 6 **Master Read Cycle $\overline{\text{IRDY}}$ Wait States**
1 = 1 wait state (default)
0 = 0 wait states

Bit 5 **Master Write Cycle IRDY Wait States**
1 = 1 wait state (default)
0 = 0 wait states

Bit 4 **FIFO Output Data 1/2 Clock Advance**
1 = Enabled
0 = Disabled (default)

Bit 3 **Bus Master IDE Status Register Read Retry**
1 = Enabled (default)
0 = Disabled

Bits 2–0 **Reserved** (always reads 0)

Miscellaneous Control 2 Function 1 Offset 45h RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Rsvd.	ISS	Reserved					
Reset	0	0	0	0	0	0	0	0

Bit 7 **Reserved** (always reads 0)

Bit 6 **Interrupt Steering Swap**
1 = Swap interrupts between the two channels (default)
0 = Do not swap channel interrupts

Bits 5–0 **Reserved** (always reads 0)

Miscellaneous Control 3 Function 1 Offset 46h RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	PCR	SCR	PCE	SCE	Reserved		MPW	
Reset	1	1	0	0	0	0	0	0

Bit 7 **Primary Channel Read DMA FIFO Flush**
1 = Enable FIFO flush for read DMA when interrupt asserts primary channel (default)
0 = Disable

Bit 6 **Secondary Channel Read DMA FIFO Flush**
1 = Enable FIFO flush for read DMA when interrupt asserts secondary channel (default)
0 = Disable

Bit 5 **Primary Channel End-of-Sector FIFO Flush**
1 = Enable FIFO flush at the end of each sector for the primary channel
0 = Disable (default)

Bit 4 **Secondary Channel End-of-Sector FIFO Flush**
1 = Enable FIFO flush at the end of each sector for the secondary channel
0 = Disable (default)

Bits 3–2 **Reserved** (always reads 0)

Bits 1–0 **Max DRDY Pulse Width**
Maximum DRDY pulse width after the cycle count. Command will deassert in spite of DRDY status to avoid hanging the system.

00 = No limitation (default)
01 = 64 PCI clocks
10 = 128 PCI clocks
11 = 192 PCI clocks

Drive Timing Control Function 1 Offset 4Bh–48 RWh

	Bit 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	Bit 16
	PD0APW				PD0RT				PD1APW				PD1RT			
Reset	1	0	1	0	1	0	0	0	1	0	1	0	1	0	0	0

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	SD0APW				SD0RT				SD1APW				SD1RT			
Reset	1	0	1	0	1	0	0	0	1	0	1	0	1	0	0	0

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.

Bits 31–28 **Primary Drive 0 Active Pulse Width** (default = 1010 binary)
Bits 27–24 **Primary Drive 0 Recovery Time** (default = 1000 binary)
Bits 23–20 **Primary Drive 1 Active Pulse Width** (default = 1010 binary)
Bits 19–16 **Primary Drive 1 Recovery Time** (default = 1000 binary)
Bits 15–12 **Secondary Drive 0 Active Pulse Width** (default = 1010 binary)
Bits 11–8 **Secondary Drive 0 Recovery Time** (default = 1000 binary)
Bits 7–4 **Secondary Drive 1 Active Pulse Width** (default = 1010 binary)
Bits 3–0 **Secondary Drive 1 Recovery Time** (default = 1000 binary)

Address Setup Time Function 1 Offset 4Ch RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	PD0AST		PD1AST		SD0AST		SD1AST	
Reset	0	1	1	0	1	0	0	0

Bits 7–6 **Primary Drive 0 Address Setup Time**

Bits 5–4 **Primary Drive 1 Address Setup Time**

Bits 3–2 **Secondary Drive 0 Address Setup Time**

Bits 1–0 **Secondary Drive 1 Address Setup Time**—Each of these bit pairs defines the corresponding address setup time as follows:
00 = 1T
01 = 2T
10 = 3T
11 = 4T (default)

Secondary Non-1F0 Port Access Timing Function 1 Offset 4Eh RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Active Pulse Width				Recovery Time			
Reset	1	1	1	1	1	1	1	1

The actual value in the field is the encoded value in the field plus one. This value indicates the number of PCI clocks.

Bits 7–4 **DIOR/DIOW Active Pulse Width** (default = 1111 binary)

Bits 4–0 **DIOR/DIOW Recovery Time** (default = 1111 binary)

Primary Non-1F0 Port Access Timing Function 1 Offset 4Fh RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Active Pulse Width				Recovery Time			
Reset	1	1	1	1	1	1	1	1

The actual value is the encoded value in the field plus one. This value indicates the number of PCI clocks.

Bits 7–4 **DIOR/DIOW Active Pulse Width** (default = 1111 binary)

Bits 4–0 **DIOR/DIOW Recovery Time** (default = 1111 binary)

Ultra DMA-33 Extended Timing Control Function 1 Offset 53h–50h RW

	Bit 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	Bit 16
	Primary Drive 0								Primary Drive 1							
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1
	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Secondary Drive 0								Secondary Drive 1							
Reset	1	1	0	0	0	0	1	1	0	0	0	0	0	0	1	1

Each byte of these registers defines Ultra DMA-33 operation for the indicated drive. The bit definitions are consistent within each byte.

Bit 31 **Primary Drive 0 Ultra DMA-33 Mode Enable Method**
1 = Enable by setting bit 6 of this register
0 = Enable by using the *Set Feature* command (default)

Bit 30 **Primary Drive 0 Ultra DMA-33 Mode Enable**
1 = Enable Ultra DMA-33 mode operation
0 = Disable (default)

Bit 29 **Primary Drive 0 Ultra DMA-33 Transfer Mode**
1 = Transfer based on Ultra DMA-33 PIO mode
0 = Based on Ultra DMA-33 DMA mode (default)

Bits 28–26 **Reserved** (always reads 0)

Bits 25–24 **Primary Drive 0 Cycle Time**
00 = 2T
01 = 3T
10 = 4T
11 = 5T (default)

Bit 23 **Primary Drive 1 Ultra DMA-33 Mode Enable Method**
1 = Enable by setting bit 6 of this register
0 = Enable by using the *Set Feature* command (default)

Bit 22 **Primary Drive 1Ultra DMA-33 Mode Enable**
1 = Enable Ultra DMA-33 mode operation
0 = Disable (default)

Bit 21 **Primary Drive 1 Ultra DMA-33 Transfer Mode**
1 = Transfer based on Ultra DMA-33 PIO mode
0 = Based on Ultra DMA-33 DMA mode (default)

Bits 20–18 **Reserved** (always reads 0)

Bits 17–16 **Primary Drive 1 Cycle Time**
00 = 2T
01 = 3T
10 = 4T
11 = 5T (default)

Bit 15 **Secondary Drive 0 Ultra DMA-33 Mode Enable Method**
1 = Enable by setting bit 6 of this register
0 = Enable by using the *Set Feature* command (default)

Bit 14 **Secondary Drive 0 Ultra DMA-33 Mode Enable**
1 = Enable Ultra DMA-33 mode operation
0 = Disable (default)

Bit 13 **Secondary Drive 0 Ultra DMA-33 Transfer Mode**
1 = Transfer based on Ultra DMA-33 PIO mode
0 = Based on Ultra DMA-33 DMA mode (default)

Bits 12–10 **Reserved** (always reads 0)

Bits 9–8 **Secondary Drive 0 Cycle Time**
00 = 2T
01 = 3T
10 = 4T
11 = 5T (default)

Bit 7 **Secondary Drive 1 Ultra DMA-33 Mode Enable Method**
1 = Enable by setting bit 6 of this register
0 = Enable by using the *Set Feature* command (default)

Bit 6 **Secondary Drive 1Ultra DMA-33 Mode Enable**
1 = Enable Ultra DMA-33 mode operation
0 = Disable (default)

Bit 5 **Secondary Drive 1 Ultra DMA-33 Transfer Mode**
1 = Transfer based on Ultra DMA-33 PIO mode
0 = Based on Ultra DMA-33 DMA mode (default)

Bits 4–2 **Reserved** (always reads 0)

Bits 1–0 **Secondary Drive 1 Cycle Time**
00 = 2T
01 = 3T
10 = 4T
11 = 5T (default)

Primary Sector Size Function 1 Offset 61h–60h RW

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Reserved				Number of Bytes per Sector											
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

The Primary Sector Size is a read-write control register whose bits 11–0 determine the size of each primary sector. The value of these bits defaults to 200h.

Secondary Sector Size Function 1 Offset 69h–68h **RW**

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Reserved				Number of Bytes per Sector											
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

The Secondary Sector Size is a read-write control register whose bits 11–0 determine the size of each secondary sector. The value of these bits defaults to 200h.

7.4.3 IDE I/O Registers

The IDE I/O registers comply with the SFF 8038I v. 1.0 standard. The base address of these registers is determined by configuration register Function 1, offset 09h (see page 7-31). The command block primary channel is 1F0h–1F7h, while the secondary channel is 170H–177h. Refer to the specification for further details.

Primary Channel Command Function 1 Offset 00h

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Primary Channel Command							
Reset								

Primary Channel Command is an I/O register.

Primary Channel Status Function 1 Offset 02h

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Primary Channel Status							
Reset	0	0	0	0	0	0	0	0

Primary Channel Status is an I/O register.

Primary Channel PRD Table Address Function 1 Offset 07h–04h

	Bit 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Primary Channel PRD Table Address															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Primary Channel PRD Table Address															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The Primary Channel PRD Table Address is an I/O register.

Secondary Channel Command Function 1 Offset 08h

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Secondary Channel Command							
Reset								

Secondary Channel Command is an I/O register.

Secondary Channel Status Function 1 Offset 0Ah

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Secondary Channel Status							
Reset	0	0	0	0	0	0	0	0

Secondary Channel Status is an I/O register.

Secondary Channel PRD Table Address Function 1 Offset 0Fh–0Ch

	Bit 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Secondary Channel PRD Table Address															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Secondary Channel PRD Table Address															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The Secondary Channel PRD Table Address is an I/O register.

7.5 Function 2 Registers (USB Controller)

All universal serial bus (USB) controller registers are located in Function 2 of the AMD-645 Peripheral Bus Controller PCI configuration space and are accessed through PCI configuration mechanism #1 via address 0CF8h/0CFCh.

This USB host controller interface is fully compatible with *UHCI specification v. 1.1*. There are two sets of software accessible-registers, PCI configuration registers and USB I/O registers.

7.5.1 Function 2 PCI Configuration Space Header

Vendor ID Function 2 Offset 01h–00h **RO**

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Vendor ID															
Reset	0	0	0	1	0	0	0	1	0	0	0	0	0	1	1	0

The Vendor ID is a read-only register containing the value 1106h.

Device ID Function 2 Offset 03h–02h **RO**

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Value 3038h															
Reset	0	0	1	1	0	0	0	0	0	0	1	1	1	0	0	0

The Device ID is a read-only register containing the value 3038h.

Command Function 2 Offset 05h–04h **RW**

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Reserved								AS	Reserved		MWI	Rsvd.	BM	MS	IOS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits 15–8 **Reserved** (always reads zero)

Bit 7 **Address Stepping**
1 = Enabled
0 = Disabled (default)

Bit 6 **Reserved** (Parity error response - fixed at 0)

Bit 5 **Reserved** (VGA Pallette Snoop - fixed at 0)

Bit 4 **Memory Write & Invalidate**
1 = Enabled
0 = Disabled (default)

Bit 3 **Reserved** (Special cycle monitoring - fixed at 0)

Bit 2 **Bus Master**
1 = Enabled
0 = Disabled (default)

Bit 1 **Memory Space**
1 = Enabled
0 = Disabled (default)

Bit 0 **I/O Space**
1 = Enabled
0 = Disabled (default)

Status Function 2 Offset 07h–06h RWC

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Rsvd.	SSE	RMA	RTA	STA	DEVSEL		Reserved								
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bit 15 **Reserved** (Detected parity error - always reads 0)

Bit 14 **Signalled System Error** (default = 0)

Bit 13 **Received Master Abort** (default = 0)

Bit 12 **Received Target Abort** (default = 0)

Bit 11 **Signalled Target Abort** (default = 0)

Bit 10–9 **DEVSEL Timing**
00 = Fast
01 = Medium (default)
10 = Slow
11 = Reserved

Bits 8–0 **Reserved** (always reads 0)

Revision ID Function 2 Offset 08h RO

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Silicon Revision Code							
Reset								

Revision ID is a read-only register containing the silicon revision code, where the value 00h indicates first silicon.

Programming Interface Function 2 Offset 09h RO

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	00h							
Reset	0	0	0	0	0	0	0	0

Programming Interface is a read-only register containing the value 00h.

Sub Class Code Function 2 Offset 0Ah RO

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	03h							
Reset	0	0	0	0	0	0	1	1

Sub Class Code is a read-only register containing the value 03h.

Base Class Code Function 2 Offset 0Bh RO

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	0Ch							
Reset	0	0	0	0	1	1	0	0

Class Code is a read-only register containing the value 0Ch.

Latency Timer Function 2 Offset 0Dh RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	16h							
Reset	0	0	0	1	0	1	1	0

Latency Timer is a read-write register containing the default 16h.

Header Type Function 2 Offset 0Eh RO

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	00h							
Reset	0	0	0	0	0	0	0	0

Header Type is a read-only register containing the value 00h.

USB I/O Register Base Address Function 2 Offset 23h–20h RW

	Bit 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	Bit 16
	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	USB I/O Register Base Address											Fixed				
Reset	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	1

Bits 15–5 of this register are used to set the port address for the base of the USB I/O register block, corresponding to AD15–AD5.

Bits 31–16 **Reserved** (always reads 0)

Bits 15–5 **USB I/O Register Base Address**

Bits 4–0 **Fixed** (value of these bits is set at 00001 binary)

Interrupt Line Function 2 Offset 3Ch **RW**

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	00h							
Reset	0	0	0	1	0	1	1	0

Interrupt Line is a read-write register containing the default 00h.

Interrupt Pin Function 2 Offset 3Dh **RO**

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	04h							
Reset	0	0	0	0	0	1	0	0

Interrupt Pin is a read-only register containing the value 04h.

7.5.2 USB-Specific Configuration Registers

Miscellaneous Control 1 Function 2 Offset 40h **RW**

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	MCO	BO	PCO	Rsvd.	DLO	PM	DMAO	WS
Reset	0	0	0	0	0	0	0	0

Bit 7 **PCI Memory Command Option**
1 = Support memory read and memory write commands only
0 = Support memory-read-line, memory-read-multiple, and memory-write-and-invalidate commands (default)

Bit 6 **Babble Option**
1 = Do not disable babbled port
0 = Disable babbled port when EOF babble occurs (default)

Bit 5 **PCI Parity Check Option**
1 = Enable parity check and $\overline{\text{PERR}}$ generation
0 = Disable parity check and $\overline{\text{PERR}}$ generation (default)

Bit 4 **Reserved** (always reads 0)

Bit 3 **USB Data Length Option**
1 = Support TD length up to 1023
0 = Support TD length up to 1280 (default)

Bit 2 **USB Power Management**
1 = Enable USB power management
0 = Disable USB power management (default)

Bit 1 **DMA Option**
1 = 8-DW burst access
0 = 16-DW burst access (default)

Bit 0 **PCI Wait States**
1 = 1 Wait State
0 = 0 Wait States (default)

Miscellaneous Control 2 Function 2 Offset 41h — RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Reserved					TO	A20PTO	Reserved
Reset	0	0	0	0	0	0	0	0

Bits 7–3 **Reserved** (always reads 0)

Bit 2 **Trap Option**
1 = Set trap 60/64 bits only when trap 60/64 enable bits are set
0 = Set trap 60/64 status bits without checking the enable bits (default)

Bit 1 **A20GATE Pass Through Option**
1 = Do not pass through I/O port 64h
0 = Pass through the A20GATE command sequence defined in UHCI (default)

Bit 0 **Reserved** (always reads 0)

Serial Bus Release Number Function 2 Offset 60h — RO

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Release Number							
Reset	0	0	0	1	0	0	0	0

Serial Bus Release Number is a read-only register that defaults to a value of 10h.

Legacy Support Function 2 Offset C1h–C0h — RO

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Fixed															
Reset	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Legacy Support is a read-only register. To achieve UHCI v. 1.1 compliance, the value of this register is fixed at 2000h.

7.5.3 USB I/O Registers

These registers are compliant with the UHCI v. 1.1 standard. The USB I/O Register Base Address register at Function 0, offset 23h–20h is used to program the base address to which each of the USB I/O registers is offset. Refer to the specification for further details.

USB Command Function 2 Offset 01h–00h

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	USB Command															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USB Command is an I/O register.

USB Status Function 2 Offset 03h–02h

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	USB Status															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USB Status is an I/O register.

USB Interrupt Enable Function 2 Offset 05h–04h

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	USB Interrupt Enable															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USB Interrupt Enable is an I/O register.

Frame Number Function 2 Offset 07h–06h

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Frame Number															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Frame Number is an I/O register.

Frame List Base Address Function 2 Offset 0Bh–08h

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	USB Command															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Frame List Base Address is an I/O register.

Start Of Frame Modify Function 2 Offset 0Ch

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Start Of Frame Modify							
Reset	0	0	0	0	0	0	0	0

Start Of Frame Modify is an I/O register.

Port 1 Status/Control Function 2 Offset 11h–10h

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Port 1 Status/Control															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Port 1 Status/Control is an I/O register.

Port 2 Status/ControlFunction 2 Offset 13h–12h

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Port 2 Status/Control															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Port 2 Status/Control is an I/O register.

7.6 Function 3 Registers (Power Management)

This section describes the ACPI (Advanced Configuration and Power Interface) power management system of the AMD-645 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.

7.6.1 Function 3 PCI Configuration Space Header

Vendor ID Function 3 Offset 01h–00h RO

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Vendor ID															
Reset	0	0	0	1	0	0	0	1	0	0	0	0	0	1	1	0

The Vendor ID is a read-only register containing the value 1106h.

Device ID Function 3 Offset 03h–02h **RO**

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Value 3040h															
Reset	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0

The Device ID is a read-only register containing the value 3040h.

Command Function 3 Offset 05h–04h **RW**

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Reserved															IOS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits 15–8 **Reserved** (always reads zero)

Bit 7 **Reserved** (Address Stepping - fixed at 0)

Bit 6 **Reserved** (Parity error response - fixed at 0)

Bit 5 **Reserved** (VGA Pallette Snoop - fixed at 0)

Bit 4 **Reserved** (Memory Write & Invalidate - fixed at 0)

Bit 3 **Reserved** (Special cycle monitoring - fixed at 0)

Bit 2 **Reserved** (Bus Master - fixed at 0)

Bit 1 **Reserved** (Memory Space - fixed at 0)

Bit 0 **I/O Space**
Set this bit to allow access to the Power Management I/O register block (see offset 23h–20h on page 7-34 to set the base address for this register block).

1 = Enabled
0 = Disabled (default)

Status Function 3 Offset 07h–06h **RWC**

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Reserved					DEVSEL		Reserved								
Reset	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0

Bit 15 **Reserved** (Detected Parity Error - always reads 0)

Bit 14 **Reserved** (Signalled System Error - always reads 0)

Bit 13 **Reserved** (Received Master Abort - always reads 0)

Bit 12 **Reserved** (Received Target Abort - always reads 0)

Bit 11 **Reserved** (Signalled Target Abort - always reads 0)

Bit 10–9 **DEVSEL Timing**
00 = Fast
01 = Medium (default)
10 = Slow
11 = Reserved

Bit 8 **Reserved** (Data Parity Detected - always reads 0)

Bit 7 **Reserved** (Fast Back-to-Back - always reads 1)

Bits 6–0 **Reserved** (always reads 0)

Revision ID Function 3 Offset 08h — RO

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Silicon Revision Code							
Reset								

Revision ID is a read-only register containing the silicon revision code, where the value 00h indicates first silicon. The register defaults to the value of current silicon.

Programming Interface Function 3 Offset 09h — RO

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	00h							
Reset	0	0	0	0	0	0	0	0

The value returned by this register can be changed by writing the desired value to PCI Configuration Function 3, offset 61h.

Sub Class Code Function 3 Offset 0Ah — RO

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	03h							
Reset	0	0	0	0	0	0	1	1

The value returned by this register can be changed by writing the desired value to PCI Configuration Function 3, offset 62h.

Base Class Code Function 3 Offset 0Bh — RO

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	0Ch							
Reset	0	0	0	0	1	1	0	0

The value returned by this register can be changed by writing the desired value to PCI Configuration Function 3, offset 63h.

Latency Timer Function 3 Offset 0Dh **RW**

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	16h							
Reset	0	0	0	1	0	1	1	0

Latency Timer is a read-write register containing the default 16h.

Header Type Function 3 Offset 0Eh **RO**

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	00h							
Reset	0	0	0	0	0	0	0	0

Header Type is a read-only register containing the value 00h.

Power Management I/O Register Base Address Function 3 Offset 23h–20h **RW**

	Bit 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	Bit 16
	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Power Management I/O Register Base Address								Fixed							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bits 31–16 **Reserved** (always reads 0)

Bits 15–8 **USB I/O Register Base Address**—These two bits determine the port address for the base of the 256-byte Power Management I/O Register block, corresponding to AD15–AD8. The "I/O Space" bit at offset 5h–4h bit 0 enables access to this register block.

Bits 7–0 **Fixed** (value of these bits is set at 00000001 binary)

7.6.2 Power Management-Specific Configuration Registers

Pin Configuration Function 3 Offset 40h **RW**

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	GPIO4	GPIO3	Reserved					
Reset	1	1	0	0	0	0	0	0

Bit 7 **GPIO4 Configuration**
0 = Define pin 136 as GPO_WE
1 = Define pin 136 as GPIO4 (default)

Bit 6 **GPIO3 Configuration**
0 = Define pin 92 as $\overline{\text{GPI\_RE}}$
1 = Define pin 92 as GPIO3 (default)

Bits 5–0 **Reserved** (always reads 0)

General Configuration Function 3 Offset 41h RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	PID	ATR	Reserved		ATCS	PFA	CTCS	Reserved
Reset	0	0	0	0	0	0	0	0

Bit 7 **PWRBTN Input Debounce**
0 = Disable (default)
1 = Enable

Bit 6 **ACPI Timer Reset**
0 = Disable (default)
1 = Enable

Bits 5–4 **Reserved** (do not program)

Bit 3 **ACPI Timer Count Select**
0 = 24-bit timer (default)
1 = 32-bit timer

Bit 2 **PCI Frame Activation in C2 as Resume Event**
0 = Disable (default)
1 = Enable

Bit 1 **Clock Throttling Clock Selection**
0 = 32 µsec (512 µsec cycle time) (default)
1 = 1 msec (16 msec cycle time)

Bit 0 **Reserved** (do not program)

SCI Interrupt Configuration Function 3 Offset 42h RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Reserved				SCI Interrupt Assignment			
Reset	0	0	0	0	0	0	0	0

Bits 7–4 **Reserved** (always reads 0)

Bits 3–10 **SCI interrupt Assignment**

0000 = disabled (default)	1000 = IRQ8
0001 = IRQ1	1001 = IRQ9
0010 = Reserved	1010 = IRQ10
0011 = IRQ3	1011 = IRQ11
0100 = IRQ4	1100 = IRQ12
0101 = IRQ5	1101 = IRQ13
0110 = IRQ6	1110 = IRQ14
0111 = IRQ7	1111 = IRQ15

Primary Interrupt Channel Function 3 Offset 45h–44h RW

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	15P	14P	13P	12P	11P	10P	9P	8P	7P	6P	5P	4P	3P	Rsvd	1P	0P
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Setting any bit except bit 2 enables the corresponding IRQ as the primary interrupt channel.

Secondary Interrupt Channel Function 3 Offset 47h–46h RW

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	15S	14S	13	12S	11S	10S	9S	8S	7S	6S	5S	4S	3S	Rsvd	1S	0S
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Setting any bit except bit 2 enables the corresponding IRQ as the secondary interrupt channel.

GP Timer Control Function 3 Offset 53h–50h RW

	Bit 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	Bit 16
	CMTCV		CMS	CME	SETCV		SEOS	SETE	GP1 Timer Count Value							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	GP0 Timer Count Value								1TS	1AR	GP1TB		0TS	0AR	GP0TB	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bits 31–30 Conserve Mode Timer Count Value
00 = 1/16 sec(default)
01 = 1/8 sec
10 = 1 sec
11 = 1 minute

Bit 29 Conserve Mode Status
This bit reads 1 when the system is in conserve mode.

Bit 28 Conserve Mode Enable—Set this bit to enable conserve mode (not used in desktop applications).

Bits 27–26 Secondary Event Timer Count Value
00 = 2 msec (default)
01 = 64 msec
10 = 1/2 sec
11 = by EOI + 0.25 msec

Bit 25 Secondary Event Occurred Status—This bit is set when a secondary event has occurred (to resume the system from suspend) and that the secondary event timer is counting down.

Bit 24 **Secondary Event Timer Enable**
0 = Disable (default)
1 = Enable

Bits 23–16 **GP1 Timer Count Value** (base defined by bits 5–4)

Bits 15–8 **GP0 Timer Count Value** (base defined by bits 1–0)

Bit 7 **GP1 Timer Start**—When this bit is set, the GP1 timer loads the value defined by bits 23–16 of this register and starts counting down. The GP1 timer is reloaded at the occurrence of certain peripheral events enabled in the GP Timer Reload Enable register in Function 3 I/O Space, offset 38h (see page 7-69). If no such event occurs and the GP1 timer counts down to zero, then the GP1 Timer Timeout Status bit is set. This bit is located at Function 3 I/O Space, offset 28h, bit 3 (see page 7-65). In addition, an SMI is generated if the GP1 Timer Timeout Enable bit is set. This bit is located at Function 3 I/O Space, offset 2Ah, bit 3(see page 7-66).

Bit 6 **GP1 Timer Automatic Reload**—Setting this bit enables the GP1 timer to reload automatically after counting down to 0.

Bits 5–4 **GP1 Timer Base**
00 = disable (default)
01=32 µsec
10=1 second
11=1 minute

Bit 3 **GP0 Timer Start**—When this bit is set, the GP0 timer loads the value defined by bits 15–8 of this register and starts counting down. The GP0 timer is reloaded at the occurrence of certain peripheral events enabled in the GP Timer Reload Enable register in Function 3 I/O Space, offset 38h (see page 7-69). If no such event occurs and the GP0 timer counts down to zero, then the GP0 Timer Timeout Status bit is set. This bit is located at Function 3 I/O Space, offset 28h, bit 2 (see page 7-65). In addition, an SMI is generated if the GP0 Timer Timeout Enable bit is set. This bit is located at Function 3 I/O Space, offset 2Ah, bit 2 (see page 7-66).

Bit 2 **GP0 Timer Automatic Reload**—Setting this bit enables the GP0 timer to reload automatically after counting down to 0.

Bits 1–0 **GP0 Timer Base**
00 = disable (default)
01=1/16 second
10=1 second
11=1 minute

Programming Interface Read Value Function 3 Offset 61h WO

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Offset 09h Read Value							
Reset	0	0	0	0	0	0	0	0

Bits 7–0 **Offset 09h Read Value**—The value returned by the register at offset 09h (Programming Interface) can be changed by writing the desired value to this location.

Sub Class Read Value Function 3 Offset 62h **WO**

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Offset 0Ah Read Value							
Reset	0	0	0	0	0	0	0	0

Bits 7–0 **Offset 0Ah Read Value**—The value returned by the register at offset 0Ah (Sub Class Code) can be changed by writing the desired value to this location.

Base Class Read Value Function 3 Offset 63h **WO**

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Offset 0Bh Read Value							
Reset	0	0	0	0	0	0	0	0

Bits 7–0 **Offset 0Bh Read Value**—The value returned by the register at offset 0Bh (Base Class Code) can be changed by writing the desired value to this location.

7.6.3 Power Management I/O Space Registers

Basic Power Management Control Status

Power Management Status Offset 01h–00h **RWC**

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	WS	Reserved			PBOS	RTCS	Rsvd	PBS	Reserved		GS	BMS	Reserved			TCS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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.

Bit 15 **Wakeup Status (WAK_STS)** (default = 0)—This bit is set when the system is in the suspend state and an enabled resume event occurs. Upon setting this bit, the system automatically transitions from the suspend state to the normal working state (from C3 to C0 for the processor).

Bits 14–12 **Reserved** (always reads 0)

Bit 11 **Power Button Override Status (PBOR_STS)** (default = 0)—This bit is set when the PWRBTN input pin is continuously asserted for more than 4 seconds. The setting of this bit will reset the PB_STS bit and transition the system into the soft off state.

Bit 10 **RTC Status (RTC_STS)** (default = 0)—This bit is set when the RTC generates an alarm in response to assertion of the RTC IRQ signal.

Bit 9 **Reserved** (always reads 0)

Bit 8 **Power Button Status (PB_STS)** (default = 0)—This bit is set when the PWRBTN signal is asserted Low. If the PWRBTN signal is held Low for more than four seconds, this bit is cleared, the PBOR_STS bit is set, and the system transitions into the soft off state.

Bit 7–6 **Reserved** (always reads 0)

Bit 5 **Global Status (GBL_STS)** (default = 0)—This bit is set by hardware when BIOS_RLS is set (typically by an SMI routine to release control of the SCI/SMI lock). When this bit is cleared by software (by writing a one to this bit position) the BIOS_RLS bit is simultaneously cleared by hardware.

Bit 4 **Bus Master Status (BM_STS)** (default = 0)—This bit is set when any system bus master requests the system bus, including all PCI master, ISA master and ISA DMA devices.

Bits 3–1 **Reserved** (always reads 0)

Bit 0 **Timer Carry Status (TMR_STS)** (default = 0)—This bit is set when the 23rd (31st) bit of the 24 (32) bit ACPI power management timer changes.

Power Management Enable Offset 3h–2h RW

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Reserved					RTCE	Rsvd	PBE	Reserved		GE	Reserved				ATE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The bits in this register correspond to the bits in the Power Management Status Register at Function 3, offset01h–00h.

Bit 15–11 **Reserved** (always reads 0)

Bit 10 **RTC Enable (RTC_EN)** (default = 0)—This bit can be set to trigger either an SCI or an SMI (depending on the setting of the SCI_EN bit) to be generated when the RTC_STS bit is set.

Bit 9 **Reserved** (always reads 0)

Bit 8 **Power Button Enable (PB_EN)** (default = 0)—This bit can be set to trigger either an SCI or an SMI (depending on the setting of the SCI_EN bit) to be generated when the PB_STS bit is set.

Bits 7–6 **Reserved** (always reads 0)

Bit 5 **Global Enable (GBL_EN)** (default = 0)—This bit can be set to trigger either an SCI or an SMI (depending on the setting of the SCI_EN bit) to be generated when the GBL_STS bit is set.

Bits 4–1 **Reserved** (always reads 0)

Bit 0 **ACPI Timer Enable (TMR_EN)** (default = 0)—This bit can be set to trigger either an SCI or an SMI (depending on the setting of the SCI_EN bit) to be generated when the TMR_STS bit is set.

Power Management Control Offset 05h–04h **RW**

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Reserved		SE	Sleep Type			Reserved							GR	BMR	SCIE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits 15–14 **Reserved** (always reads 0)

Bit 13 **Sleep Enable (SLP_EN)** (always reads 0)—This is a write-only bit. Reads from this bit always return zero. Writing a one to this bit causes the system to sequence into the sleep (suspend) state defined by the SLP_TYP field, bits 12–10.

Bits 12–10 **Sleep Type (SLP_TYP)**

000 = Soft Off (also called Suspend to Disk). The V_{DD5} power plane is turned off while the V_{DD}-5VSB and V_{DD}-RTC (VBAT) planes remain on.

010 = Power On Suspend. All power planes remain on but the processor is put into the C3 state.

0x1 = Reserved

1xx = Reserved

Note: To facilitate hardware design, minimal interface exists between powered and non-powered planes in either sleep state.

Bits 9–3 **Reserved** (always reads 0)

Bit 2 **Global Release (GBL_RLS)** (default = 0)—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 BIOS_STS bit. GBL_RLS is cleared by hardware when the BIOS_STS bit is cleared by software. Note that setting this bit will generate an SMI if the BIOS_EN bit is set (bit 5 of the Global Enable register at offset 2Ah).

Bit 1 **Bus Master Reload (BMS_RLD)** (default = 0)—This bit is used to enable the occurrence of a bus master request to transition the processor from the C3 state to the C0 state.

Bit 0 **SCI Enable (SCI_EN)**—This bit determines whether a power management event generates an SCI or SMI.

0 = Generate SMI (default)

1 = Generate SCI

Note: Certain power management events can be programmed individually to generate an SCI or SMI independent of the setting of this bit. Refer to the General Purpose SCI Enable and General Purpose SMI Enable registers at Function 3, offsets 22h and 24h, on page 7-63. Also,

TMR_STS & GBL_STS always generate an SCI and BIOS_STS always generates an SMI.

Power Management Timer Offset 0Bh–08h RW

	Bits 31–24	Bits 23–0
Bit Name	Extended Timer Value	Timer Value
Reset	0	0

Bits 31–24 **Extended Timer Value (ETM_VAL)**—This field reads back 0 if the 24-bit timer option is selected in configuration register Function 3, offset 41h, bit 3 (see page 7-55).

Bits 23–0 **Timer Value (TMR_VAL)**—This read-only field returns the running count of the power management timer. This timer is a 24-/32-bit counter driven by a 3.579545-MHz clock derived from an external 14.31818-MHz input when the system is in the S0 (working) state. The timer is reinitialized to zero during a reset and continues counting until the 14.31818 MHz input to the chip is stopped. The clock retains its value when the external timing source is stopped, and continues to count from that value when the clock is restarted without a reset.

Processor Power Management Registers

Processor Control Offset 13h-10h RW

	Bits 31–5	Bit 4	Bits 3–1	Bit 0
Bit Name	Reserved	TE	TDC	Reserved
Reset	0	0	0	0

Bits 31–5 **Reserved** (always reads 0)

Bit 4 **Throttling Enable (THT_EN)**—This bit determines the effect of reading the P_LVL2 port

0 = Reading the P_LVL2 port asserts $\overline{\text{STPCLK}}$ and suspends the processor

1 = Reading the P_LVL2 port enables clock throttling by modulating the $\overline{\text{STPCLK}}$ signal with a duty cycle determined by bits 3–1 of this register.

Bits 3–1 **Throttling Duty Cycle (THT_DTY)**—This 3-bit field determines the duty cycle of the STPCLK signal when the system is in throttling mode (i.e., THT_EN is set to one and the register P_LVL2 is read). The duty cycle indicates the percentage of time the STPCLK signal is asserted while the THT_EN bit is set. The field is decoded as follows:
000 = Reserved
001 = 0-12.5%
010 = 12.5-25%
011 = 25-37.5%
100 = 37.5-50%
101 = 50-62.5%
110 = 62.5-75%
111 = 75-87.5%

Bit 0 **Reserved** (always reads 0)

Processor Level 2 (P_LVL2) Offset 14h — RO

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Processor Level 2							
Reset	0	0	0	0	0	0	0	0

Bits 7–0 **LVL2** (always reads 0)—Reads from this register put the processor in the C2 clock state determined by the THT_EN bit. Reads from this register return all zeros; writes to this register have no effect.

Processor Level 3 (P_LVL3) Offset 15h — RO

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Processor Level 3							
Reset	0	0	0	0	0	0	0	0

Bits 7–0 **LVL3** (always reads 0)—Reads from this register put the processor in the C3 clock state with the STPCLK signal asserted. Reads from this register return all zeros. Writes to this register have no effect.

General Purpose Power Management Registers

General Purpose Status (GP_STS) Offset 21h–20h — RWC

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Reserved						USBS	RS	ES7	ES6	ES5	ES4	ES3	ES2	ES1	ES0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits 15–10 **Reserved** (always reads 0)

Bit 9 **USB Resume Status (USB_STS)**—This bit is set when a USB peripheral generates a resume event.

Bit 8 **Ring Status (RI_STS)**—This bit is set when the RI input is asserted low.

Bit 7 **EXTSMI7 Toggle Status (EXT7_STS)**—This bit is set when the EXTSMI7 pin is toggled.

Bit 6 **EXTSMI6 Toggle Status (EXT6_STS)**—This bit is set when the EXTSMI6 pin is toggled.

Bit 5 **EXTSMI5 Toggle Status (EXT5_STS)**—This bit is set when the EXTSMI5 pin is toggled.

Bit 4 **EXTSMI4 Toggle Status (EXT4_STS)**—This bit is set when the EXTSMI4 pin is toggled.

Bit 3 **EXTSMI3 Toggle Status (EXT3_STS)**—This bit is set when the EXTSMI3 pin is toggled.

Bit 2 **EXTSMI2 Toggle Status (EXT2_STS)**—This bit is set when the EXTSMI2 pin is toggled.

Bit 1 **EXTSMI1 Toggle Status (EXT1_STS)**—This bit is set when the EXTSMI1 pin is toggled.

Bit 0 **EXTSMI0 Toggle Status (EXT0_STS)**—This bit is set when the EXTSMI0 pin is toggled.

Note: *Bits 9–0 correspond one-for-one with the bits of the General Purpose SCI Enable (offset 23h–22h) and General Purpose SMI Enable registers (offset 25h–24h). An SCI or SMI is generated if the corresponding bit of the General Purpose SCI or SMI Enable registers, respectively, is set. Bits 9–0 are set only by hardware and can be cleared only by writing a one to the desired bit.*

General Purpose SCI Enable Offset 23h–22h RW

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Reserved						EUSB	ERI	E7	E6	E5	E4	E3	E2	E1	E0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit 15–10 **Reserved** (always reads 0)

Bit 9 **Enable SCI on setting of the USB_STS bit** (default = 0)

Bit 8 **Enable SCI on setting of the RI_STS bit** (default = 0)

Bit 7 **Enable SCI on setting of the EXT7_STS bit** (default = 0)

Bit 6 **Enable SCI on setting of the EXT6_STS bit** (default = 0)

Bit 5 **Enable SCI on setting of the EXT5_STS bit** (default = 0)

Bit 4 **Enable SCI on setting of the EXT4_STS bit** (default = 0)

Bit 3 **Enable SCI on setting of the EXT3_STS bit** (default = 0)

Bit 2 **Enable SCI on setting of the EXT2_STS bit** (default = 0)

Bit 1 **Enable SCI on setting of the EXT1_STS bit** (default = 0)

Bit 0 **Enable SCI on setting of the EXT0_STS bit** (default = 0)
These bits allow generation of an SCI using a separate set of conditions from those used for generating an SMI.

General Purpose SMI Enable Offset 25h–24h RW

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Reserved						EUSB	ERI	E7	E6	E5	E4	E3	E2	E1	E0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit 15–10 **Reserved** (always reads 0)

Bit 9 **Enable SMI on setting of the USB_STS bit** (default = 0)

Bit 8 **Enable SMI on setting of the RI_STS bit** (default = 0)

Bit 7 **Enable SMI on setting of the EXT7_STS bit** (default = 0)

Bit 6 **Enable SMI on setting of the EXT6_STS bit** (default = 0)

Bit 5 **Enable SMI on setting of the EXT5_STS bit** (default = 0)

Bit 4 **Enable SMI on setting of the EXT4_STS bit** (default = 0)

Bit 3 **Enable SMI on setting of the EXT3_STS bit** (default = 0)

Bit 2 **Enable SMI on setting of the EXT2_STS bit** (default = 0).

Bit 1 **Enable SMI on setting of the EXT1_STS bit** (default = 0)

Bit 0 **Enable SMI on setting of the EXT0_STS bit** (default = 0)
These bits allow generation of an SMI using a separate set of conditions from those used for generating an SCI.

Power Supply Control Offset 27h–26h RW

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Reserved					RPSC	PBC	RSC	Reserved							ES0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit 15–11 **Reserved** (always reads 0)

Bit10 **Ring PS Control (RI_PS_CTL)** (default = 0)—This bit enables setting the RI_STS bit to turn on the V_{DD}_5V power plane by setting PWRON = 1.

Bit 9 **Power Button Control (PB_CTL)** (default = 0)—This bit is used to set the PB_STS bit to resume the system from suspend (turn on the V_{DD}_5V power plane by setting PWRON = 1).

Bit 8 **RTC PS Control (RTC_PS_CTL)** (default = 0)—This bit enables setting the RTC_STS bit to resume the system from suspend (turn on the V_{DD}_5V power plane by setting PWRON = 1).

Bit 7–1 **Reserved** (always reads 0)

Bit 0 **EXTSMI0 Toggle PS Control (E0_PS_CTL)** (default = 0)—This bit enables the setting of the EXT0_STS bit to resume the system from suspend (turn on the V_{DD}_5V power plane by setting PWRON = 1).

Generic Power Management Registers

Global Status Offset 29h–28h — RWC

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Reserved									SSS	BS	LUS	G1TS	G2TS	SETO	PAS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit 15–7 **Reserved** (always reads 0)

Bit 6 **Software SMI Status (SW_SMI_STS)** (default = 0)—This bit is set when the SMI_CMD port (offset 2Fh) is written.

Bit 5 **BIOS Status (BIOS_STS)** (default = 0)—This bit is set when the GBL_RLS bit is set (typically by the ACPI software to release control of the SCI/SMI lock). When this bit is reset (by writing a one to this bit position) the GBL_RLS bit is reset at the same time by hardware.

Bit 4 **Legacy USB Status (LEG_USB_STS)** (default = 0)—This bit is set when a legacy USB event occurs.

Bit 3 **GP1 Timer Time Out Status (GP1TO_STS)** (default = 0)—This bit is set when the GP1 timer times out.

Bit 2 **GP0 Timer Time Out Status (GP0TO_STS)** (default = 0)—This bit is set when the GP0 timer times out.

Bit 1 **Secondary Event Timer Time Out Status (STTO_STS)** (default = 0)—This bit is set when the secondary event timer times out.

Bit 0 **Primary Activity Status (PACT_STS)** (default = 0)—This bit is set at the occurrence of any enabled primary system activity (see the Primary Activity Detect Status register at offset 30h, page 7-67, and the Primary Activity Detect Enable register at offset 34h, page 7-68). After checking this bit, software can check the status bits in the Primary Activity Detect Status register at offset 30h to identify the specific source of the primary event. Setting this bit can be enabled to reload the GP0 timer (see bit 0 of the GP Timer Reload Enable register at offset 38h, page 7-69). Note that an SMI can be generated based on the setting of any of the above bits (see the bit descriptions of the Global Enable register, offset 2Ah, page 7-66). The bits in this register are set only by hardware and can be cleared only by writing a one to the desired bit position.

Global Enable Offset 2Bh–2Ah RW

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Reserved									SSE	BE	LUE	G1E	G0E	SETE	PAE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit 15–7 **Reserved** (always reads 0)

Bit 6 **Software SMI Enable (SW_SMI_EN)** (default = 0)—This bit can be set to trigger an SMI when the SW_SMI_STS bit is set.

Bit 5 **BIOS Enable (BIOS_EN)** (default = 0)—This bit can be set to trigger an SMI when the BIOS_STS bit is set.

Bit 4 **Legacy USB Enable (LEG_USB_EN)** (default = 0)—This bit can be set to trigger an SMI when the LEG_USB_STS bit is set.

Bit 3 **GP1 Timer Time Out Enable (GP1TO_EN)** (default = 0)—This bit can be set to trigger an SMI when the GP1TO_STS bit is set.

Bit 2 **GP0 Timer Time Out Enable (GP0TO_EN)** (default = 0)—This bit can be set to trigger an SMI when the GP0TO_STS bit is set.

Bit 1 **Secondary Event Timer Time Out Enable (STTO_EN)** (default = 0) This bit can be set to trigger an SMI when the STTO_STS bit is set.

Bit 0 **Primary Activity Enable (PACT_EN)** (default = 0)—This bit can be set to trigger an SMI when the PACT_STS bit is set.

Global Control (GBL_CTL) Offset 2Dh–2Ch RW

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Reserved							SA	Reserved			SMIL	Rsvd	BPT	BR	SMIE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit 15–9 **Reserved** (always reads 0)

Bit 8 **SMI Active (INSMI)**
0 = SMI Inactive (default)
1 = SMI Active. If bit 4 (SMIIG) is set, bit 8 must be cleared by writing a 1 to it before the next SMI can be generated.

Bit 7–5 **Reserved** (always reads 0)

Bit 4 **SMI Lock (SMIIG) (RWC)**
0 = Disable SMI Lock (default)
1 = Enable SMI Lock (SMI low to gate for the next SMI)

Bit 3 **Reserved** (always reads 0)

Bit 2 **Power Button Triggering**—Set this bit to prevent the situation in which PB_STS is set to wake up the system, then reset by PBOR_STS to switch the system into the soft-off state. Bit 2 must be cleared to comply with ACPI v. 0.9.
0 = SCI/SMI generated by PWRBTN low level
1 = SCI/SMI generated by PWRBTN rising edge

Bit 1 **BIOS Release (BIOS_RLS)**—This bit is set by legacy software to indicate release of the SCI/SMI lock. Upon setting of this bit, hardware automatically sets the GBL_STS bit. This bit is cleared by hardware when the GBL_STS bit cleared by software.

Note: If the GBL_EN bit is set (bit 5 of the Power Management Enable register at offset 2h), then setting this bit causes an SCI to be generated (because setting this bit causes the GBL_STS bit to be set).

Bit 0 **SMI Enable (SMI_EN)**
0 = Disable all SMI generation
1 = Enable SMI generation

SMI Command (SMI_CMD) Offset 2Fh — RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	SMI Command							
Reset	0	0	0	0	0	0	0	0

Bit 7-0 **SMI Command**—Writing to this port sets the SW_SMI_STS bit. Note that if the SW_SMI_EN bit is set (see bit 6 of the Global Enable register at offset 2Ah), then an SMI is generated.

Primary Activity Detect Status Offset 33h-30h — RWC

	Bit 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	Bit 16
	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Reserved								KCAS	SPAS	PPAS	VAS	IFAS	Rsvd	PIAS	IDAS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

These bits correspond to the Primary Activity Detect Enable bits in offset 37h–34h.

Bit 31-8 **Reserved** (always reads 0)

Bit 7 **Keyboard Controller Access Status (KBC_STS)**—Set if the keyboard controller is accessed via I/O port 60h.

Bit 6 **Serial Port Access Status (SER_STS)**—Set if the serial port is accessed via I/O ports 3F8h–3FFh, 2F8h–2FFh, 3E8h–3EFh, or 2E8h–2Efh (COM1–4, respectively).

Bit 5 **Parallel Port Access Status (PAR_STS)**—Set if the parallel port is accessed via I/O ports 278h–27Fh or 378h–37Fh (LPT2 or LPT1).

Bit 4 **Video Access Status (VID_STS)**—Set if the parallel port is accessed via I/O ports 278h–27Fh or 378h–37Fh (LPT2 or LPT1).

Bit 3 **IDE / Floppy Access Status (IDE_STS)**—Set if the parallel port is accessed via I/O ports 278h–27Fh or 378h–37Fh (LPT2 or LPT1).

Bit 2 **Reserved** (always reads 0)

Bit 1 **Primary Interrupt Activity Status (PIRQ_STS)**—This bit is set when a primary interrupt occurs. Primary interrupts are enabled in the Primary Interrupt Channel register at Function 3, PCI configuration register offset 44h (see page 7-56).

Bit 0 **ISA Master / DMA Activity Status (DRQ_STS)**—This bit is set by ISA master or DMA activity.

The bits in this register correspond to the bits in the Primary Activity Detect Enable register at offset 34h (page 7-67). If the corresponding bit is set in that register, setting the bit in this register will cause the PACT_STS bit to be set (bit 0 of the Global Status register at offset 28h, page 7-65). Setting of PACT_STS can be set up to enable a "Primary Activity Event", where an SMI will be generated if PACT_EN is set (bit 0 of the Global Enable register at offset 2Ah, page 7-66) and/or the GP0 timer will be reloaded if the GP0 Timer Reload on Primary Activity bit is set (bit 0 of the GP Timer Reload Enable register at offset 38h, page 7-69).

Bits 3–7 in this register also correspond to bits 3–7 of the GP Timer Reload Enable register at offset 38h. If the corresponding bit is set in that register, setting the bit in this register will cause the GP1 timer to be reloaded.

All bits of this register are set only by hardware and can be cleared only by writing a one to the desired bit. All bits default to 0.

Primary Activity Detect Enable Offset 37h–34h RW

	Bit 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	Bit 16
	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Reserved								KCSE	SPSE	PPSE	VSE	IFSE	Rsvd	PISE	IDSE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

These bits correspond to the Primary Activity Detect Status bits in offset 33h–30h.

Bit 31–8 **Reserved** (always reads 0)

Bit 7 **Keyboard Controller Status Enable (KBC_EN)**
0 = Don't set PACT_STS if KBC_STS is set (default)
1 = Set PACT_STS if KBC_STS is set

Bit 6 **Serial Port Status Enable (SER_EN)**
0 = Don't set PACT_STS if SER_STS is set (default)
1 = Set PACT_STS if SER_STS is set

Bit 5 **Parallel Port Status Enable (PAR_EN)**
0 = Don't set PACT_STS if PAR_STS is set (default)
1 = Set PACT_STS if PAR_STS is set

Bit 4 **Video Status Enable (VID_EN)**
0 = Don't set PACT_STS if VID_STS is set (default)
1 = Set PACT_STS if VID_STS is set

Bit 3 **IDE / Floppy Status Enable (IDE_EN)**
0 = Don't set PACT_STS if IDE_STS is set (default)
1 = Set PACT_STS if IDE_STS is set

Bit 2 **Reserved** (always reads 0)

Bit 1 **Primary INTR Status Enable (PIRQ_EN)**
0 = Don't set PACT_STS if PIRQ_STS is set (default)
1 = Set PACT_STS if PIRQ_STS is set

Bit 0 **ISA Master / DMA Status Enable (DRQ_EN)**
0 = Don't set PACT_STS if DRQ_STS is set (default)
1 = Set PACT_STS if DRQ_STS is set

Note: *Setting any of bits 7–0 also sets the PACT_STS bit (bit 0 of offset 28h), which reloads the GP0 timer (if PACT_GP0_EN is set) or generates an SMI (if PACT_EN is set).*

GP Timer Reload Enable Offset 3Bh–38h RW

	Bit 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	Bit 16
	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	Reserved								E1KA	K1SA	Rsvd	E1VA	E1IA	Reserved		IDSE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

All bits in this register default to 0 on power up.

Bit 31–8 **Reserved** (always reads 0)

Bit 7 **Enable GP1 Timer Reload on KBC Access**
1 = setting KBC_STS causes GP1 timer to reload

Bit 6 **Enable GP1 Timer Reload on Serial Port Access**
1 = setting SER_STS causes GP1 timer to reload

Bit 5 **Reserved** (always reads 0)

Bit 4 **Enable GP1 Timer Reload on Video Access**
1 = setting VID_STS causes GP1 timer to reload

Bit 3 **Enable GP1 Timer Reload on IDE/Floppy Access**
1 = setting IDE_STS causes GP1 timer to reload

Bit 2–1 **Reserved** (always reads 0)

Bit 0 **Enable GP0 Timer Reload on Primary Activity**
1 = setting PACT_STS causes GP0 timer to reload. Primary activities are enabled via the Primary Activity Detect Enable register (offset 37h–34h) with status recorded in the Primary Activity Detect Status register (offset 33h–30h).

General Purpose I/O Registers

GPIO Direction Control (GPIO_DIR) Offset 40h RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Reserved			G4D	G3D	G2D	G1D	G0D
Reset	0	0	0	0	0	0	0	0

Bit 7–5 **Reserved** (always reads 0)

Bit 4 **GPIO4_DIR**
0 = Pin 136 is GPIO4 input (default)
1 = Pin 136 is GPIO4 output (if offset 40h bit 7 = 1)
If offset 40h bit 7 = 0 (PCI Configuration Function 3 offset 40h GPIO4_CFG bit), pin 136 is the GPO_WE output, independent of the state of this bit.

Bit 3 **GPIO3_DIR**
0 = Pin 92 is GPIO3 input (default)
1 = Pin 92 is GPIO3 output (if offset 40h bit 6 = 1)
If offset 40h bit 6 = 0 (PCI Configuration Function 3 offset 40h GPIO3_CFG bit), pin 92 is the $\overline{\text{GPI\_RE}}$ output, independent of the state of this bit.

Bit 2 **GPIO2_DIR**
0 = Pin 88 is GPIO2 / I2CD1 input (default)
1 = Pin 88 is GPIO2 / I2CD1 output

Bit 1 **GPIO1_DIR**
0 = Pin 87 is GPIO1 / I2CD2 input (default)
1 = Pin 87 is GPIO1 / I2CD2 output

Bit 0 **GPIO0_DIR**
0 = Pin 94 is GPIO0 input (default)
1 = Pin 94 is GPIO0 output

GPIO Port Output Value (GPIO_VAL) Offset 42h RW

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	Reserved			G4V	G3V	G2V	G1V	G0V
Reset	0	0	0	0	0	0	0	0

Bit 7–5 **Reserved** (always reads 0)

Bit 4 **GPIO4_VAL**—Write output value for the GPIO4 pin if the port is available (GPIO4_CFG = 1 in PCI configuration register Function 3, offset 40h). The input state of the GPIO4 pin can be read from register EXTSMI_VAL bit 4.

Bit 3 **GPIO3_VAL**—Write output value for the GPIO3 pin if the port is available (GPIO3_CFG = 1 in PCI configuration register Function 3, offset 40h). The input state of the GPIO3 pin can be read from register EXTSMI_VAL bit 3.

Bit 2 **GPIO2_VAL**—Write output value for the GPIO2 (I2CD2) pin. The input state of the GPIO2 pin can be read from register EXTSMI_VAL bit 2.

Bit 1 **GPIO1_VAL**—Write output value for the GPIO1 (I2CD1) pin. The input state of the GPIO1 pin can be read from register EXTSMI_VAL bit 1.

Bit 0 **GPIO0_VAL**—Write output value for the GPIO0 pin. The input state of the GPIO0 pin can be read from register EXTSMI_VAL bit 0.

GPIO Port Input Value (EXTSMI_VAL) Offset 44h RO

	Bit 7	6	5	4	3	2	1	Bit 0
Bit Name	E7IV	E6IV	E5IV	E4IV	E3IV	E2IV	E1IV	E0IV
Reset	0	0	0	0	0	0	0	0

Depending on the configuration, up to eight external SCI/SMI ports are available as indicated below. The state of inputs $\overline{\text{EXTSMI7}}$–$\overline{\text{EXTSMI0}}$ can be read in this register.

Bit 7 **$\overline{\text{EXTSMI7}}$ Input Value**
GPIO3_CFG = 0: $\overline{\text{EXTSMI7}}$ on XD7 (pin 122)
GPIO3_CFG = 1: $\overline{\text{EXTSMI7}}$ function not available

Bit 6 **$\overline{\text{EXTSMI6}}$ Input Value**
GPIO3_CFG = 0: $\overline{\text{EXTSMI6}}$ on XD6 (pin 121)
GPIO3_CFG = 1: $\overline{\text{EXTSMI6}}$ function not available

Bit 5 **$\overline{\text{EXTSMI5}}$ Input Value**
GPIO3_CFG = 0: $\overline{\text{EXTSMI5}}$ on XD5 (pin 119)
GPIO3_CFG = 1: $\overline{\text{EXTSMI5}}$ function not available

Bit 4 **EXTSMI4 Input Value**
GPIO4_CFG = 0:
GPIO3_CFG = 0: EXTSMI4 on XD4 (pin 118)
GPIO3_CFG = 1: EXTSMI4 function not available
GPIO4_CFG = 1: EXTSMI4 on GPIO4 (pin 136)

Bit 3 **EXTSMI3 Input Value**
GPIO3_CFG = 0: EXTSMI3 on XD3 (pin 117)
GPIO3_CFG = 1: EXTSMI3 on GPIO3 (pin 92)

Bit 2 **EXTSMI2 Input Value (on GPIO2 pin 88)**

Bit 1 **EXTSMI1 Input Value (on GPIO1 pin 87)**

Bit 0 **EXTSMI0 Input Value (on GPIO0 pin 94)**

Note: GPIO3_CFG and GPIO4_CFG are located in PCI configuration register Function 3, offset 40h.

GPO Port Output Value (GPO_VAL) Offset 47h–46h RW

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	GPO15–8 value								GPO7–0 value							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Reads from this register return the last value written (held on chip).

Bit 15–8 **GPO15–8 Value.**—Output port value for the external GPO port connected to SD15–8. This port is available only if the GPIO4_CFG bit is cleared to define pin 136 as GPO_WE.

Bit 7–0 **GPO7–0 Value.**—Output port value for the external GPO port connected to XD7–0. This port is available only if the GPIO4_CFG bit is cleared to define pin 136 as GPO_WE.

Note: GPIO4_CFG is in PCI register Function 3, offset 40h, page 7-54.

GPI Port Input Value (GPI_VAL) Offset 49h–48h RO

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	GPI15–8 value								GPI7–0 value							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Reads from this register are ignored (and return a value of 0).

Bit 15-8 **GPI15-8 Value.** Input port value for the external GPI port connected to SD15-8. This port is available only if the GPIO3_CFG bit is cleared to define pin 92 as GPI_RE.

Bit 7-0 **GPI7-0 Value.** Input port value for the external GPI port connected to XD7–XD0. This port is available only if the GPIO3_CFG bit is cleared to define pin 92 as GPI_RE.

Note: GPIO3_CFG is in PCI configuration register Function 3, offset 40h.

8 Electrical Data

8.1 Absolute Ratings

Long-term reliability and functional integrity of the AMD-645 Peripheral Bus Controller are guaranteed as long as it is not subjected to conditions exceeding the absolute ratings listed in Table 11-1.

Table 8-1. Absolute Maximum Ratings

Parameter	Minimum	Maximum	Unit
Ambient Operating Temperature	0	70	°C
Storage Temperature	-55	125	°C
Input Voltage	-0.5	5.5	Voltage
Output Voltage (V_{DD} = 5 V)	-0.5	5.5	Voltage
Output Voltage (V_{DD} = 3.1 V - 3.6 V)	-0.5	V_{DD} + 0.5	Voltage

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.

Table 8-2. Absolute Ratings

Parameter	Minimum	Maximum	Comments
V_{DD}	-0.5 V	5.5 V	
V_{DD3}	-0.5 V	3.6	
V_{PIN}	-0.5 V	V_{DD3}+0.5 V and ≤ 4.0 V	note 1
T_{CASE} (under bias)	-65°C	+110°C	
$T_{STORAGE}$	-65°C	+150°C	

Note:

1. *V_{PIN} (the voltage on any I/O pin) must not be greater than 0.5 V above the voltage being applied to V_{DD3}. In addition, the V_{PIN} voltage must never exceed 4.0 V.*

8.2 Operating Ranges

The functional operation of the AMD-645 Peripheral Bus Controller is guaranteed if the voltage and temperature parameters are within the limits defined in Table 11-2.

Table 8-3. Operating Ranges

Parameter	Minimum	Typical	Maximum	Comments
V_{DD}	4.75 V	5.0 V	5.25 V	(note 1)
V_{DD3}	3.135 V	3.3 V	3.465 V	(note 1)
T_{CASE}	0°C		70°C	

Note:
1. V_{DD} and V_{DD3} are referenced from V_{SS}

8.3 DC Characteristics

Table 8-4. DC Characteristics

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		0.45 V	I_{OL} = 4.0-mA load
V_{OH}	Output High Voltage	2.4 V		I_{OH} = 1.0-mA load
I_{DD}	5 V Power Supply Current			33 MHz, Note 2
I_{DD3}	3.3 V Power Supply Current			33 MHz, Note 3
I_{LI}	Input Leakage Current		±10 µA	note 4
I_{LO}	Output Leakage Current		±20 µA	note 4

Notes:
1. V_{DD3} refers to the voltage being applied to V_{DD3} during functional operation.
2. V_{DD} = 5.25 V – The maximum power supply current must be taken into account when designing a power supply.
3. V_{DD3} = 3.465 V – The maximum power supply current must be taken into account when designing a power supply.
4. Refers to inputs and I/O without an internal pullup resistor and $0 \leq V_{IN} \leq V_{DD3}$.
5. Refers to inputs with an internal pullup and V_{IL} = 0.4 V.
6. Refers to inputs with an internal pulldown and V_{IH} = 2.4 V.

Table 8-4. DC Characteristics (continued)

Symbol	Parameter Description	Preliminary Data		Comments
		Min	Max	
I_{IL}	Input Leakage Current Bias with Pullup		-10 µA	note 5
I_{IH}	Input Leakage Current Bias with Pulldown		10 µA	note 6
C_{IN}	Input Capacitance		10 pF	
C_{OUT}	Output Capacitance		15pF	
C_{OUT}	I/O Capacitance		20 pF	
C_{CLK}	CLK Capacitance		10 pF	
C_{TIN}	Test Input Capacitance (TDI, TMS, $\overline{TRST}$)		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.*
2. *V_{DD} = 5.25 V – The maximum power supply current must be taken into account when designing a power supply.*
3. *V_{DD3} = 3.465 V – The maximum power supply current must be taken into account when designing a power supply.*
4. *Refers to inputs and I/O without an internal pullup resistor and $0 \leq V_{IN} \leq V_{DD3}$.*
5. *Refers to inputs with an internal pullup and V_{IL} = 0.4 V.*
6. *Refers to inputs with an internal pulldown and V_{IH} = 2.4 V.*

8.4 Power Dissipation

Table 11-4 shows typical and maximum power dissipation of the AMD-645 Peripheral Bus Controller during normal and reduced power states. The measurements are taken with PCLK = 33 MHz, V_{DD} = 5.0V and $V_{DD}3$ = 3.3V.

Table 8-5. Typical and Maximum Power Dissipation

Clock Control State	**Typical** (Note 1)	**Maximum** (Note 2)	Comments
Normal (Thermal Power)	2.3 W?	0.40 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-645 Peripheral Bus Controller processor.*

9 Switching Characteristics

This section summarizes the AMD-645 Peripheral Bus Controller signal switching characteristics. Valid delay, float, setup, and hold timing specifications are listed.

All signal timings are based on the following conditions:

- 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 (CCLK, 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 8-1 on page 8-1.
- The load capacitance (C_L) on each signal is 0 pF.

9.1 PCLK Switching Characteristics

Table 9-1 contains the switching characteristics of the PCLK input to the AMD-645 Peripheral Bus Controller as measured at the voltage levels indicated by Figure 9-1.

The PCLK period stability specifies the variance (jitter) allowed between successive periods of the CLK input measured at 1.5 V. This parameter must be considered as one of the elements of clock skew between the AMD-645 Peripheral Bus Controller and the system logic.

Table 9-1. CLK Switching Characteristics for 33-MHz PCI Bus

Symbol	Parameter Description	Preliminary Data		Figure	Comments
		Min	Max		
t_1	CLK cycle	30 ns	∞		
t_2	CLK high time	11.0 ns		9-1	
t_3	CLK Low time	11.0 ns		9-1	
t_4	CLK fall time	1 V/ns	4V/ns	9-1	
t_5	CLK rise time	1 V/ns	4V/ns	9-1	
	CLK period stability		± 250 ps		note 1

Note:
1. *Jitter frequency power spectrum peaking must occur at frequencies greater than (CCLK frequency)/3 or less than 500 KHz.*

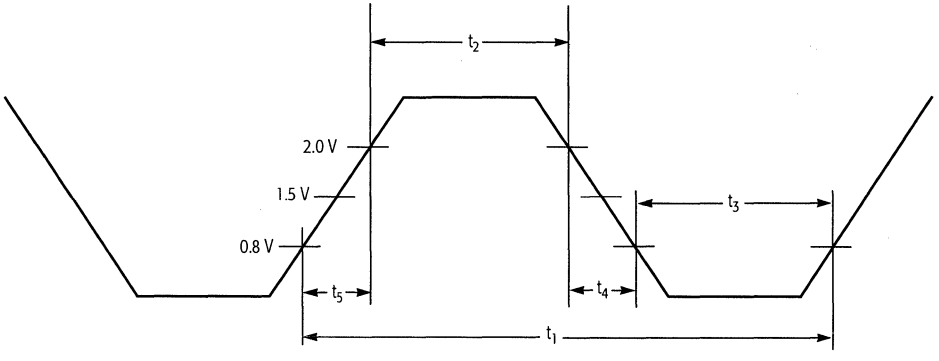


Figure 9-1. PCLK Waveform

Table 9-2. USBCLK Switching Characteristics for 12-MHz USB Bus

Symbol	Parameter Description	Preliminary Data		Figure	Comments
		Min	Max		
t_1	Driver jitter		3 ns		
t_2	Receiver jitter		25 ns	9-1	
t_3	Output fall time	4 ns	20 ns	9-1	
t_4	Output rise time	4 ns	20 ns	9-1	
t_5	Source differential skew		5 ns	9-1	
	Receiver differential skew		10 ns		
	Single-ended driver skew		10 ns		
	Frequency	11.97 Mbps	12.03 Mbps		

Note:
Jitter frequency power spectrum peaking must occur at frequencies greater than (CCLK frequency)/3 or less than 500 KHz.

Table 9-3. USBCLK Switching Characteristics for 1.5-MHz USB Bus

Symbol	Parameter Description	Preliminary Data		Figure	Comments
		Min	Max		
t_1	Driver jitter		3 ns		
t_2	Receiver jitter		25 ns	9-1	
t_3	Output fall time	75 ns	300 ns	9-1	
t_4	Output rise time	75ns	300 ns	9-1	
t_5	Source differential skew		5 ns	9-1	
	Receiver differential skew		10 ns		
	Single-ended driver skew		10 ns		
	Frequency	1.48 Mbps	1.52 Mbps		

Note:
Jitter frequency power spectrum peaking must occur at frequencies greater than (CCLK frequency)/3 or less than 500 KHz.

Table 9-4. BCLK Switching Characteristics for 8-MHz Bus

Symbol	Parameter Description	Preliminary Data		Figure	Comments
		Min	Max		
	Frequency		8 MHz		
t_1	Clock period	125 ns			
t_2	Clock High time	49 ns		9-1	
t_3	Clock Low time	49 ns	s	9-1	
t_4	Clock rise time		4 ns	9-1	
t_5	Clock fall time		4 ns	9-1	

Note:
Jitter frequency power spectrum peaking must occur at frequencies greater than (CCLK frequency)/3 or less than 500 KHz.

Table 9-5. OSC Switching Characteristics for 14.3182-MHz Bus

Symbol	Parameter Description	Preliminary Data		Figure	Comments
		Min	Max		
	Frequency		14.3182 MHz	9-1	
t_1	Clock period	67 ns	70 ns	9-1	
t_2	Clock High time	20 ns		9-1	
t_3	Clock Low time	20 ns	s	9-1	

Note:
Jitter frequency power spectrum peaking must occur at frequencies greater than (CCLK frequency)/3 or less than 500 KHz.

9.2 Valid Delay, Float, Setup, and Hold Timings

The following valid delay and float timings for output signals during functional operation are relative to the rising edge of the given clock. The maximum valid delay timings are provided to allow a system designer to determine if setup times can be met. Likewise, the minimum valid delay timings are used to analyze hold times.

The setup and hold time requirements for the AMD-645 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-645 Peripheral Bus Controller.

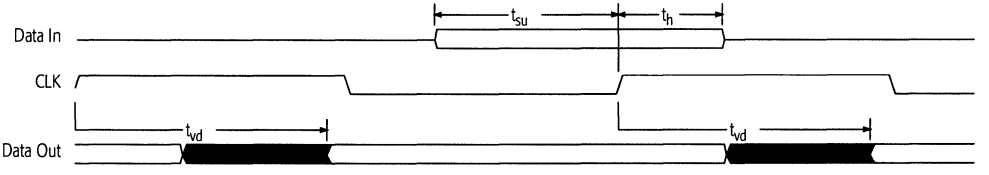


Figure 9-2. Setup, Hold, and Valid Delay Timing Diagram

9.3 PCI Interface Timing

Table 9-6. PCI Interface Timing

Symbol	Parameter Description	Preliminary Data		Figure	Comments
		Min	Max		
t_{su}	AD31–AD0 setup time	7 ns		9-2	
	PREQ, REQD–REQA setup Time	12 ns		9-2	
	Setup time for FRAME, STOP, TRDY, DEVSEL, IRDY, C/BE3–C/BE0	7 ns		9-2	
t_h	AD31–AD0 hold Time	0 ns		9-2	
	Hold time for FRAME, STOP, TRDY, DEVSEL, IRDY, C/BE3–C/BE0	0 ns		9-2	
t_{vd}	AD31–AD0 valid delay (address phase)	2 ns	11 ns	9-2	Pad 12 (note 1)
	AD31–AD0 valid delay (data phase)	2 ns	11 ns	9-2	Pad 12 (note 1)
	valid delay for FRAME, STOP, TRDY, DEVSEL, IRDY C/BE3–C/BE0	2 ns	11 ns	9-2	Pad 13 (note 1)
	PGNT valid delay	2 ns	12 ns	9-2	
t_{fd}	Float delay for FRAME, STOP, TRDY, DEVSEL, IRDY C/BE3–C/BE0		28 ns	9-2	(note 1)
t_{lat}	PREQ to PGNT Latency	2 clks	clks	9-2	

Note:
1. *Measurements are taken with a 50pF load, unless otherwise noted.*

9.4 ISA Interface Timing

Table 9-7. ISA Master Interface Timing

Symbol	Parameter Description	Preliminary Data		Figure	Comments
		Min	Max		
t_{su2a}	LA23–LA17 setup to BALE	150 ns		9-3	
t_{su2b}	LA23–LA17 setup to $\overline{MEMx}$	173 ns		9-3	
t_{su3a}	SA19–SA0 setup to BALE	37 ns		9-3	
t_{su3b}	SA19–SA0 setup time to $\overline{MEMx}$	34 ns		9-3	
t_{su9}	SD15–SD0 setup to $\overline{MEMR}$	24 ns		9-3	
t_{su10}	SD15–SD0 setup to $\overline{MEMW}$	-40 ns		9-3	
t_{h2}	BALE to LA23–LA17 hold time	26 ns		9-3	
t_{h3}	$\overline{MEMx}$ to SA19–SA0Hold	41 ns		9-3	
t_{h6}	LA23–LA17 to $\overline{MEMCS16}$ hold	0 ns		9-3	
t_{h9}	$\overline{MEMR}$ to SD15–SD0 hold time	0 ns		9-3	
t_{h10}	$\overline{MEMW}$ to SD15–SD0 hold time	45 ns		9-3	
t_{vd1}	$\overline{MEMx}$ to BALE valid delay	44 ns		9-3	
t_{vd5}	$\overline{MEMx}$ to $\overline{SMEMR}$ & $\overline{SMEMW}$		16 ns	9-3	
t_{vd6a}	SA19–SA0, $\overline{SBHE}$ to $\overline{MEMCS16}$ valid delay		35 ns	9-3	
t_{vd6b}	LA23–LA17 to $\overline{MEMCS16}$ valid delay		94 ns	9-3	
t_{vd7a}	SA19–SA0, $\overline{SBHE}$ to $\overline{ZEROWS}$ delay		200 ns	9-3	
t_{vd7b}	$\overline{MEMW}$ to $\overline{ZEROWS}$ delay		16 ns	9-3	
t_{vd8}	$\overline{MEMx}$ to IOCHRD valid delay		78 ns	9-3	
t_{vd9}	$\overline{MEMR}$ to SD15–SD0 valid delay		150 ns	9-3	
t_{pw1}	BALE pulse width	50 ns		9-3	
t_{pw4a}	$\overline{MEMx}$ active pulse width	225 ns		9-3	
t_{pw4b}	$\overline{MEMx}$ inactive pulse width	163 ns		9-3	
t_{pw8}	IOCHRDY inactive pulse width	120 ns		9-3	
t_{fd9}	$\overline{MEMR}$ to SD15–SD0 float delay		41 ns	9-3	
t_{fd10}	$\overline{MEMW}$ to SD15–SD0 float delay		105 ns	9-3	

Note:
Measurements are taken with no load.

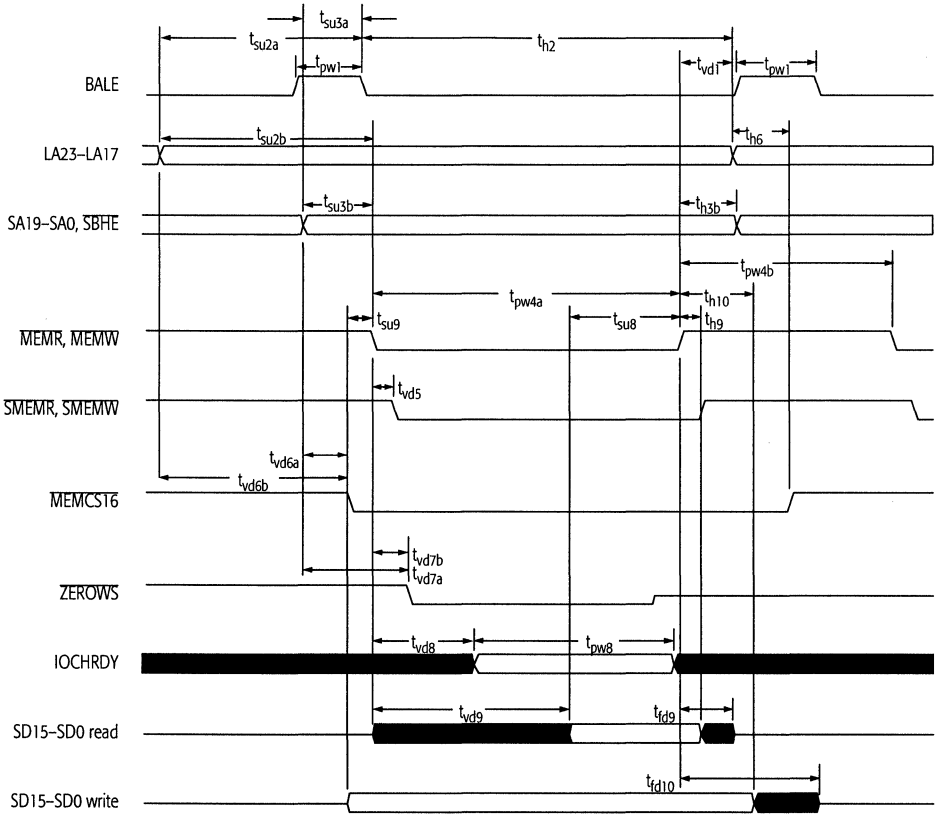


Figure 9-3. ISA Master Interface Timing

Table 9-8. ISA 8-Bit Slave Interface Timing

Symbol	Parameter Description	Preliminary Data		Figure	Comments
		Min	Max		
t_{su2a}	AEN setup to BALE	111 ns		9-4	
t_{su2b}	AEN setup to $\overline{IOx}$	111 ns		9-4	
t_{su3a}	SA19–SA0 setup to BALE	37 ns		9-4	
t_{su3b}	SA19–SA0 setup to $\overline{IOx}$	100 ns		9-4	
t_{su8}	SD15–SD0 setup to $\overline{IOR}$	24 ns		9-4	
t_{su9}	SD15–SD0 setup to $\overline{IOW}$	-40 ns		9-4	
t_{h2}	$\overline{IOx}$ to AEN hold	41 ns		9-4	
t_{h3}	$\overline{IOx}$ to SA19–SA0Hold	41 ns		9-4	
t_{h4a}	$\overline{IOR}$ to SD15–SD0 hold	0 ns		9-4	
t_{h4b}	$\overline{IOW}$ to SD15–SD0 hold	45 ns		9-4	
t_{vd1}	$\overline{IOx}$ to BALE valid delay	44 ns		9-4	
t_{vd5}	SA19–SA0 to $\overline{IOCS16}$ valid delay		91 ns	9-4	
t_{vd6}	SA19–SA0, $\overline{SBHE}$ to $\overline{ZEROWS}$ valid delay		200 ns	9-4	
t_{vd6b}	$\overline{IOW}$ to $\overline{ZEROWS}$ valid delay		80 ns	9-4	
t_{vd7}	$\overline{IOx}$ to IOCHRD valid delay		366 ns	9-4	
t_{vd8}	$\overline{IOR}$ to SD15–SD0 valid delay		500 ns	9-4	
t_{pw1}	BALE pulse width	50 ns		9-4	
t_{pw4a}	$\overline{IOx}$ active pulse width	160 ns		9-4	
t_{pw4a}	$\overline{IOx}$ inactive pulse width	163 ns		9-4	
t_{pw7}	IOCHRDY inactive pulse width	120 ns		9-4	
t_{fd8}	$\overline{IOR}$ to SD15–SD0 float delay		41 ns	9-4	
t_{fd9}	$\overline{IOW}$ to SD15–SD0 float delay		105 ns	9-4	

Note:
Measurements are taken with no load.

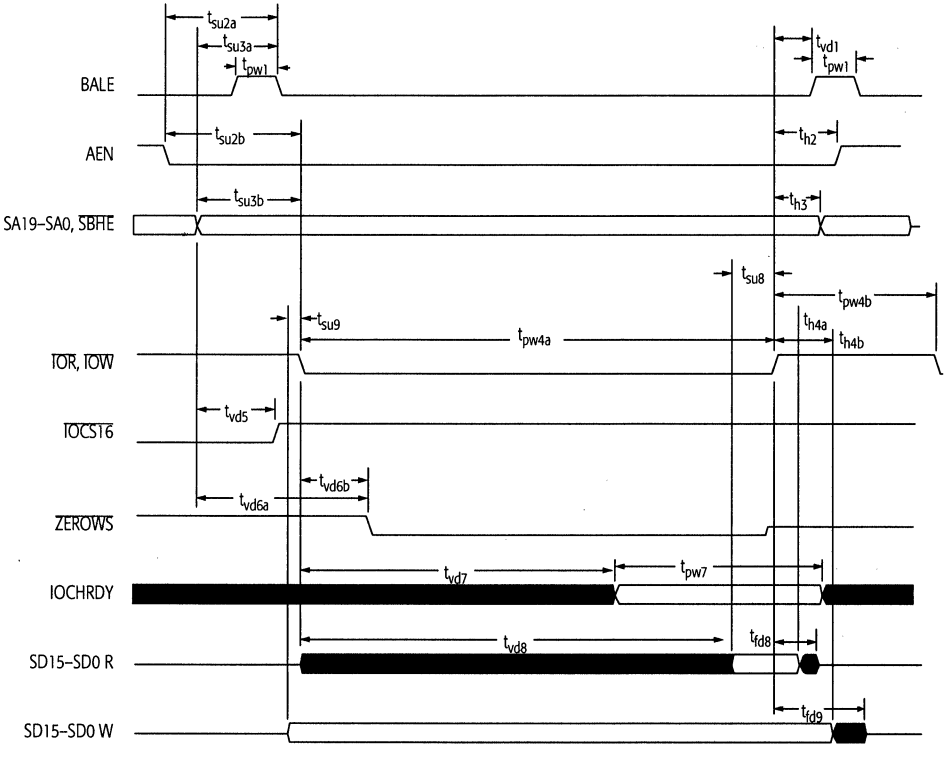


Figure 9-4. ISA 8-Bit Slave Interface Timing

Table 9-9. ISA 16-Bit Slave Interface Timing

Symbol	Parameter Description	Preliminary Data		Figure	Comments
		Min	Max		
t_{su2a}	AEN setup to BALE	150 ns		9-5	
t_{su2b}	AEN setup to $\overline{IOx}$	150 ns		9-5	
t_{su3a}	SA19–SA0 setup to $\overline{IOx}$	34 ns		9-5	
t_{su3b}	SA19–SA0 setup to BALE	37 ns		9-5	
t_{su7}	SD15–SD0 setup to $\overline{IOR}$	24 ns		9-5	
t_{su8}	SD15–SD0 setup to $\overline{IOW}$	-40 ns		9-5	
t_{h2}	$\overline{IOx}$ to AEN hold	26 ns		9-5	
t_{h3}	$\overline{IOx}$ to SA19–SA0 hold	41 ns		9-5	
t_{h5}	SA19–SA0 to $\overline{IOCS16}$ hold	0 ns		9-5	
t_{h7}	$\overline{IOR}$ to SD15–SD0 hold	0 ns		9-5	
t_{h8}	$\overline{IOW}$ to SD15–SD0 hold	45 ns		9-5	
t_{vd1}	$\overline{IOx}$ to BALE valid delay	44 ns		9-5	
t_{vd5a}	$\overline{IOx}$ to $\overline{IOCS16}$ valid delay		16 ns	9-5	
t_{vd5b}	SA19–SA0 to $\overline{IOCS16}$ valid delay		35 ns	9-5	
t_{vd6}	$\overline{IOx}$ to IOCHRD valid delay		78 ns	9-5	
t_{vd7}	$\overline{IOR}$ to SD15–8 valid delay	1.5 ns	8.5 ns	9-5	
t_{pw1}	BALE pulse width	50 ns		9-5	
t_{pw4a}	$\overline{IOx}$ active pulse width	160 ns		9-5	
t_{pw4b}	$\overline{IOx}$ inactive pulse width	163 ns		9-5	
t_{pw6}	IOCHRDY inactive pulse width	120 ns		9-5	
t_{fd7}	$\overline{IOR}$ to SD15–SD0 float delay		41 ns	9-5	
t_{fd8}	$\overline{IOW}$ to SD15–SD0 float delay		105 ns	9-5	

Note:
Measurements are taken with no load.

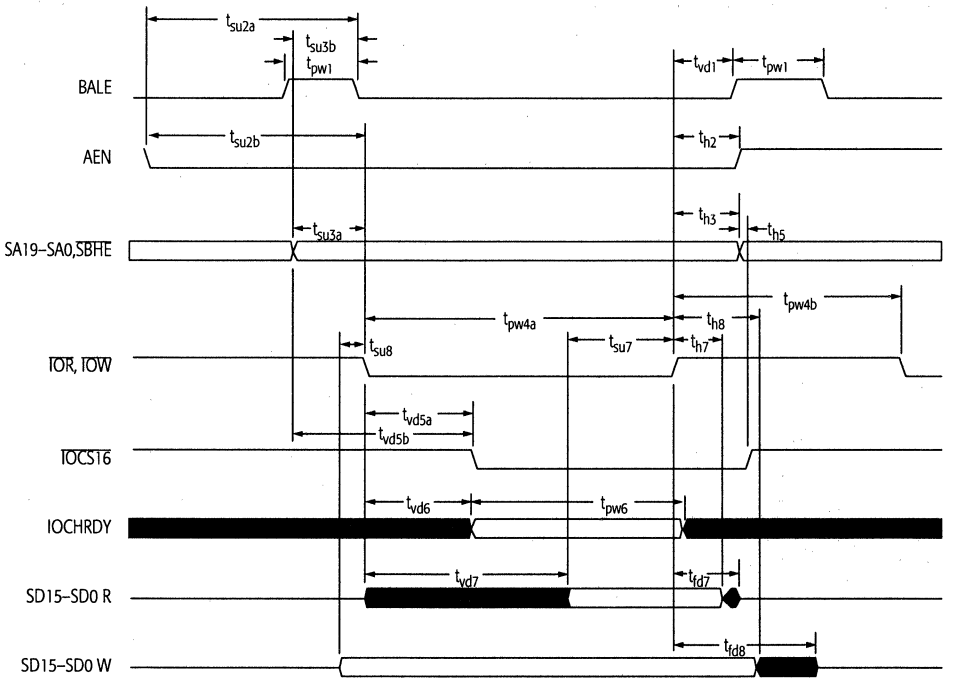


Figure 9-5. ISA 16-Bit Slave Interface Timing

Table 9-10. ISA Master-to-PCI Access Timing

Symbol	Parameter Description	Preliminary Data		Figure	Comments
		Min	Max		
t_{su2}	$\overline{LA23}$–$\overline{LA17}$ setup to $\overline{MEMx}$	23 ns		9-6	
t_{su3}	SA19–SA0 setup to $\overline{MEMx}$	23 ns		9-6	
t_{su7}	SD15–SD0 setup to $\overline{MEMR}$	ns		9-6	
t_{su8}	SD15–SD0 setup to $\overline{MEMW}$	-54 ns		9-6	
t_{h2}	$\overline{MEMx}$ to $\overline{LA23}$–$\overline{LA17}$ hold	ns		9-6	
t_{h3}	$\overline{MEMx}$ to SA19–SA0 hold	30 ns		9-6	
t_{h7}	$\overline{MEMR}$ to SD15–SD0 hold time	0 ns		9-6	
t_{h8}	$\overline{MEMW}$ to SD15–SD0 hold Time	14 ns		9-6	
t_{vd5}	LA23–LA17 to $\overline{MEMCS16}$ valid delay		31 ns	9-6	
t_{vd6}	$\overline{MEMx}$ to IOCHRDY valid delay		85 ns	9-6	
t_{vd7}	IOCHRDY to SD15–SD0 valid delay		69 ns	9-6	
t_{pw4a}	$\overline{MEMx}$ active pulse width	214 ns		9-6	
t_{pw4b}	$\overline{MEMx}$ inactive pulse width	92 ns		9-6	
t_{pw5}	IOCHRDY inactive pulse width	120 ns		9-6	
t_{fd7}	$\overline{MEMR}$ to SD15–SD8 float delay		55 ns	9-6	
t_{fd8}	$\overline{MEMW}$ to SD15–SD8 float delay		ns	9-6	

Note:
Measurements are taken with no load.

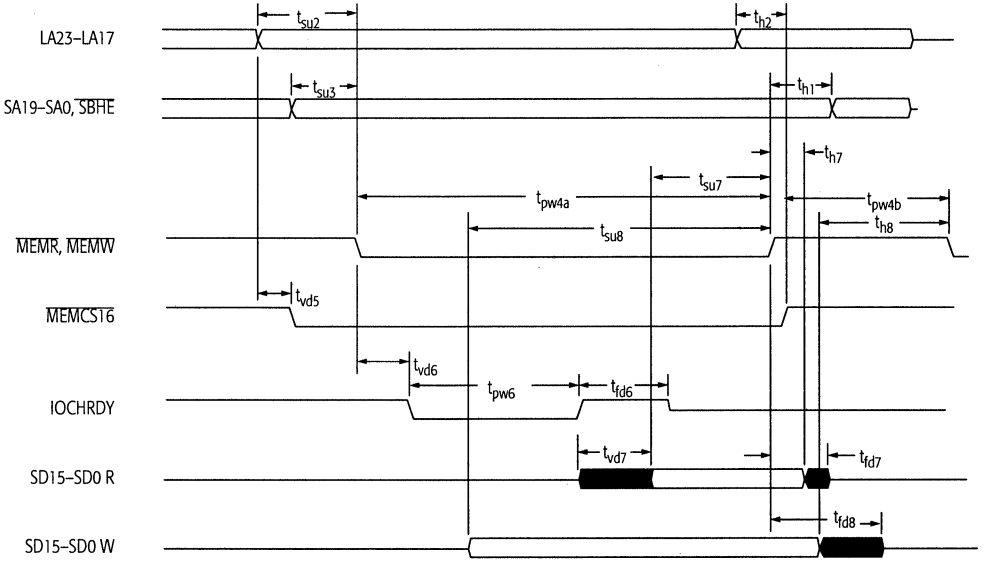


Figure 9-6. ISA Master-to-PCI Access Timing

Table 9-11. Other ISA Master Timing

Symbol	Parameter Description	Preliminary Data		Figure	Comments
		Min	Max		
t_{vd1}	DREQ to $\overline{DACK}$ valid delay	240 ns		9-7	
t_{vd2}	$\overline{DACK}$ to Address, Data and Control valid delay	71 ns		9-7	
t_{fd}	$\overline{DACK}$ to Address, Data and Control float delay	0 ns		9-7	

Note:
Measurements are taken with no load.

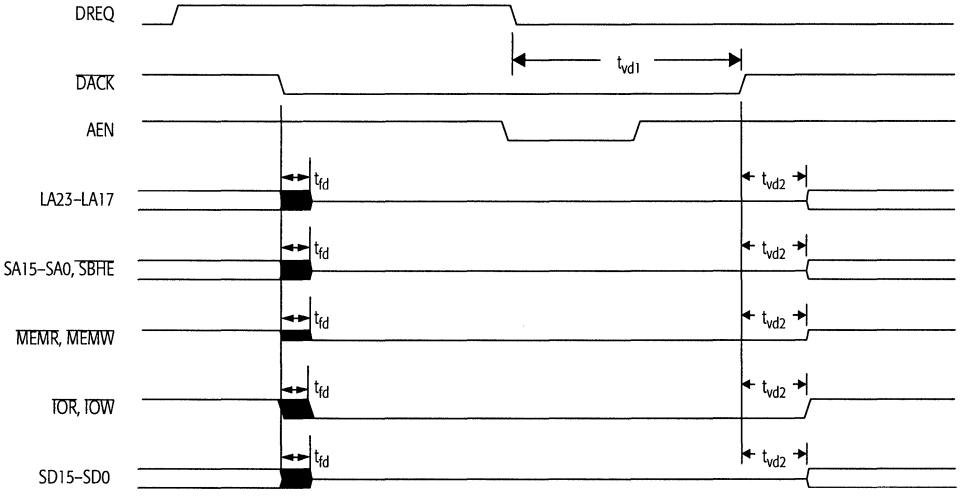


Figure 9-7. Other ISA Master Timing

9.5 DMA Interface Timing

Table 9-12. DMA Read Cycle Timing

Symbol	Parameter Description	Preliminary Data		Figure	Comments
		Min	Max		
t_{su2}	AEN setup to $\overline{IOW}$	111 ns		9-8	
t_{su3}	$\overline{DACK}$ setup to $\overline{IOW}$	312 ns		9-8	
t_{su4}	SA19–SA0,LA23–LA17 setup to $\overline{MEMR}$	99 ns		9-8	
t_{su6}	$\overline{MEMR}$ setup to $\overline{IOW}$	-26 ns		9-8	
t_{su10}	SD15–SD0 setup to $\overline{IOW}$	225		9-8	
t_{su11}	TC setup to $\overline{IOW}$	511		9-8	
t_{h2}	$\overline{IOW}$ to AEN hold	41 ns		9-8	
t_{h3}	$\overline{IOW}$ to $\overline{DACK}$ hold	155 ns		9-8	
t_{h4}	$\overline{MEMR}$ to SA19–SA0, LA23–LA17 hold	51 ns		9-8	
t_{h6}	$\overline{IOW}$ to $\overline{MEMR}$ hold	40 ns		9-8	
t_{h9}	IOCHRDY to $\overline{MEMR}$ hold	120 ns		9-8	
t_{h10}	$\overline{IOW}$ to SD15–SD0 hold	36 ns		9-8	
t_{h11}	$\overline{IOW}$ to TC hold	71 ns		9-8	
t_{vd1}	$\overline{IOW}$ to DRQ inactive valid delay		315 ns	9-8	
t_{vd7}	$\overline{MEMR}$ to $\overline{SMEMR}$ valid delay		15 ns	9-8	
t_{vd9}	$\overline{MEMR}$ to IOCDRY valid delay		315 ns	9-8	
t_{pw6a}	$\overline{MEMR}$ active pulse width	495 ns		9-8	
t_{pw6b}	$\overline{MEMR}$ inactive pulse width	465 ns		9-8	
t_{pw8a}	$\overline{IOW}$ active pulse width	495 ns		9-8	
t_{pw8b}	$\overline{IOW}$ inactive pulse width	465 ns		9-8	
t_{pw9}	IOCHRDY inactive pulse width	125 ns		9-8	
t_{pw11}	TC active pulse width	700 ns		9-8	

Note:
Measurements are taken with no load.

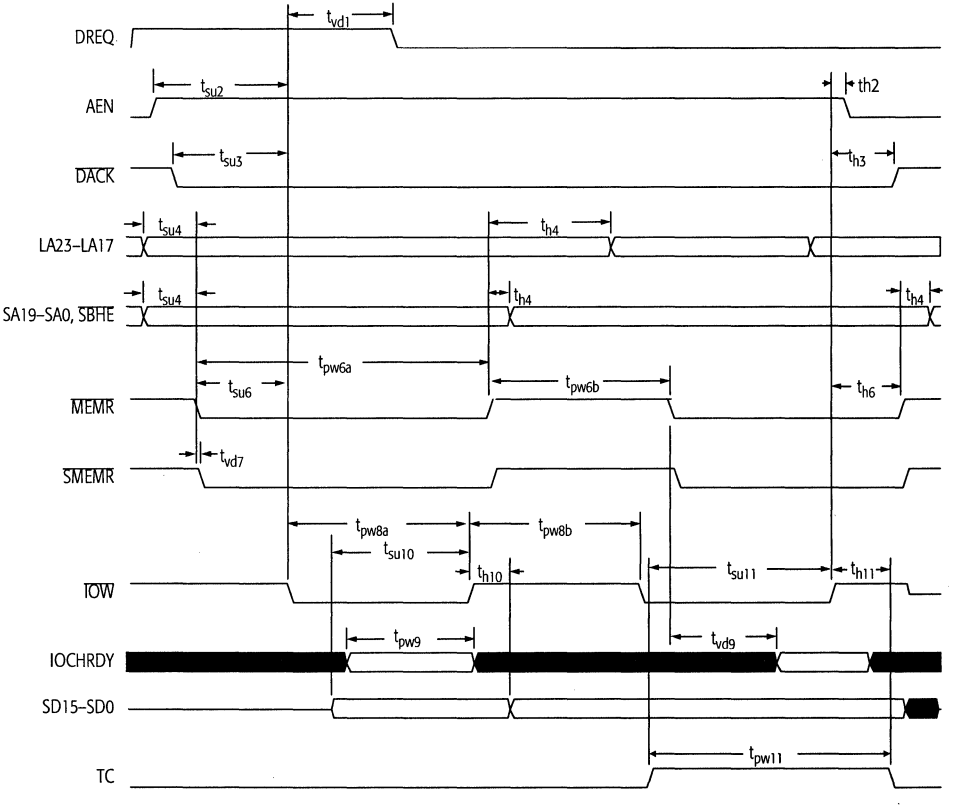


Figure 9-8. DMA Read Cycle Timing

Table 9-13. DMA Write Cycle Timing

Symbol	Parameter Description	Preliminary Data		Figure	Comments
		Min	Max		
t_{su2}	AEN setup to $\overline{IOR}$	111 ns		9-9	
t_{su3}	$\overline{DACK}$ Setup to $\overline{IOR}$	73 ns		9-9	
t_{su4}	SA19–SA0,LA23–LA17 setup to $\overline{MEMW}$	99 ns		9-9	
t_{su10}	SD15–SD0 setup to $\overline{IOR}$			9-9	
t_{su11}	TC setup to $\overline{IOR}$	511 ns		9-9	
t_{h2}	$\overline{IOR}$ to AEN hold	41 ns		9-9	
t_{h3}	$\overline{IOR}$ to $\overline{DACK}$ hold	100 ns		9-9	
t_{h4}	$\overline{MEMW}$ to SA19–SA0, LA23–LA17 hold	51 ns		9-9	
t_{h6}	$\overline{IOR}$ to $\overline{MEMW}$ hold	40 ns		9-9	
t_{h10}	$\overline{IOR}$ to SD15–SD0 hold	0 ns		9-9	
t_{h11}	$\overline{IOR}$ to TC hold	71 ns		9-9	
t_{vd1}	$\overline{IOR}$ to DRQ valid delay		558 ns	9-9	
t_{vd6}	$\overline{IOR}$ to $\overline{MEMW}$ valid delay	230 ns		9-9	
t_{vd7}	$\overline{MEMW}$ to $\overline{SMEMW}$ valid delay		15 ns	9-9	
t_{vd8}	$\overline{IOR}$ to SD15–SD0 valid delay		237 ns	9-9	
t_{vd9}	$\overline{MEMW}$ to IOCDRY valid delay		315 ns	9-9	
t_{pw6a}	$\overline{MEMW}$ active pulse width	495 ns		9-9	
t_{pw6b}	$\overline{MEMW}$ inactive pulse width	465 ns		9-9	
t_{pw8a}	$\overline{IOR}$ active pulse width	760 ns		9-9	
t_{pw8b}	$\overline{IOR}$ inactive pulse width	160 ns		9-9	
t_{pw9}	IOCHRDY inactive pulse width	125 ns		9-9	
t_{pw11}	TC active pulse width	700 ns		9-9	

Note:
Measurements are taken with no load.

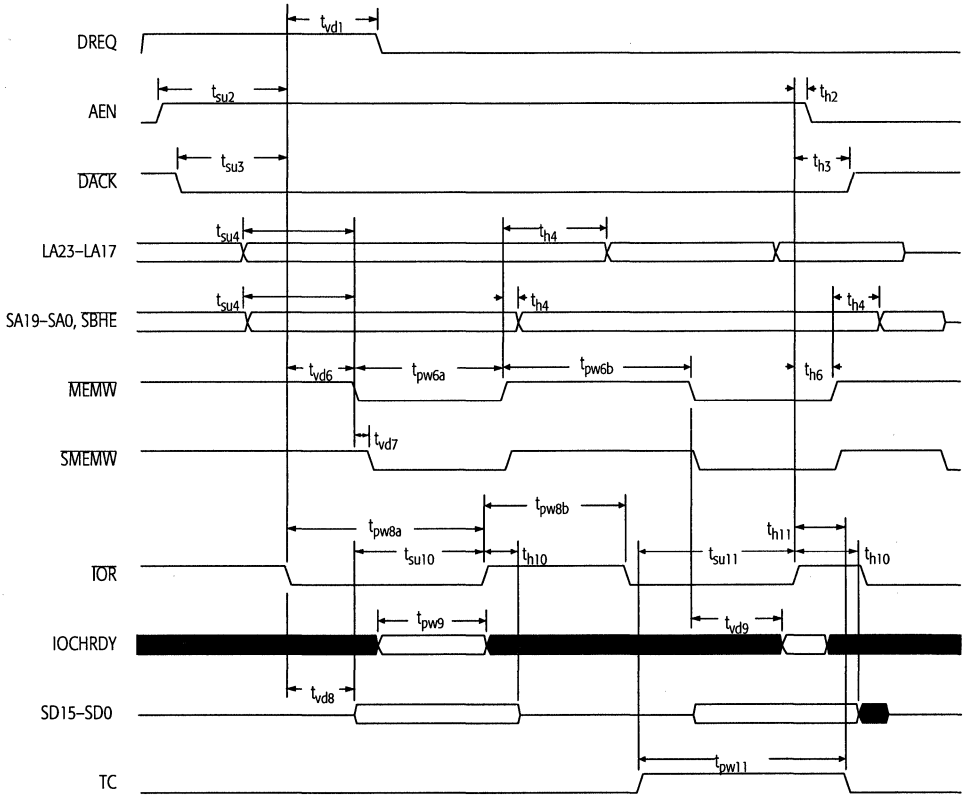


Figure 9-9. DMA Write Cycle Timing

Table 9-14. Type F DMA Interface Timing

Symbol	Parameter Description	Preliminary Data		Figure	Comments
		Min	Max		
t_{h1a}	$\overline{IOW}$ to DREQ hold	82 ns		9-9	
t_{h1b}	$\overline{IOR}$ to DREQ hold	82 ns		9-9	
t_{h3a}	$\overline{IOW}$ to DACK hold	30 ns		9-9	
t_{h3b}	$\overline{IOR}$ to DACK hold	30 ns		9-9	
t_{h10}	$\overline{IOW}$ to TC hold	0 ns		9-9	
t_{vd4a}	AEN to $\overline{IOW}$ valid delay	111 ns		9-9	
t_{vd4b}	$\overline{DACK}$ to $\overline{IOW}$ valid delay	77 ns		9-9	
t_{vd7a}	AEN to $\overline{IOR}$ valid delay	111 ns		9-9	
t_{vd7b}	$\overline{DACK}$ to $\overline{IOR}$ valid delay	77 ns		9-9	
t_{pw4a}	$\overline{IOW}$ active pulse width	110 ns		9-9	
t_{pw4b}	$\overline{IOW}$ inactive pulse width	115 ns		9-9	
t_{pw7a}	$\overline{IOR}$ active pulse width	110 ns		9-9	
t_{pw7b}	$\overline{IOR}$ inactive pulse width	115 ns		9-9	
t_{fd7}	$\overline{IOR}$ to SD15–SD8 float delay		61 ns	9-9	

Note:
Measurements are taken with no load.

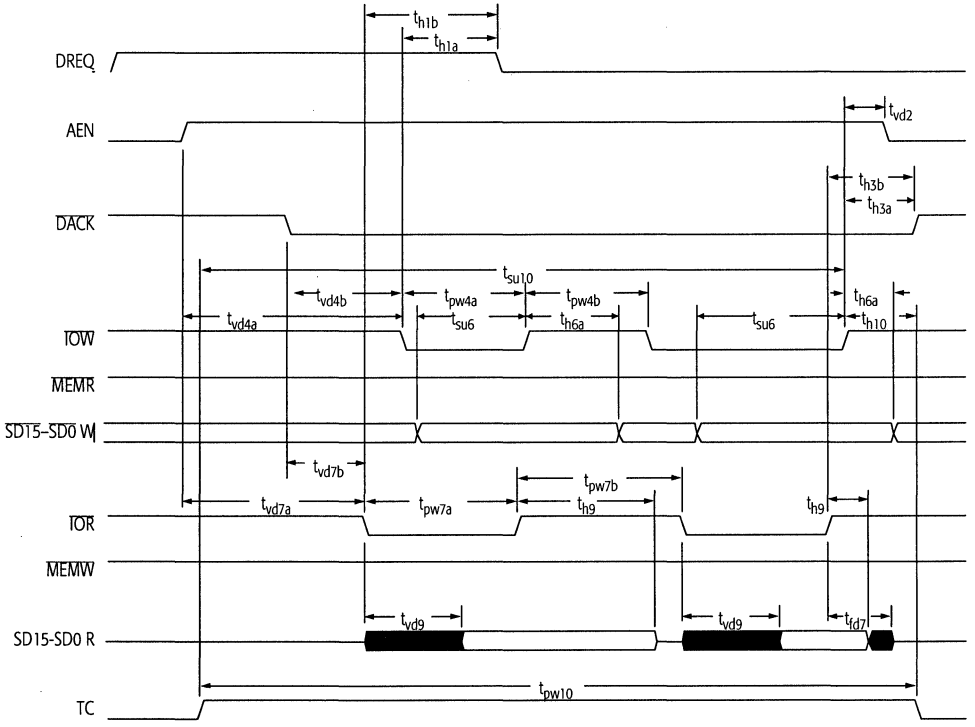


Figure 9-10. Type F DMA Interface Timing

9.6 X-Bus Interface Timing

Table 9-15. X-BUS Interface Timing

Symbol	Parameter Description	Preliminary Data		Figure	Comments
		Min	Max		
t_{su7}	$\overline{XOET}$ setup to $\overline{XDIR}$	-2 ns	8 ns	9-11	
t_{su8}	SD15–SD0R setup to $\overline{MEMR}$, $\overline{IOR}$	24 ns		9-11	
t_{su10}	SD15–SD0W setup to $\overline{MEMW}$, $\overline{IOW}$	24 ns		9-11	
t_{su11}	$\overline{XOE}$ setup to $\overline{XDIRT}$	0 ns		9-11	
t_{h8}	$\overline{MEMR}$, $\overline{IOR}$ to SD15–SD0R hold	0 ns		9-11	
t_{h10}	$\overline{MEMW}$, $\overline{IOW}$ to SD15–SD0W hold			9-11	
t_{vd4}	LA23–LA17, SA19–SA0 to $\overline{PCCS}$ valid delay		35 ns	9-11	
t_{vd6}	$\overline{MEMR}$, $\overline{IOR}$ to $\overline{XOET}$		29 ns	9-11	
t_{vd7}	$\overline{MEMR}$, $\overline{IOR}$ to $\overline{XDIR}$ valid delay		25 ns	9-11	
t_{vd8}	$\overline{MEMR}$, $\overline{IOR}$, to SD15–SD0R valid delay			9-11	
t_{vd10}	$\overline{MEMW}$, $\overline{IOW}$ to SD15–SD0W valid delay			9-11	
t_{vd11}	$\overline{MEMW}$, $\overline{IOW}$ to XOE		29 ns	9-11	
t_{vd12}	$\overline{MEMW}$, $\overline{IOW}$ to $\overline{XDIRT}$ valid delay		25 ns	9-11	

Note:
Measurements are taken with no load.

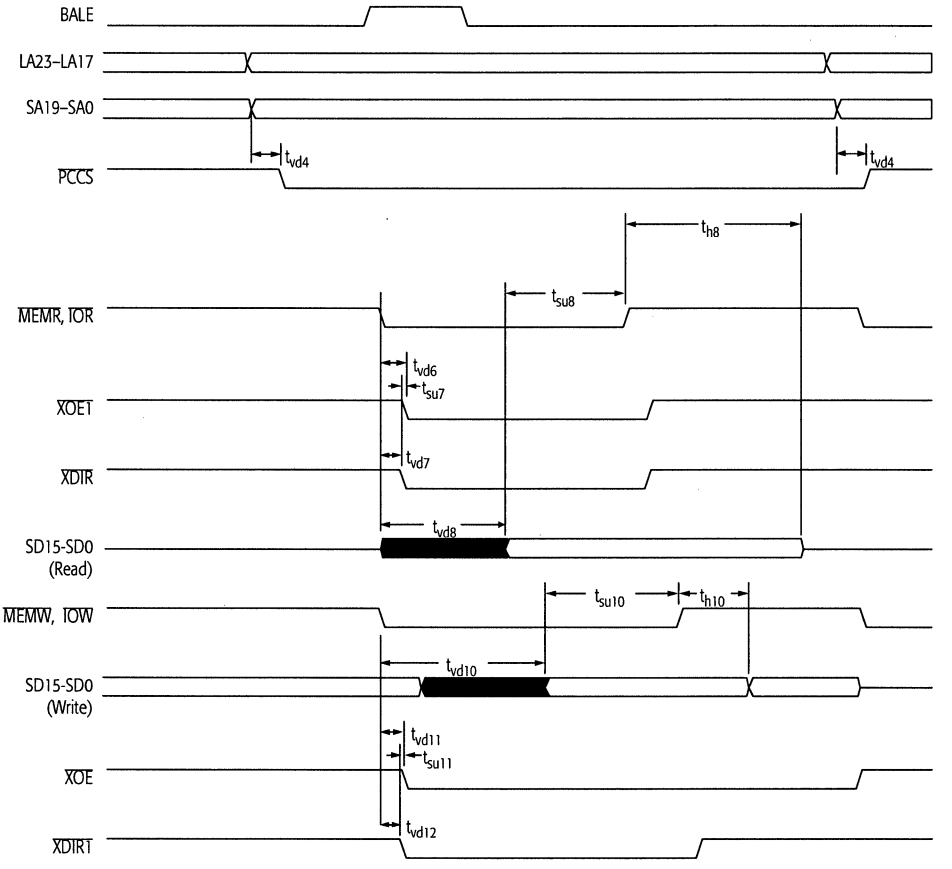


Figure 9-11. X-Bus Interface Timing

9.7 EIDE Interface

Table 9-16. EIDE PIO

Symbol	Description		Mode 0	Mode 1	Mode 2	Mode 3	Mode 4
t_{cyc}	Cycle time (DIOW/R to DIOW/R)	min	600	383	240	180	120
t_{su1}	IDE address setup DIOW/R	max	70	50	30	30	25
t_{pw1}	8-bit DIOW/R pulse width	min	290	290	290	80	70
t_{pw1}	16-bit DIOW/R pulse width	min	165	125	100	80	70
t_{rec}	DIOW/R recovery time	min	–	–	–	70	25
t_{su2}	Write data setup	min	60	45	30	30	20
t_{h2}	Write data hold	min	30	20	15	10	10
t_{su3}	Read data setup from drive	min	50	350	20	20	20
t_{h3}	Read data hold from drive	min	5	5	5	5	5
t_{vd1}	PCLK to DD15–DD0 valid delay	min	2	2	2	2	2
t_{vd1}	PCLK to DD15–DD0 valid delay	max	20	20	20	20	20
t_{su4}	DD15–DD0 to PCLK setup	min	10	10	10	10	10
t_{h4}	PCLK to DD15–DD0 hold	min	4	4	4	4	4
t_{vd2}	PCLK to DA2–DA0 valid delay	min	2	2	2	2	2
t_{vd2}	PCLK to DA2–DA0 valid delay	max	20	20	20	20	20
t_{vd3}	PCLKC to MASTER, SOE, DIOx, DCSxx valid delay	min	2	2	2	2	2
t_{vd3}	PCLKC to MASTER, SOE, DIOx, DCSxx valid delay	max	20	20	20	20	20
t_{su5}	IORDY setup	min	20	20	20	20	20
t_{h5}	PCLK to DRDYx hold	min	5	5	5	5	5

Note:
All timings are in nanoseconds.

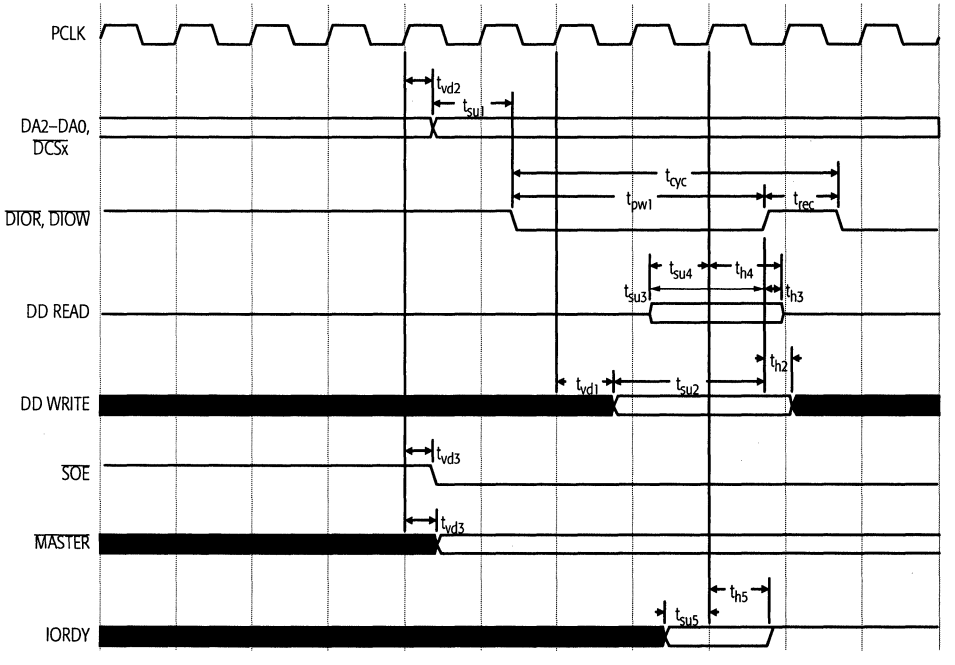


Figure 9-12. EIDE PIO

Table 9-17. EIDE DMA

Symbol	Description		Single-Word			Multi-Word		
			Mode 0	Mode 1	Mode 2	Mode 0	Mode 1	Mode 2
t_{cyc}	Cycle time (DMACK to DMACK)	min	960	480	240	480	150	120
t_{vd1}	DMACK to DMARQ valid delay	max	200	100	80			35
t_{pw1}	DIOW/R pulse width	min	480	240	120	215	80	70
t_{pw2}	DIOR deasserted pulse width	min	–	–	–	50	50	25
t_{pw3}	DIOW deasserted pulse width	min	–	–	–	215	50	25
t_{h1}	DIOR data hold time	min	5	5	5	5	5	5
t_{su1}	DIOW data setup time	min	250	100	35	100	30	20
t_{su2}	DMACK to DIOW/R setup	min	0	0	0	0	0	0
t_{h2}	DIOW data hold	min	5	5	5	5	5	5
t_{su2}	DMACK to DIOW/R	min	0	0	0	0	0	0
t_{h3}	DIOW/R to DMACK hold	min	0	0	0	20	5	5
t_{vd3}	DIOR to DMARQ valid delay	min				120	40	35
t_{vd4}	DIOW to DMARQ valid delay					40	40	35
t_{vd5}	PCLK to DD15–DD0 valid delay	min	2	2	2	2	2	2
t_{vd5}	PCLK to DD15–DD0 valid delay	max	20	20	20	20	20	20
t_{su4}	DD15–DD0 to PCLK setup	min	10	10	10	10	10	10
t_{h4}	PCLK to DD15–DD0 hold	min	4	4	4	4	4	4
t_{vd7}	PCLKC to MASTER, SOE, DIOx, DCSxx valid delay	min	2	2	2	2	2	2
t_{vd7}	PCLKC to MASTER, SOE, DIOx, DCSxx valid delay	max	20	20	20	20	20	20
t_{su5}	DDRQx to PCLK	min	10	10	10	10	10	10
t_{h5}	PCLK to DDRQx hold	min	2	2	2	2	2	2
t_{vd8}	DDACKx to PCLK valid delay	min	2	2	2	2	2	2
t_{vd8}	DDACKx to PCLK valid delay	min	20	20	20	20	20	20

Note:
All timings are in nanoseconds.

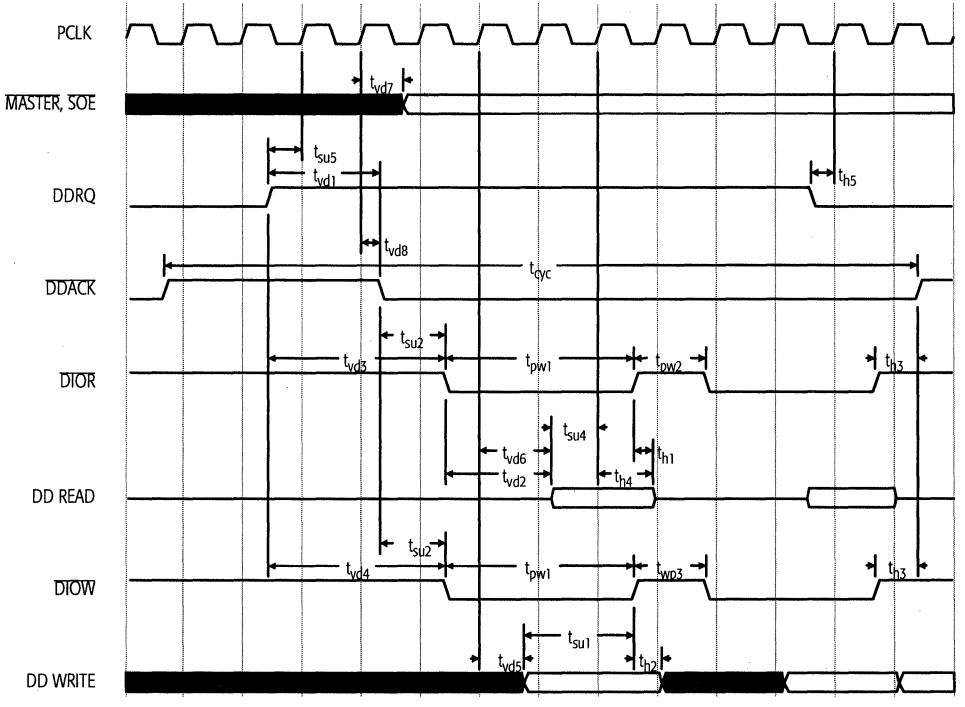


Figure 9-13. EIDE DMA

9.8 Ultra DMA-33 IDE Bus Interface Timing

Table 9-18. UltraDMA-33 IDE Bus Interface Timing

Symbol	Parameter Description	Preliminary Data		Figure	Comments
		Min	Max		
t_{env1}	Envelope time for read initial	20 ns	70 ns	5-20	
t_{ds1}	Data setup time for read initial	34 ns		5-20	
t_{dh1}	Data hold time for read initial (rise)	6 ns		5-20	
t_{env2}	Envelope time for write initial (rise)	20 ns	70 ns	5-24	
t_{dvs2}	Data setup time for write initial (fall)	34 ns		5-24	
t_{dvh2}	Data hold time for write initial (fall)	6 ns		5-24	
t_{dvs2}	Data setup time for write initial	34 ns		5-24	
t_{dvh2}	Data hold time for write initial	6 ns		5-24	
t_{rfs}	READY to final STROBE time		50 ns	5-21	
t_{rp}	READY to Pause time	100 ns		5-21	
t_{li4}	Limited interlock time (to STOP)	0 ns	150 ns	5-22	
t_{li4}	Limited interlock time (to Host DMARDY)	0 ns	150 ns	5-22	
t_{za4}	Delay time required for output drives turning on	20 ns		5-22	
t_{dvs4}	Data setup time for read terminating	34 ns		5-22	
t_{dvh4}	Data hold time for read terminating	6 ns		5-22	
t_{li5}	Limited interlock time (to STOP)	0 ns	150 ns	5-25	
t_{li5}	Limited interlock time (to Host STROBE)	0 ns	150 ns	5-25	
t_{mli5}	Limited interlock time with minimum	20 ns		5-25	
t_{dvs5}	Data setup time for write terminating	34 ns		5-25	
t_{dvh5}	Data hold time for write terminating	6 ns		5-25	
t_{mli6}	Limited interlock time with minimum	20 ns		5-23	
t_{za6}	Delay time required for output drives turning on	34 ns		5-25	
t_{li5}	Limited interlock time	0 ns	150 ns	5-27	
t_2	Delay time of PCLK to $\overline{DCS3}$, $\overline{DCS1}$	2 ns	20 ns	5-27	
t_3	Delay time of PCLK to DA2-DA0	2 ns	20 ns	5-27	
t_4	Delay time of PCLK to $\overline{DIOW}$	2 ns	20 ns	5-27	
t_5	Delay time of PCLK to $\overline{DIOR}$	2 ns	20 ns	5-27	
t_{wds}	Data setup time during PIO and DMA write	30 ns		5-27	
t_{wdh}	Data hold time during PIO and DMA write	20 ns		5-27	
t_{rds}	Data setup time during PIO and DMA read	30 ns		5-27	
t_{rdh}	Data hold time during PIO and DMA read	20 ns		5-27	

10 IBIS Models

All of the AMD-645 Peripheral Bus Controller's inputs, outputs, and bidirectional buffers are implemented using a 3.3-V buffer design. In addition, a subset of the controller's I/O buffers includes a second, higher drive strength option.

AMD has developed several I/O buffer models that represent the characteristics of each of the possible drive strength configurations supported by the AMD-645 Peripheral Bus Controller.

AMD developed the models to allow system designers to perform analog simulations of AMD-645 Peripheral Bus Controller signals that interface with the rest of the system. Analog simulations are used to determine a signal's time of flight from source to destination and to ensure that the system's signal quality requirements are met. Signal quality measurements include overshoot, undershoot, slope reversal, and ringing.

10.1 I/O Buffer Model

AMD provides models of the AMD-645 Peripheral Bus Controller I/O buffers for system designers to use in board-level simulations. These I/O buffer models conform to the *I/O Buffer Information Specification (IBIS), Version 2.1.*

Each I/O model contains voltage versus current (V/I) and voltage versus time (V/T) data tables for accurate modeling of I/O buffer behavior.

The following list summarizes the properties of each I/O buffer model:

- All data tables contain minimum, typical, and maximum values to allow for worst-case, typical, and best-case simulations, respectively.
- The pullup, pulldown, power clamp, and ground clamp device V/I tables contain enough data points to accurately

represent the nonlinear nature of the V/I curves. In addition, the voltage ranges provided in these tables extend beyond the normal operating range of the AMD-645 Peripheral Bus Controller for those simulators that yield more accurate results based on this wider range.

- Rising and falling ramp rates are specified.
- The min/typ/max V_{CC3} operating range is specified as 3.135 V, 3.3 V, and 3.465 V, respectively.
- V_{il} = 0.8 V, V_{ih} = 2.0 V, and V_{meas} = 1.5 V.
- The R/L/C of the package is modeled.
- The capacitance of the silicon die is modeled.
- The model assumes 0 capacitance, resistance, inductance, and voltage in the test load.

10.2 I/O Model Application Note

For the AMD-645 Peripheral Bus Controller I/O Buffer IBIS Models and their application, refer to the *AMD-645 Peripheral Bus Controller I/O Model (IBIS) Application Note*, order# 21340.

The model is available at http://www.amd.com

10.3 I/O Buffer AC and DC Characteristics

Refer to Section 9 for the AMD-645 Peripheral Bus Controller AC timing specifications.

Refer to Section 8 for the AMD-645 Peripheral Bus Controller DC specifications.

10.4 References

Ease System Simulation With IBIS Device Models by Syed Huq, *Electronics Design*, Dec 2, 1996

IBIS 2.1 Specification at http://vhdl.org/

IBIS Forum I/O Buffer Modeling Cook Book

11 Pin Designations

11.1 Pin Designation Table

The 208 pins of the AMD-645 Peripheral Bus Controller are listed in the following tables, grouped according to their functions.

Table 11-1. Functional Grouping

EIDE Interface		USB Interface		Keyboard Interface		Internal RTC	
Pin No.	**Pin Name**	**Pin No.**	**Pin Name**	**Pin No.**	**Pin Name**	**Pin No.**	**Pin Name**
50	$\overline{\text{DIORA}}$	95	USBDATA0+	108	KBCK/KA20G	104	RTCX1/$\overline{\text{IRQ8}}$
	HDMARDYA/	96	USBDATA0-	109	KBDT/KBRC	105	RTCX2/$\overline{\text{RTCCS}}$
	HSTROBEA	97	USBDATA1+	110	MSCK/IRQ1	102	VBAT
51	DIOWA/	98	USBDATA1-	111	MSDT/IRQ12		
	STOPA	99	USBCLK	147	A20M		
54	DIORB/			106	KEYLOCK/MIRQ1		
	HDAMRDYB/						
	HSTROBEB						
55	DIOWB/						
	STOPB						
49	DRDYA/						
	DDMARDYA/						
	DSTROBEA						
89	DRDYB/						
	DDMARDYB						
	DSTROBEB						
56	SOE						
45	DDRQA						
46	DDRQB						
47	DDACKA						
48	DDACKB						

Table 11-1. Functional Grouping (continued)

CPU Interface		Reset & Clock		PCI Bus Interface		ISA Bus Control			
Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name
142	CPURST	138	PWRGD	2	PCLK	20	SA15/DD15	5	IOCHCK
145	INTR	3	PCIRST	181	FRAME	21	SA14/DD14	8	IOCHRDY
146	NMI	4	RSTDRV	204	AD31	22	SA13/DD13	29	REFRESH
143	INIT	14	BCLK	203	AD30	23	SA12/DD12	15	AEN
148	STPCLK	6	OSC	202	AD29	24	SA11/DD11	32	TC
149	SMI			201	AD28	25	SA10/DD10	128	IRQ15
141	FERR			200	AD27	27	SA9/DD9	129	IRQ14
139	IGENN			199	AD26	28	SA8/DD8	127	IRQ11
				196	AD25	36	SA7/DD7	126	IRQ10
				195	AD24	37	SA6/DD6	61	IRQ9
				192	AD23	38	SA5/DD5	71	IRQ7
				191	AD22	40	SA4/DD4	72	IRQ6
				190	AD21	41	SA3/DD3	73	IRQ5
				189	AD20	42	SA2/DD2	74	IRQ4
				187	AD19	43	SA1/DD1	75	IRQ3
				186	AD18	44	SA0/DD0	132	DRQ7
				185	AD17	19	SA16	130	DRQ6
				183	AD16	63	LA23/DCS3B	57	DRQ5
				172	AD15	64	LA22/DCSTB	30	DRQ3
				170	AD14	65	LA21/DCS3A	7	DRQ2
				169	AD13	66	LA20/DCSTA	16	DRQ1
				168	AD12	67	LA19/DA2	59	DRQ0
				167	AD11	69	LA18/DA1	133	DACK7
				165	AD10	70	LA17DA0	131	DACK6
				164	AD9	86	SD15/	58	DACK5
				163	AD8		GPI15/	31	DACK3
				161	AD7	85	GPO15	33	DACK2
				160	AD6		SD14/	18	DACK1
				159	AD5	83	GPI14/	60	DACK0
				158	AD4		GPO14	134	SPKR
				155	AD3	82	SD13/		
				154	AD2		GPI13/		
				153	AD1	81	GPO13		
				152	AD0		SD12/		
				194	C/BE3	80	GPI12/		
				182	C/BE2		GPO12		
				173	C/BE1	78	SD11/		
				162	C/BE0		GPI11/		
				180	IRDY	77	GPO11		
				179	TRDY		SD10/		
				176	STOP	62	GPI10/		
				178	DEVSEL	12	GPO10		
				174	PAR	11	SD9/		
				175	SERR	123	GPI9/GPO9		
				193	IDSEL	124	SD8/		
				1	PIRQA	10	GPI8/GPO8		
				207	PIRQB	9	SBHE		
				206	PIRQC	35	IOR		
				205	PIRQD	125	IOW		
				151	PREQ	76	MEMR		
				150	PGNT		MEMW		
							SMEMR		
							SMEMW		
							BALE		
							IOCS16		
							MEMCS16		

Table 11-1. Functional Grouping (continued)

Onboard Plug-N-Play		XD Interface		Power & Ground		Power Management/ General Purpose I/O	
Pin No.	**Pin Name**	**Pin No.**	**Pin Name**	**Pin No.**	**Pin Name**	**Pin No.**	**Pin Name**
90	MDRQ0/APICCS	122	XD7/EXTSMI7/ GPI7/GPO7	17	VDD5	91	PWRBTN
106	MIRQ1/KEYLOCK			34	VDD5	107	PWRON
137	MIRQ2/MASTER	121	XD6/EXTSMI6/ GPI6/GPO6	53	VDD5	93	RI
				79	VDD5	94	GPIO0/EXTSMI0
		119	XD5/EXTSMI5/ GPI5/GPO5	115	VDD5	87	GPIO11/EXTSMI1 I2CD1
				103	VDD-5VSB		
		118	XD4/EXTSMI4/ GPI4/GPO4	144	VDD3	88	GPIO12/EXTSMI2 I2CD2
				157	VDD_PCI		
		117	XD3/EXTSMI3/ GPI3/GPO3	171	VDD_PCI	92	GPIO13/EXTSMI3 GPI_RE
				184	VDD_PCI		
		116	XD2/EXTSMI2/ GPI2/GPO2	198	VDD_PCI	136	GPIO14/EXTSMI4 GPO_WE
				100	AVDD		
		114	XD1/EXTSMI1/ GPI1/GPO1	101	AGND		
				13	GND		
		113	XD0/EXTSMI0/ GPI0/GPO0	26	GND		
				39	GND		
		112	XDIR	52	GND		
		135	ROMCS/KBCS	68	GND		
				84	GND		
				120	GND		
				140	GND		
				156	GND		
				166	GND		
				177	GND		
				188	GND		
				197	GND		
				208	GND		

11.2 Pin Diagram

Figure 11-1 shows the pin arrangement of the AMD-645 Peripheral Bus Controller.

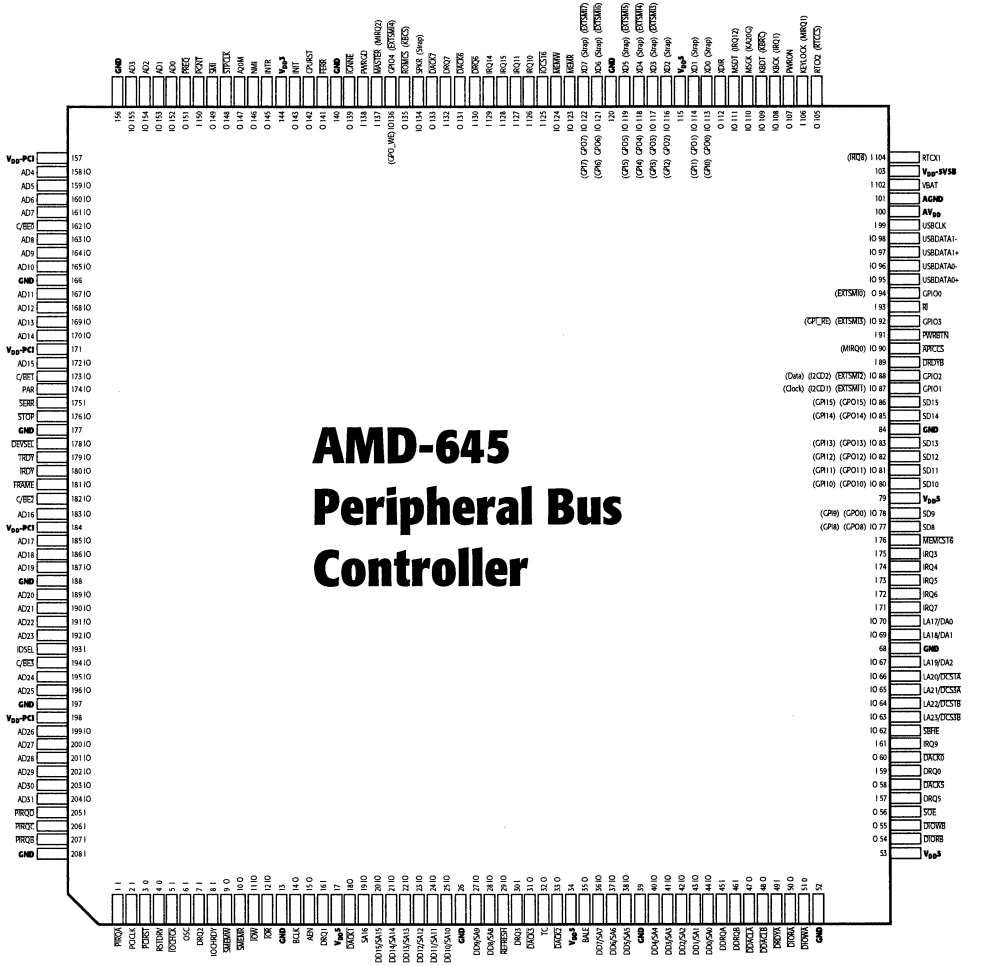


Figure 11-1. AMD-645 Peripheral Bus Controller Pin Diagram

12 Package Specifications

The AMD-645 Peripheral Bus Controller is available as a 208-pin plastic quad flat pack (PQFP). The thermal specifications are as follows:

$\theta_{JA} = 37\ ^{o}C/W$

$\theta_{JC} = 4.7\ ^{o}C/W$

Figure 12-1 is a drawing of the 208-pin PQFP.

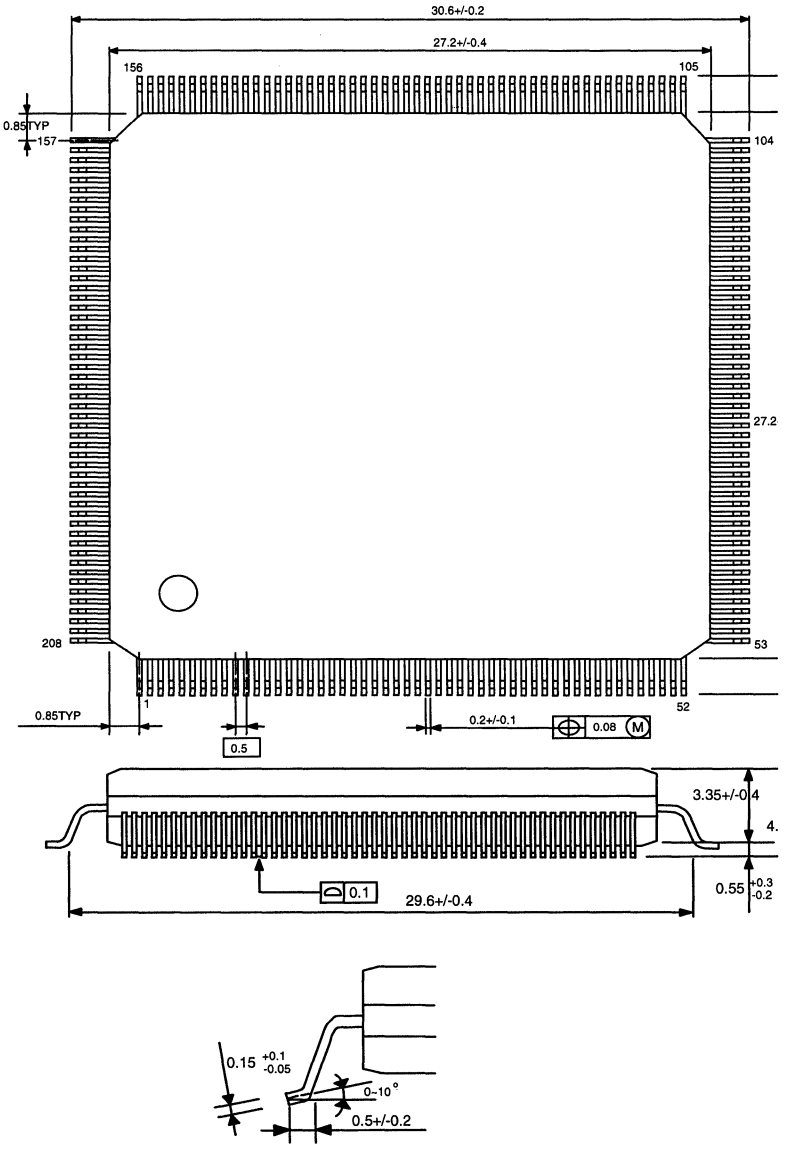


Figure 12-1. 208-Pin Plastic Quad Flat Pack Outline Drawing

Sales Offices

North American

ALABAMA ..... (205) 830-9192
ARIZONA ..... (602) 242-4400
CALIFORNIA,
Calabasas ..... (818) 878-9988
Irvine ..... (714) 450-7500
Sacramento (Roseville) ..... (916) 786-6700
San Diego ..... (619) 560-7030
San Jose ..... (408) 922-0300
CANADA, Ontario,
Kanata ..... (613) 592-0060
Woodbridge ..... (905) 856-3377
COLORADO ..... (303) 741-2900
CONNECTICUT ..... (203) 264-7800
FLORIDA,
Clearwater ..... (813) 530-9971
Ft. Lauderdale ..... (954) 938-9550
Orlando (Longwood) ..... (407) 862-9292
GEORGIA ..... (770) 449-7920
IDAHO ..... (208) 377-0393
ILLINOIS, Chicago (Itasca) ..... (708) 773-4422
KENTUCKY ..... (606) 224-1353
MARYLAND ..... (410) 381-3790
MASSACHUSETTS ..... (617) 273-3970
MINNESOTA ..... (612) 938-0001
NEW JERSEY,
Cherry Hill ..... (609) 662-2900
Parsippany ..... (201) 299-0002
NEW YORK,
Brewster ..... (914) 279-8323
Rochester ..... (716) 425-8050
NORTH CAROLINA,
Charlotte ..... (704) 875-3091
Raleigh ..... (919) 878-8111
OHIO,
Columbus (Westerville) ..... (614) 891-6455
Dayton ..... (513) 439-0268
OREGON ..... (503) 245-0080
PENNSYLVANIA ..... (610) 398-8006
TEXAS,
Austin ..... (512) 346-7830
Dallas ..... (214) 934-9099
Houston ..... (713) 376-8084

International

AUSTRALIA, N Sydney ..... TEL ..... (61) 2 9959-1937
FAX ..... (61) 2 9959-1037
BELGIUM, Antwerpen ..... TEL ..... (03) 248-4300
FAX ..... (03) 248-4642
CHINA,
Beijing ..... TEL ..... (8610) 501-1566
FAX ..... (8610) 465-1291
Shanghai ..... TEL ..... (8621) 6267-8857
TEL ..... (8621) 6267-9883
FAX ..... (8621) 6267-8110
FINLAND, Helsinki ..... TEL ..... (358) 9 881 3117
FAX ..... (358) 9 804 1110
FRANCE, Paris ..... TEL ..... (1) 49-75-1010
FAX ..... (1) 49-75-1013
GERMANY,
Bad Homburg ..... TEL ..... (06172) 92670
FAX ..... (06172) 23195
München ..... TEL ..... (089) 450530
FAX ..... (089) 406490
HONG KONG, Kowloon ..... TEL ..... (852) 2956-0388
FAX ..... (852) 2956-0588
ITALY, Milano ..... TEL ..... (02) 381961
FAX ..... (02) 3810-3458
JAPAN,
Osaka ..... TEL ..... (06) 243-3250
FAX ..... (06) 243-3253
Tokyo ..... TEL ..... (03) 3346-7600
FAX ..... (03) 3346-5197
KOREA, Seoul ..... TEL ..... (82) 2784-0030
FAX ..... (82) 2784-8014
SINGAPORE, Singapore ..... TEL ..... (65) 337-7033
FAX ..... (65) 338-1611
SCOTLAND, Stirling ..... TEL ..... (44) 7186-450024
FAX ..... (44) 1786-446188
SWITZERLAND, Geneva ..... TEL ..... (41) 22-788-0251
FAX ..... (41) 22-788-0617
SWEDEN,
Stockholm area ..... TEL ..... (08) 629-2850
(Bromma) FAX ..... (08) 98-0906
TAIWAN, Taipei ..... TEL ..... (886) 2715-3536
FAX ..... (886) 2712-2182
UNITED KINGDOM,
London area ..... TEL ..... (01483) 74-0440
(Woking) FAX ..... (01483) 75-6196
Manchester area ..... TEL ..... (01925) 83-0380
(Warrington) FAX ..... (01925) 83-0204

North American Representatives

ARIZONA,
Scottsdale – THORSON DESERT STATES ..... (602) 998-2444
CALIFORNIA,
Chula Vista – SONIKA ELECTRONICA ..... (619) 498-8340
CANADA,
Burnaby, B.C. – DAVETEK MARKETING ..... (604) 430-3680
Dorval, Quebec – POLAR COMPONENTS ..... (514) 683-3141
Kanata, Ontario – POLAR COMPONENTS ..... (613) 592-8807
Woodbridge, Ontario – POLAR COMPONENTS .... (416) 410-3377
ILLINOIS,
Skokie – INDUSTRIAL REPS, INC. ..... (847) 967-8430
INDIANA,
Kokomo – SCHILLINGER ASSOC. ..... (317) 457-7241
IOWA,
Cedar Rapids – LORENZ SALES ..... (319) 377-4666
KANSAS,
Merriam – LORENZ SALES ..... (913) 469-1312
Wichita – LORENZ SALES ..... (316) 721-0500
MEXICO,
Guadalajara – SONIKA ELECTRONICA ..... (523) 647-4250
Mexico City – SONIKA ELECTRONICA ..... (525) 754-6480
Monterrey – SONIKA ELECTRONICA ..... (528) 358-9280
MICHIGAN,
Brighton – COM-TEK SALES, INC ..... (810) 227-0007
Holland – COM-TEK SALES, INC ..... (616) 335-8418
MINNESOTA,
Edina – MEL FOSTER TECH. SALES, INC ..... (612) 941-9790
MISSOURI,
St Louis – LORENZ SALES ..... (314) 997-4558
NEBRASKA,
Lincoln – LORENZ SALES ..... (402) 475-4660
NEW YORK,
Plainview – COMPONENT CONSULTANTS ..... (516) 273-5050
East Syracuse – NYCOM ..... (315) 437-8343
Fairport – NYCOM ..... (716) 425-5120
OHIO,
Centerville – DOLFUSS ROOT & CO ..... (513) 433-6776
Powell – DOLFUSS ROOT & CO ..... (614) 781-0725
Middleburg Hts – DOLFUSS ROOT & CO ..... (216) 816-1660
PUERTO RICO,
Caguas – COMP REP ASSOC, INC ..... (787) 746-6550
UTAH,
Murray – FRONT RANGE MARKETING ..... (801) 288-2500
WASHINGTON,
Kirkland – ELECTRA TECHNICAL SALES ..... (206) 821-7442
WISCONSIN,
Pewaukee – Industrial Representatives, Inc. ..... (414) 574-9393

One AMD Place
P.O. Box 3453
Sunnyvale,
California 94088-3453
408-732-2400
Toll Free 800-538-8450
TWX 910-339-9280
TELEX 34-6306

TECHNICAL SUPPORT &
LITERATURE ORDERING

USA 800-222-9323
USA PC CPU Technical Support 408-749-3060

JAPAN 03-3346-7550
Fax 03-3346-9628
FAR EAST Fax 852-2956-0599

EUROPE & UK 44-(0)-1276-803299
Fax 44-(0)-1276-803298
BBS 44-(0)-1276-803211
FRANCE 0590-8621
GERMANY 089-450-53199
ITALY 1678-77224

ARGENTINA 001-800-200-1111,
after tone 888-263-8500
BRAZIL 000-811-718-5573
CHILE 800-570-048
MEXICO 95-800-263-4758

PC CPU Technical Support E-mail: hwsupt@brahms.amd.com
Europe Technical Support E-mail: euro.tech@amd.com
Europe Literature Request E-mail: euro.lit@amd.com
http://www.amd.com

RECYCLED &
RECYCLABLE

Printed in USA
Con-5.4M-3/97-1
21095A