*Preliminary Information*

# AMD-640™

# System Controller

## Data Sheet

AMD

# Contents

# List of Figures

# List of Tables

# 1 Features

The AMD-640™ chipset is a highly integrated system solution that delivers superior performance for the AMD-K5™ processor, AMD-K6™ MMX™ enhanced 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-645TM peripheral bus controller in a 208-pin PQFP package. The AMD-640 system controller features the 64-bit Socket 7 interface, integrated writeback cache controller, system memory controller, and PCI bus controller.

This document describes the features and operation of the AMD-640 system controller. For a description of the AMD-645 peripheral bus controller, see the *AMD-645 Peripheral Bus Controller Data Sheet*, order# 21095A. Key features of the AMD-640 system controller are provided in this section.

## 1.1 Processor Interface

- Supports all 64-bit AMD-K5 processors and AMD-K6 processors
- Operates at processor bus speeds up to 66 MHz
- Supports processors with internal (L1) writeback cache write allocate feature
- Supports processor bus NA# pipeline protocol
- Low voltage 3.3-V processor interface
- System management interrupt, memory remap, and STPCLK#

## 1.2 Integrated Cache Controller

- Direct-mapped writeback or writethrough secondary cache
- Global write enable to support pipelined burst synchronous SRAM (PBSRAM) cache access
- Flexible cache size — 0, 256 Kbytes, 512 Kbytes, 1 Mbyte, and 2 Mbytes

- 32-byte line size compatible with L1 cache
- Integrated 10-bit tag comparator
- 3-1-1-1 read/write timing for PBSRAM access at 66 MHz
- 3-1-1-1-1-1-1-1 back-to-back read timing for PBSRAM access at 66 MHz
- Sustained three-cycle access to PBSRAM, DRAM write buffer, and PCI write buffer at 66 MHz
- Data streaming for simultaneous primary and secondary cache line fills
- Cacheable, write-protected system and video BIOS
- Programmable cacheable region and cache timing

## 1.3 Integrated Memory Controller

- Supports the following combination of DRAM types:
  - Fast page mode (FPM), extended data out (EDO), and synchronous DRAM (SDRAM)
  - 1-, 2-, 4-, and 16-Mbit by n-bit DRAMs in one to six banks up to a total of 768 Mbytes
  - 32-bit and 64-bit data widths
  - Flexible row and column addressing
- 3.3-V and 5-V operation with no external buffers
- Bank-by-bank error correcting code options
- Two interleave options:
  - Two-bank interleaving for 16-Mbit SDRAMs
  - Two- and four-bank interleaving for 64-Mbit SDRAMs
- Four cache lines (16 quadwords) of processor-to-DRAM posted write buffers with full read-around and combine-and-store capability
- Concurrent DRAM writeback, read-around-write, and speculative DRAM read ahead
- Burst reads and writes
- Supports the following timings using 60 ns DRAM:
  - EDO DRAMs on a 50-MHz or 60-MHz bus:

    4-2-2-2 on-page, 7-2-2-2 start-page, and 9-2-2-2 page-miss

  - EDO DRAMs on a 66-MHz bus:

    5-2-2-2 on-page, 8-2-2-2 start-page, 11-2-2-2 page-miss, and 5-2-2-2-3-2-2-2 back-to-back access

- -15 SDRAMs on a 60-MHz bus, CAS latency = 2:

  5-1-1-1 on-page, 8-1-1-1 start-page, 10-1-1-1 page-miss, and 5-1-1-1-3-1-1-1 back-to-back access

- -12/-10 SDRAMs on a 100-MHz bus, CAS latency = 3:

  6-1-1-1 on-page, 9-1-1-1 start-page, 11-1-1-1 page-miss, and 6-1-1-1-3-1-1-1 back-to-back access

- Supports BIOS shadowing on 16-Kbyte boundaries
- Decoupled and burst DRAM refresh with staggered RAS timing
- Provides the following refresh options:
  - Programmable refresh rate
  - CAS-before-RAS
  - Populated banks only

## 1.4 PCI Bus Controller

- Support for five PCI masters
- 32-bit 3.3-V and 5-V PCI interface
- Synchronous PCI bus operation up to 33 MHz
- PCI initiator snoop-ahead and snoop filtering
- PCI initiator peer concurrence
- Automatic processor-to-PCI burst cycle detection
- Five-doubleword processor-to-PCI post write buffer
- 48-doubleword PCI-to-DRAM post write buffer (16 + 32)
- 26-doubleword DRAM-to-PCI prefetch buffer (10 +16)
- Byte merging on processor-to-PCI posted writes to reduce the number of PCI write cycles
- Zero wait state PCI initiator and target burst transfers
- PCI-to-DRAM data streaming up to 132 Mbytes per second
- Full compliance with *PCI Bus Specification, Revision 2.1*
- Enhanced PCI command optimization (MRL, MRM, MWI)
- Timer-enforced fair arbitration between PCI initiators

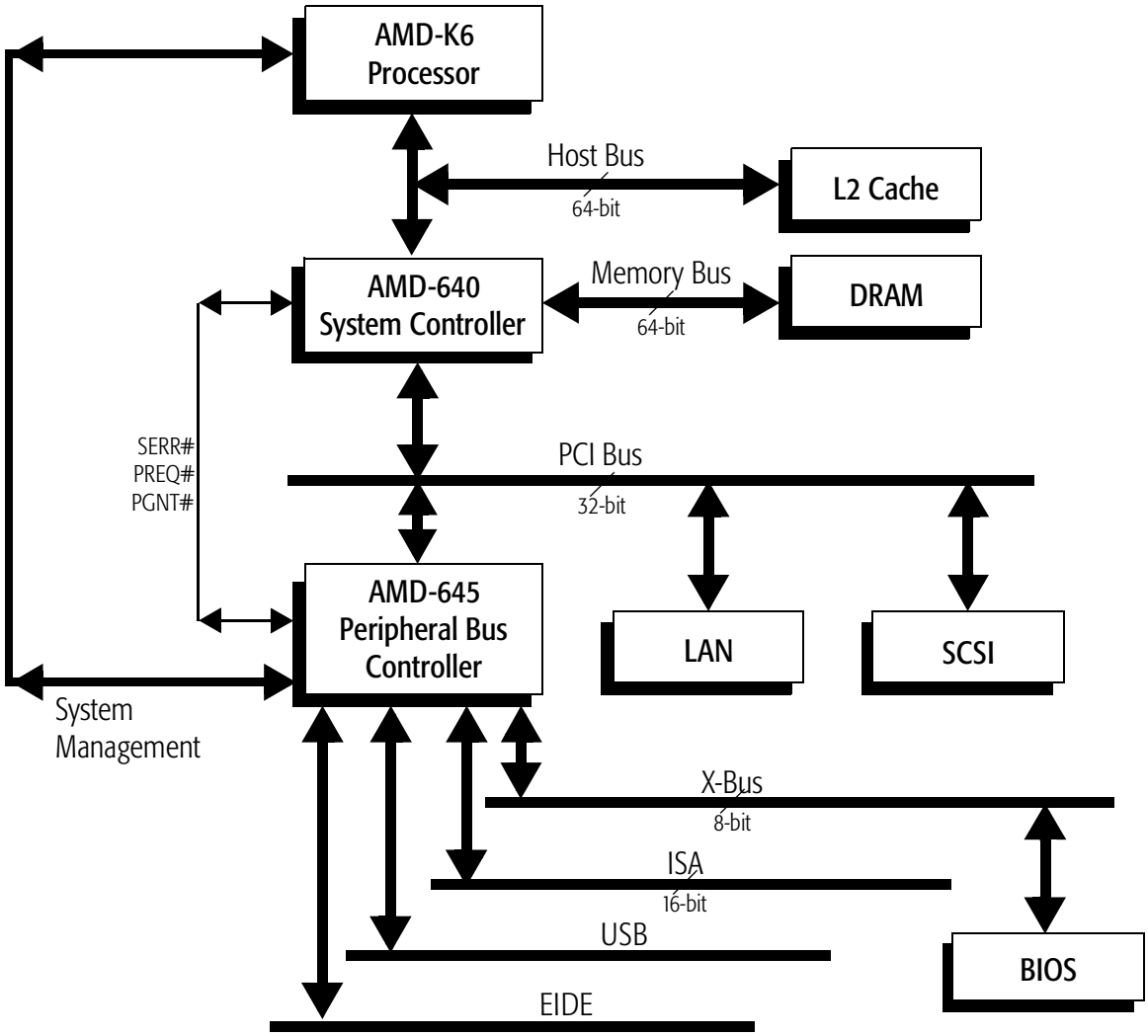



**Figure 1-1. AMD-640 Chipset System Block Diagram**

# 2 Overview

The AMD-640 system controller optimizes the interaction between the processor, optional synchronous L2 cache, DRAM, and the PCI bus with pipelined burst and concurrent transactions. It provides 3-1-1-1-1-1-1-1 timing for both read and write transactions with pipelined burst synchronous SRAMs running at 66 MHz. The AMD-640 system controller includes four cache lines (16 quadwords) of processor-to-DRAM or cache-to-DRAM write buffering with concurrent writeback capability to accelerate writeback and write miss cycles.

## 2.1 System

The local bus is a non-multiplexed bus based on AMD and Intel processors. The AMD-640 system controller is capable of performing I/O, single memory, and block memory transactions. The AMD-640 system controller memory controller can perform zero wait state memory reads and writes using an advanced data buffering design. However, in the event of a buffer miss, the memory controller inserts wait states using the BRDY# wait procedure. The controller responds only to I/O cycles within its configuration register space and memory requests as defined in its configuration registers. All cycle timing on the local bus is derived from the CPU clock (CCLK). This same signal drives the AMD-640 system controller host clock (HCLK) input, from which the controller derives all of its timing.

The AMD-640 system controller incorporates a high-performance, flexible 64-bit DRAM controller that provides the DRAM interface for either an AMD-K5 or AMD-K6 processor. The memory controller can perform zero wait state reads or writes through the use of a prefetch read buffer or a deep write buffer, respectively. It can address up to six banks of DRAMs in various combinations of 1 Mbit, 2 Mbit, 4 Mbit, and 16 Mbit by 32 or 64 bits, up to a total of 768 Mbytes. The DRAM can be any combination of fast page mode (FPM) DRAM, extended data out (EDO) DRAM, and Synchronous DRAM (SDRAM). Synchronous DRAM allows zero wait state

bursting between the DRAM and the controller's internal data buffers at 66 MHz. The DRAM controller can be configured to implement error correction code (ECC) data integrity checking. The BIOS must determine the type of memory installed and program the configuration registers accordingly.

The AMD-640 system controller supports shadowing to accelerate video and system BIOS accesses. The shadow RAM can also be configured to be cacheable and write-protected. The unused portion of DRAM can be relocated to increase overall system size. In addition, the AMD-640 system controller can be programmed to perform writes to flash EPROM, enabling field BIOS upgrades.

The AMD-640 system controller is fully compatible with the *PCI Local Bus Specification, Revision 2.1*. It can operate at either 3.3 V or 5 V, and offers 64-bit to 32-bit data conversion. A five-doubleword posted write buffer enables concurrent processor and PCI operation. Consecutive processor addresses are converted into burst PCI cycles with byte-merging capability for optimal processor-to-PCI throughput. A 48-doubleword PCI post write buffer and a 32-doubleword PCI prefetch buffer facilitate concurrent PCI, DRAM, and cache transactions. Enhanced PCI bus commands such as Memory-Read-Line, Memory-Read-Multiple, and Memory-Write-Invalid maximize data throughput. The AMD-640 system controller employs a variety of techniques to minimize PCI initiator read latency and DRAM access, including snoop ahead, snoop filtering, forwarding L1 writebacks to the PCI initiator, and merging L1 writebacks into the PCI posted write buffers. These techniques minimize PCI initiator read latency and DRAM utilization. The combination of these features allows a PCI initiator to achieve the full 133-Mbyte burst transfer rate.

Figure 2-1 illustrates the full complement of features and functions built into the AMD-640 system controller's system logic. The configuration of the AMD-640 system controller can be programmed via I/O-mapped configuration registers. A PCI-to-CPU read buffer can assemble up to eight bytes of data. A five-doubleword CPU-to-PCI write buffer allows the processor to post up to five writes without adding delay on the local bus for PCI to complete the cycles. A 26-doubleword DRAM-to-PCI read buffer enables the controller logic to prefetch data, eliminating stalls on PCI while waiting for data from DRAM. A

48-doubleword PCI-to-DRAM write buffer allows PCI initiators to post writes to memory without adding delay on either the PCI or processor bus. In addition, the AMD-640 system controller contains a PCI arbiter.

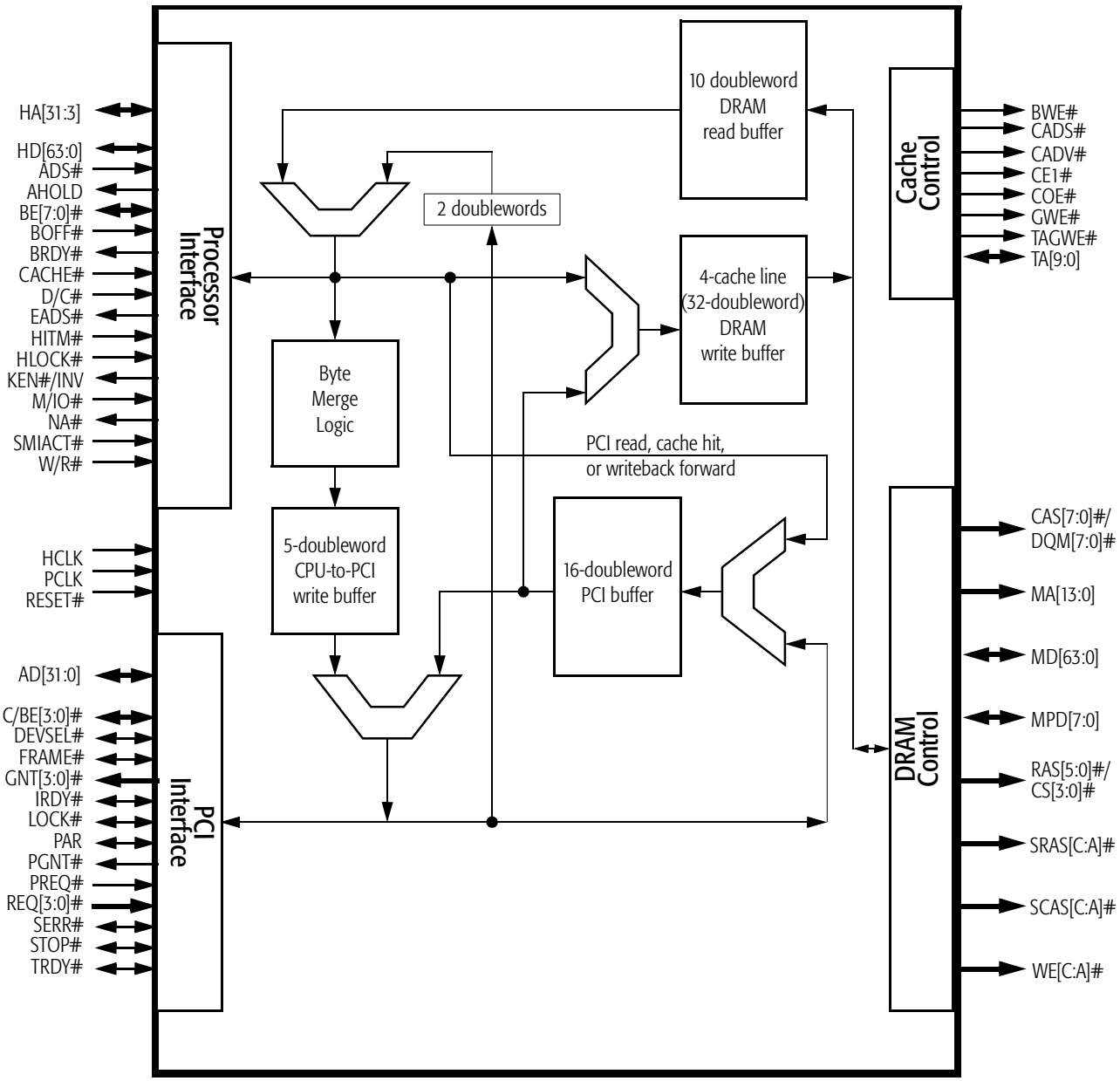



**Figure 2-1. AMD-640 System Controller Block Diagram**

# 2.2 AMD-640 System Controller Buffers

Figure 2-2 and Figure 2-3 show the basic construction of the buffers in the AMD-640 system controller. Figure 2-2 shows a path from a 64-bit bus to a 32-bit bus (memory-to-PCI). Figure 2-3 shows a path from a 32-bit bus to a 64-bit bus (PCI-to-memory). The control logic assembles 32-bit words into 64-bit words or disassembles 64-bit words into 32-bit words.

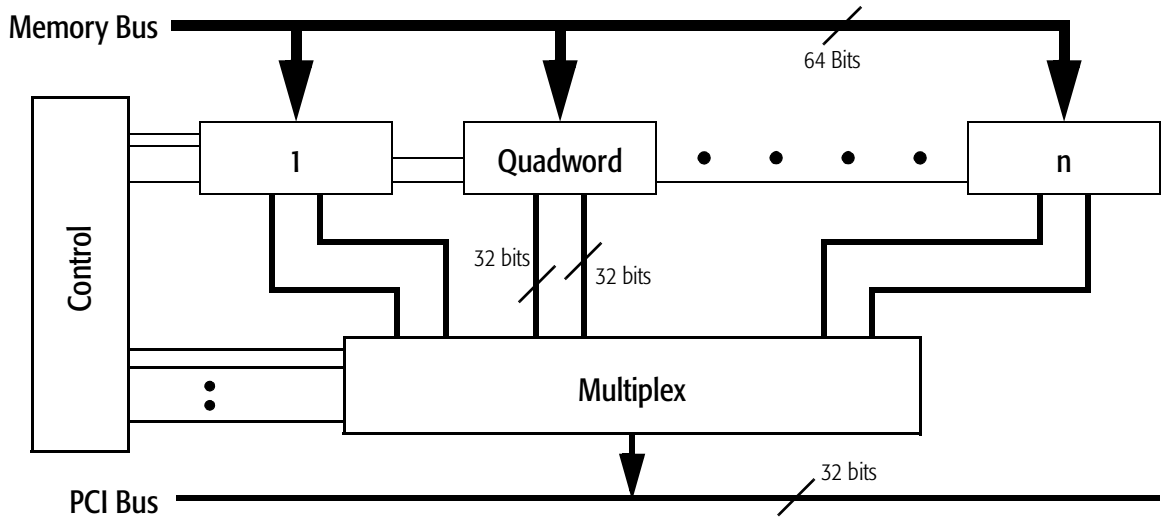



**Figure 2-2. Memory-to-PCI Buffer**

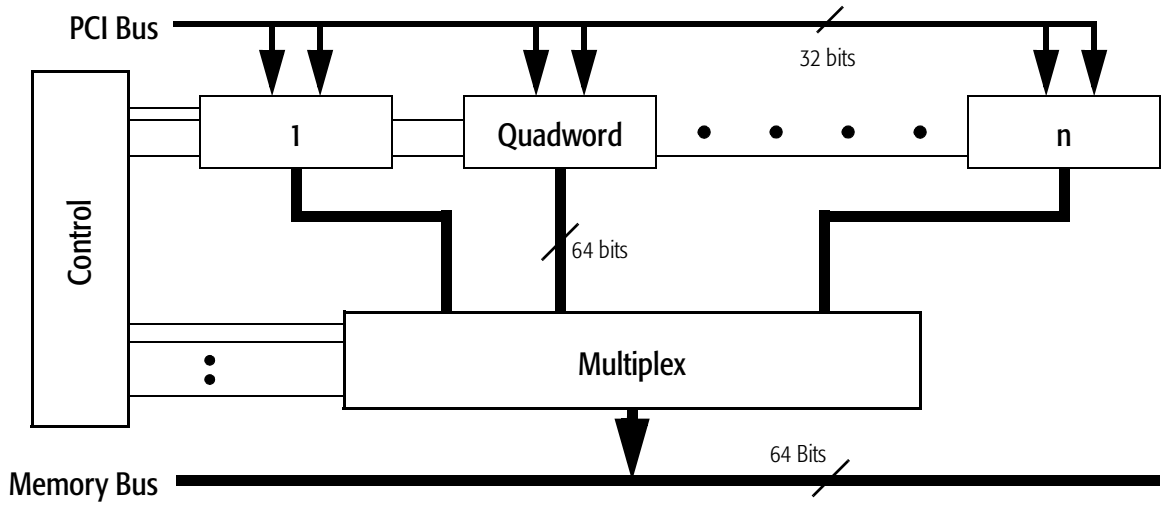



**Figure 2-3. PCI-to-Memory Buffer**

## 2.3 Definitions, Conventions, and References

- Active-Low Signals—Signal names containing a pound sign, such as ADS#, indicate active-low signals. They are asserted in their low-voltage state and negated in their high-voltage state.
- Reserved Bits and Signals—Signals or bus bits marked *reserved* must be driven inactive or left unconnected, as indicated in the signal descriptions. These bits and signals are reserved by AMD for future implementations. When software reads registers with reserved bits, the reserved bits must be masked. When software writes such registers, it must first read the register and change only the non-reserved bits before writing back to the register.
- Three-State—In timing diagrams, signal ranges that are high impedance are shown as a straight horizontal line half-way between the high and low levels.
- Invalid and Don't Care—In timing diagrams, signal ranges that are invalid or don't care are filled with a screen pattern.

### 2.3.1 Data

- Quantities—A word is two bytes (16 bits), a dword or doubleword is four bytes (32 bits), and a qword or quadword is eight bytes (64 bits).
- Addressing—Memory is addressed as a series of bytes on eight-byte (64-bit) boundaries, in which each byte can be separately enabled.
- Abbreviations—The following notation is used for bits and bytes:
  - kilo k.....as in 4 kbytes/page
  - Mega M.....as in 4 Mbits/sec
  - Giga G.....as in 4 Gbytes of memory space
- Little-Endian Convention—The byte with the address xx...xx00 is in the least-significant byte position (little end). In byte diagrams, bit positions are numbered from right to left: the little end is on the right and the big end is on the left. Data structure diagrams in memory show small addresses at the bottom and high addresses at the top. When data items are aligned, bit notation on a 64-bit data bus maps directly to bit notation in 64-bit-wide memory. Because byte addresses increase from right to left, strings appear in reverse order when illustrated.
- Bit Ranges—In a range of bits, the highest and lowest bit numbers are separated by a dash, as in 63–00.
- Bit Values—Bits can either be set to 1 or cleared to 0.
- Hexadecimal and Binary Numbers—Unless the context makes interpretation clear, hexadecimal numbers are followed by an h, binary numbers are followed by a b, and decimal numbers are followed by a d.

### 2.3.2 Related Publications

The following books discuss various aspects of computer architecture that may be useful for your understanding of AMD products:

**AMD Publications**

*AMD-K5 Processor Data Sheet*, order# 18522

*AMD-K5 Processor Technical Reference Manual*, order# 18524

*AMD-K6 MMX Enhanced Processor Data Sheet*, order# 20695

*AMD-645 Peripheral Bus Controller Data Sheet*, order# 21095

**Bus Architecture**

*PCI Local Bus Specification, Revision 2.1*, PCI Special Interest Group, Hillsboro, Oregon, 1993.

*AT Bus Design*, Edward Solari, IEEE P996 Compatible, Annabooks, San Diego, CA, 1990.

**x86 Architecture**

*Programming the 80386*, John Crawford and Patrick Gelsinger, Sybex, San Francisco, 1987.

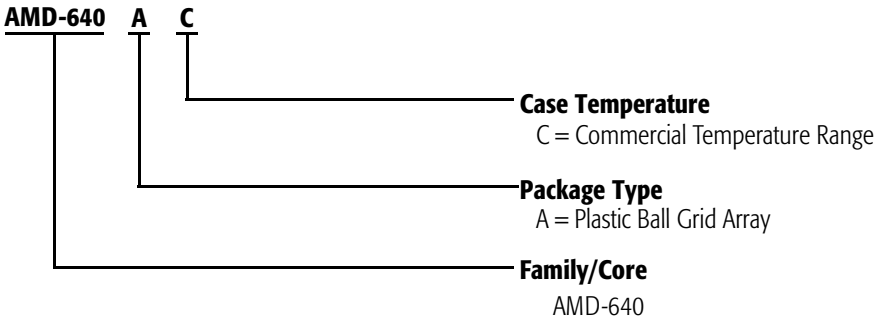
*80x86 Architecture & Programming*, Rakesh Agarwal, Volumes I and II, Prentice-Hall, Englewood Cliffs, NJ, 1991.

**General References**

*Computer Architecture*, John L. Hennessy and David A. Patterson, Morgan Kaufman Publishers, San Mateo, CA, 1990.

# 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. Contact your AMD representative for detailed ordering information.




**Valid Combinations**

| OPN | Package Type | Operating Voltage | Case Temperature |
|---|---|---|---|
| AMD-640AC | 328-pin PBGA | 4.75 V–5.25 V | 70°C |

***Notes:***

1. *Valid Combinations lists configurations planned to 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 Processor Interface Signals

**ADS#** *Address Strobe* *Input*

ADS# indicates to the AMD-640 system controller that a new bus cycle is starting. When ADS# is asserted, the AMD-640 system controller latches the address bus and all cycle definition signals corresponding to this bus cycle on the rising edge of HCLK.

**AHOLD** *Address Hold* *Output*

The AMD-640 system controller asserts AHOLD off the rising edge of HCLK while a PCI initiator accesses main memory to perform an inquire cycle. The host processor responds by floating HA[31:3] to allow the AMD-640 system controller to drive the address bus. See the timing diagrams in Section 6 for example cycles.

**BE[7:0]#** *Byte Enables* *Inputs*

The AMD-640 system controller samples BE[7:0]# to determine the valid data bytes during a write cycle and the requested data bytes during a read cycle. The eight byte enable signals correspond to the eight bytes of the data bus as follows:

- BE7#: D[63:56]
- BE6#: D[55:48]
- BE5#: D[47:40]
- BE4#: D[39:32]
- BE3#: D[31:24]
- BE2#: D[23:16]
- BE1#: D[15:8]
- BE0#: D[7:0]

**BOFF#** *Backoff* *Output*

The AMD-640 system controller asserts BOFF# to acquire the host bus during PCI-to-DRAM cycles in order to perform snoop cycles and access the L1 and L2 caches in the event of a cache hit. The processor unconditionally aborts any cycles in

progress and transitions to a bus hold state by floating the following signals:

- A[31:3]
- ADS#
- BE[7:0]#
- CACHE#
- D[63:0]
- D/C#
- LOCK#
- M/IO#
- W/R#

These signals remain floated until BOFF# is negated.

**BRDY#** ***Burst Ready*** ***Output***

The AMD-640 system controller asserts BRDY# to the host processor off the rising HCLK edge. During a read cycle, asserting BRDY# indicates that the data bus is being driven with valid data. During a write cycle, it indicates that the data bus has been latched.

**CACHE#** ***Cacheable Access*** ***Input***

The host processor asserts CACHE# during a cacheable read cycle to indicate that it will perform a burst line fill. It asserts CACHE# during a cacheable write cycle to indicate that it will perform a burst writeback cycle. When the AMD-640 system controller samples CACHE# low, it stores processor read or write data in the L2 cache.

**D/C#** ***Data/Control*** ***Input***

When the AMD-640 system controller samples D/C# low, it generates a command on the C/BE[3:0]# PCI bus signals in the command phase of processor-to-PCI bus cycles.

**EADS#** ***External Address Strobe*** ***Output***

The AMD-640 system controller asserts EADS# off the rising HCLK edge to snoop each cache line transferred during all PCI-to-DRAM cycles. EADS# strobes the snoop address into the L1 cache. On L1 cache hits, the processor invalidates unmodified data during writes, and sources (drives) modified data during PCI initiator reads and writes.

**HA[31:3]** ***Host Address Bus*** ***Input/Output***

The AMD-640 system controller samples addresses driven by the processor on HA[31:3] during memory and I/O cycles and

forwards them to the PCI bus or DRAM, depending on the address range.

During PCI-to-DRAM cycles, the AMD-640 system controller drives the address bus to snoop the processor's cache and the L2 cache.

**HD[63:0]** ***Host Data Bus*** ***Bidirectional***

HD[63:0] connects to the host processor's 64-bit data bus. Each of the eight bytes of data that comprise this bus is qualified by a corresponding byte enable signal (BE[7:0]#).

**HITM#** ***Inquire Cycle Hit To Modified Line*** ***Input***

The AMD-640 system controller samples HITM# to determine if an L1 cache snoop has found a modified line. A low on HITM# indicates that a cache line write by the processor is imminent. HITM# is deasserted after the line is written.

**HLOCK#** ***Host Bus Lock*** ***Input***

The host processor asserts HLOCK# to indicate that it requires exclusive access to the local bus during a sequence of bus cycles. When the AMD-640 System Controller samples HLOCK# low, it withholds bus grants to other PCI initiators. If a grant has already been issued to a PCI initiator, the AMD 640 will not assert BOFF# for L1 snoops. These actions effectively suspend a PCI-DRAM transfer until HLOCK# is deasserted.

**KEN#/INV** ***Cache Enable/Invalidate*** ***Output***

During host processor read cycles, KEN#/INV functions as the cache enable signal (KEN#), indicating a cacheable address when low and a non-cacheable address when high. KEN#/INV is driven off the rising HCLK edge.

During inquire cycles, KEN#/INV functions as the invalidate signal (INV), which determines whether an addressed cache line that is found in the host processor's L1 cache transitions to the invalid or shared state.

**M/IO#** ***Memory or I/O*** ***Input***

The AMD-640 System Controller samples M/IO# during a bus cycle to determine whether the host processor is addressing

memory or I/O space. When M/IO# is high, the AMD-640 System Controller enables accesses to DRAM and the L2 cache. When the access is not targeted to the cache or the DRAM, the AMD-640 system controller uses M/IO# to generate the PCI commands on C/BE[3:0] during the command phase of CPU-to-PCI cycles.

**NA#** ***Next Address*** ***Output***

The AMD-640 system controller asserts NA# off the rising HCLK edge to indicate to the host processor that it is ready to accept a pipelined address.

**SMIACT#** ***System Management Interrupt Acknowledge*** ***Input***

When SMIACT# from the host processor is sampled asserted, it indicates to the AMD-640 system controller that the processor has entered system management mode (SMM). If configuration register 63h bit, 1 is set, asserted SMIACT# redirects memory accesses from 30000h:3FFFFh to B0000h:BFFFFh. Accesses to this memory area are passed through to the PCI bus when SMIACT# is high, as this is normally the Video Buffer area.

**W/R#** ***Write/Read*** ***Input***

The AMD-640 system controller samples W/R# to determine whether the current processor cycle is a write or a read.

## 4.2 PCI Interface Signals

**AD[31:0]** ***PCI Address/Data Bus*** ***Bidirectional***

AD[31:0] contain the PCI address during the first clock cycle in which FRAME# is asserted, and contain data during subsequent clock cycles. As an initiator, the AMD-640 system controller drives these lines with the address of the target. As a target, the AMD-640 system controller decodes these lines to determine what area of memory to read or write.

**C/BE[3:0]#** ***PCI Command/Byte Enables*** ***Bidirectional***

C/BE[3:0]# contain the PCI command during the first clock cycle that FRAME# is asserted. These lines serve as byte enable signals for subsequent cycles.

**DEVSEL#** ***PCI Device Select*** ***Bidirectional***

The AMD-640 system controller samples DEVSEL# when it is the initiator in a PCI cycle to determine if the target device has responded. The AMD-640 system controller asserts DEVSEL# when it is the targeted device in a PCI cycle.

**FRAME#** ***PCI Cycle Frame*** ***Bidirectional***

The AMD-640 system controller asserts FRAME# at the beginning of a PCI cycle when it is the initiator, and holds it low until the beginning of the last data transfer in the cycle.

If the AMD-640 system controller is the targeted PCI device, it samples and latches the C/BE[3:0]# and AD[31:3] signals and asserts DEVSEL# at the first PCLK edge on which it samples FRAME# asserted.

**GNT[3:0]#** ***PCI Bus Grant*** ***Outputs***

As the PCI bus arbiter, the AMD-640 system controller asserts one of these device-specific bus grant signals off the rising clock edge to indicate to an initiator that it has been granted control of the PCI bus.

**IRDY#** ***Initiator Ready*** ***Bidirectional***

IRDY# indicates that a PCI initiator is ready to complete the current data phase of the transaction. During a read cycle, asserted IRDY# indicates the master is ready to accept the data. During a write cycle, asserted IRDY# indicates that write data is valid on AD31:AD0. Data is transferred on the PCI bus on each PCLK in which both IRDY# and TRDY# are asserted. Wait states are inserted on the bus until both IRDY# and TRDY# are asserted together.

IRDY# is an output when the AMD-640 system controller is the PCI initiator. The AMD-640 system controller drives IRDY# low one PCLK after it asserts FRAME# and holds it low until one cycle before the end of all transactions.

IRDY# is an input when the AMD-640 system controller is a PCI target. The AMD-640 system controller does not terminate a read or write cycle until it samples both IRDY# and TRDY# low.

**LOCK#** ***PCI Bus Lock*** ***Bidirectional***

As a PCI initiator, the AMD-640 system controller asserts LOCK# to prevent other devices from accessing the target device during atomic CPU-to-PCI transactions.

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

The AMD-640 system controller drives PAR as a PCI initiator one clock after the address phase and each data write phase to generate even parity across AD[31:3] and C/BE3:0]#. It drives PAR as a PCI target one clock after each data read phase.

**PGNT#** ***PCI Grant to AMD-645 Peripheral Bus Controller*** ***Output***

PGNT# is asserted off the rising clock edge and grants control of the PCI bus to the PCI-ISA/IDE bridge functions implemented in the AMD-645 peripheral bus controller.

**PREQ#** ***PCI Request from AMD-645 Peripheral Bus Controller*** ***Input***

The AMD-640 system controller samples PREQ# to determine if the AMD-645 peripheral bus controller needs PCI bus access.

**REQ[3:0]** ***PCI Bus Request*** ***Inputs***

As the PCI bus arbiter, the AMD-640 system controller samples these device-specific bus request signals to determine if another agent requires control of the PCI bus.

**SERR#** ***System Error*** ***Output***

A PCI agent (the AMD-640 system controller or other device) asserts SERR# off the rising clock edge one clock after it detects a system error. SERR# is an input to the AMD-645 peripheral bus controller, which can be programmed to generate an NMI.

**STOP#** ***PCI Bus Stop*** ***Input***

As a PCI initiator, the AMD-640 system controller samples STOP# to determine if the target device requires it to abort or retry a transaction.

**TRDY#** ***Target Ready*** ***Bidirectional***

As a PCI initiator, the AMD-640 system controller samples TRDY# to determine when the target agent is able to complete the data phase of a transaction.

As a PCI target, the AMD-640 system controller asserts TRDY# to indicate that it has latched the data on AD31:AD0 during a write phase or driven data on AD31:AD0 during a read phase.

# 4.3 DRAM Interface Signals

**CAS[7:0]#/ DQM[7:0]#** — ***Column Address Strobe/ Data Mask*** — ***Outputs***

CAS[7:0]# generate column address strobes for FPM and EDO DRAM during processor-to-DRAM and PCI-to-DRAM cycles. CAS7# connects to the most-significant byte of each bank, and CAS0# connects to the least-significant byte. The width and delay of these signals are adjustable.

For SDRAM, these lines function as data masks DQM[7:0]# for each byte during SDRAM write cycles.

**MA[13:0]** — ***Memory Address*** — ***Outputs***

The multiplexed row and column address bits MA[13:0] connect to the system DRAMs. They can address any size DRAM from 256 Kbits to 16 Mbits by n bits.

**MD[63:0]** — ***Memory Data*** — ***Bidirectional***

MD[63:0] connect to the DRAM data bus. They are driven by the DRAM when reading. They are driven by the AMD-640 system controller during writes.

**MPD[7:0]** — ***Memory ECC*** — ***Bidirectional***

MPD[7:0] carry error correction codes for the eight bytes of data on MD[63:0]. They are inputs to the AMD-640 system controller during DRAM read cycles and outputs during DRAM write cycles.

**RAS[5:0]#/ CS[5:0]#** — ***Row Address Strobe 5:0/ Chip Selects 5:0*** — ***Outputs***

RAS[5:0]# generate row address strobes for the DRAM banks, either during CPU-to-DRAM or PCI-to-DRAM accesses or in sequence during DRAM refresh cycles.

CS[5:0]# function as chip select lines for SDRAMs if bits 5–4 in configuration register 60h select SDRAM.

**SCASA#, SCASB#, SCASC#** — *Synchronous DRAM Column Address Strobe* — *Outputs*

SCASA#, SCASB#, and SCASC# are column address strobe pins for synchronous DRAM. They operate in parallel to drive greater loads than a single pin can support.

**SRASA#, SRASB#, SRASC#** — *Synchronous DRAM Row Address Strobe* — *Outputs*

SRASA#, SRASB#, and SRASC# are row address strobe pins for synchronous DRAM. They operate in parallel to drive greater loads than a single pin can support.

**WEA#, WEB#, WEC#** — *Synchronous DRAM Memory Write Enable* — *Outputs*

WEA#, WEB#, and WEC# are write enable pins for all DRAM. They operate in parallel to drive greater loads than a single pin can support.

# 4.4 Cache Controller Interface Signals

**BWE#** ***Byte Write Enable*** ***Output***

BWE# connects to the BWE# input on each of the L2 cache SRAMs. When the AMD-640 system controller L2 cache controller asserts BWE# off the rising clock edge during cache writes, data on the processor bus (D[63:0]) is written to those bytes of the cache SRAM whose byte-enable lines (BE[7:0]#) are asserted.

**CADS#** ***Cache Address Strobe*** ***Output***

The AMD-640 system controller normally drives CADS# high. It enables CADS# to be asserted when it acquires the host processor bus by asserting BOFF#, and asserts CADS# off the rising clock edge during PCI-to-DRAM cycles that hit the L2 cache.

**CADV#** ***Cache Advance*** ***Output***

CADS# connects to the ADS# inputs of the L2 cache SRAMs. The AMD-640 system controller asserts CADS# off the rising clock edge during L2 cache line read and write hits as well as during line fills and line writebacks, incrementing the SRAM's internal counters to advance to the next quadword in the cache line.

**CE1#** ***Chip Enable 1*** ***Output***

The CE1# chip select signal enables the L2 cache for both reads and writes. It is asserted off the rising clock edge.

**COE#** ***Cache SRAM Output Enable*** ***Output***

The AMD-640 system controller asserts COE# off the rising clock edge of a cache read hit or writeback cycle and holds it low for the duration of the cycle to enable cache SRAM output. It also asserts COE# during the first two clock cycles of CPU-to-PCI memory reads, non-cacheable reads, or read misses without writeback.

**GWE#** *Global Write Enable* *Output*

GWE# connects to the global write inputs of the cache SRAMs. The AMD-640 system controller asserts GWE# off the rising clock edge during L2 cache line fills to enable the SRAMs to receive each quadword of the line being returned by the DRAM controller.

**TA[9:0]** *Tag Address* *Bidirectional*

TA[9:0] are used to read and write the cache page number to and from external tag RAM. They function as outputs during L2 cache line fills and as inputs at all other times.

**TAGWE#** *TAG Write Enable* *Output*

The AMD-640 system controller asserts TAGWE# on the rising edge of HCLK to enable the L2 cache tag SRAMs to receive the next tag address.

# 4.5 Clocks and Reset

**HCLK** ***Host Clock*** ***Input***

HCLK receives a buffered host clock. It is used by all of the AMD-640 system controller logic in the host clock domain. It is the primary reference for all bus cycles on the processor and memory buses as well as most of the internal logic.

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

PCLK receives a buffered host clock divided by two. It is used by all of the AMD-640 system controller logic in the PCI clock domain.

**RESET#** ***Reset*** ***Input***

Asserting RESET# resets the AMD-640 system controller and sets all register bits to their default values. Bidirectional pins are three-stated and outputs are driven inactive. This signal is driven by the PCIRST# signal from the AMD-645 peripheral bus controller.

# 5 Functional Operation

## 5.1 Processor Interface

The AMD-640 system controller responds to CPU-generated bus signals and activates the PCI, DRAM, and cache state machines according to the command type and address range. On memory cycles it drives the processor address onto the memory bus from its integrated DRAM controller. For PCI target cycles it drives the PCI bus from its integrated PCI buffers and control logic.

The AMD-640 system controller maintains coherency of the processor primary (L1) cache with the rest of the system using the KEN#, EADS#, and HITM# pins. It monitors the CACHE# signal from the processor to determine burst cycles and returns KEN# asserted when data is cacheable. KEN# is normally active during a memory read cycle unless the processor address lies outside the cacheable region. In this case, the AMD-640 system controller deasserts KEN# before the completion of the first burst transfer so that the data is not written to the L1 cache. The AMD-640 system controller does not write data to the secondary (L2) cache when CACHE# is inactive unless it is programmed to do so by setting bit 2 of register 52h. It asserts the EADS# signal during DMA and PCI initiator cycles to snoop the L1 cache. The processor responds to a cache hit by asserting the HITM# line. This action notifies the AMD-640 system controller that a modified cache line must be written back to the system before the intended memory access can be performed. A snoop filtering mechanism in the AMD-640 system controller minimizes snoop overhead by ensuring that consecutive accesses to the same cache line are snooped only once.

### 5.1.1 Write Posting

The AMD-640 system controller contains four write buffers to enhance memory write performance. Each buffer can hold one entire cache line, also referred to as a data block, which is 32

bytes (four quadwords). The write buffers are always enabled. The memory controller supports both single and block writes. Block writes are more common in a typical system than single writes because most processors use writeback caches, which transfer data in blocks. When a writeback cache is employed, the AMD-640 system controller sees a block transaction every time the processor clears a cache line. The controller's posted write buffers can handle four back-to-back block transactions without wait states. Figure 5-1 shows how the posted write buffers are organized.

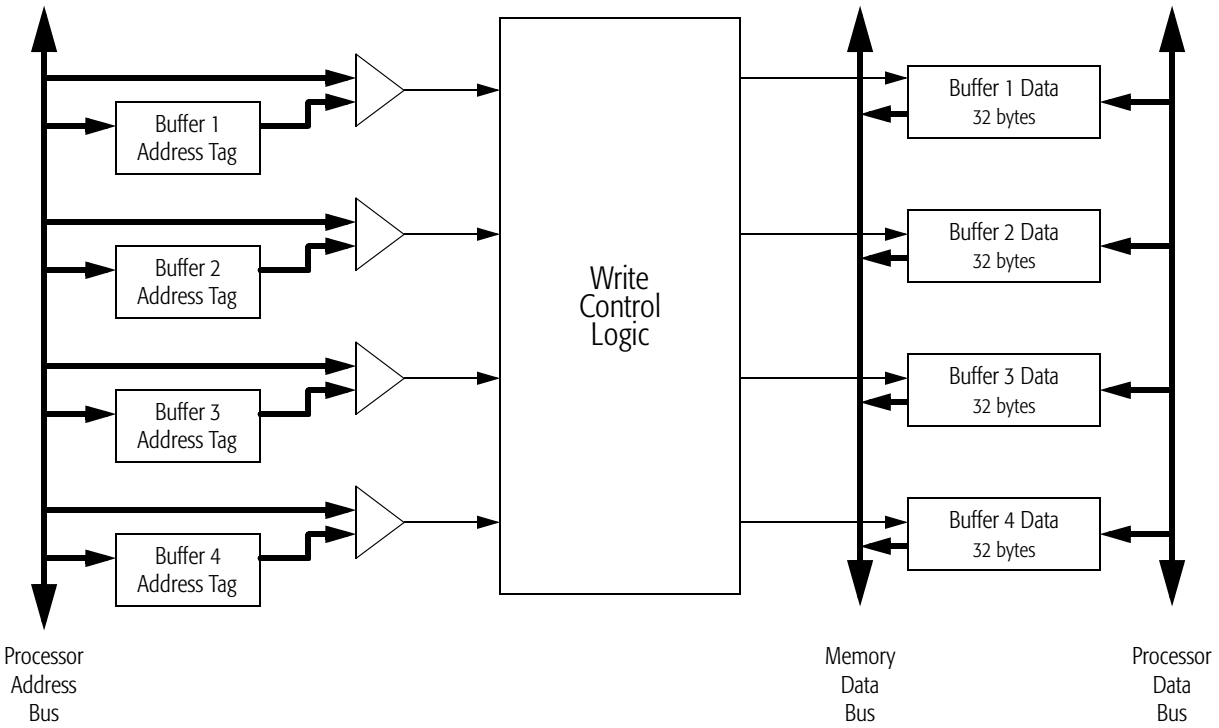



**Figure 5-1. Posted Write Buffer Organization**

The write buffers are organized as pseudo FIFO (First-In-First-Out) buffers, i.e. writes from the buffers to memory are performed in the order they are received from the processor. Four consecutive write transactions, whether single or block, will fill all four 32-byte buffers. Write buffers continue to accept data until either the buffers are full or all data from the processor has been received, at which point the controller begins writing data to the DRAM. As each pending write to main memory is performed, freeing the corresponding buffer,

the memory controller will assert BRDY# to accept another block from the processor if one is pending.

Each write buffer has its own address tag bits, which are compared to the address on the processor address bus. In a write cycle, the comparators determine the next buffer (if any) available to accept processor data. In a read cycle, the comparators are used to snoop the write buffers to maintain data coherency. If a read address matches one of the write buffer address tags (buffer hit), the read cycle is stalled by deasserting BRDY# until the write to memory is retired. If no match occurs, a read around write can be performed (page 5-5).

### 5.1.2 Read Buffer

The AMD-640 system controller contains five 8-byte read buffers, each of which can hold an entire 64-bit word of data. The buffers are designed to increase memory read performance by prefetching data from the main memory and supplying the data to the processor with zero wait states.

The read buffers are organized in a manner similar to a five-way set-associative cache, with the set-associativity dictated by an address affinity. Each of the read buffers has it's own address tag bits. On every read cycle, the address being requested is compared to the addresses of the read buffer lines. Figure 5-2 shows how the read buffers are organized.

If one of the buffers contains a requested quadword address, the data is presented to the processor with zero wait states and the next quadword is then prefetched into the same buffer. If no buffer contains the quadword, the controller reads it and the next quadword from memory into the read buffer. For a block read cycle, the next line (four quadwords) of data is prefetched into the read buffer.

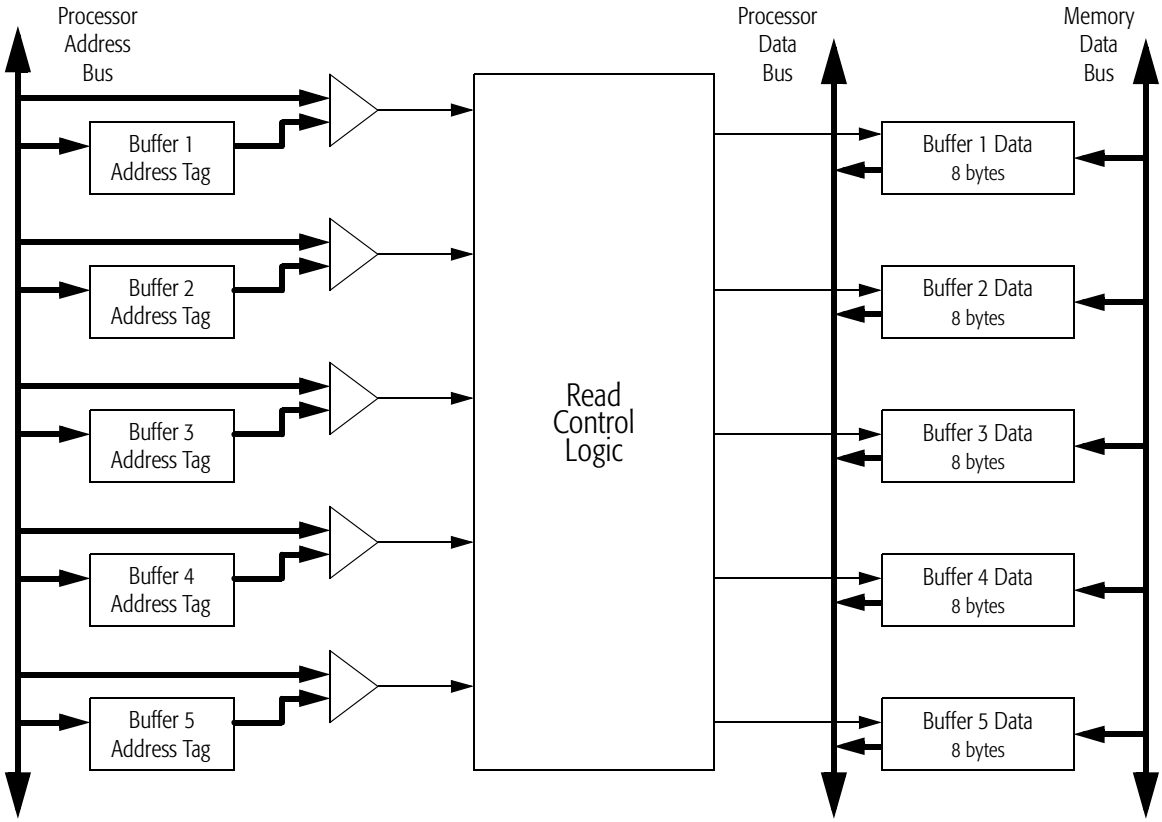



**Figure 5-2. Read Buffers**

Memory reads that fill the processor's caches are by far the most common types of reads. These reads occur as a burst read of four quadwords (32 bytes). When a burst read hits in the controller's read buffer, the transactions are identical to the single qword read described above except that the BRDY# signal is extended for three more clocks and, following the first qword transfer, three more quadwords of data are output onto the data bus at the rate of one quadword per clock. Thereafter, subsequent blocks that access sequential locations are all prefetch-queue hits because the memory controller fills the read queue as an integral part of the read cycle.

The read buffers snoop write transactions to maintain data coherency. If a write transaction occurs to an address whose data is contained in one of the read buffers, that read buffer is invalidated.

### 5.1.3 Read-Around-Writes

This feature minimizes processor stalling by interrupting a write in progress to service a processor read, effectively giving read priority over write. The DRAM controller finishes writing the current word, reads the desired data into the CPU read buffer, then continues the write from the post write buffer. In the special case of a read to an address contained in the post write buffer, the read will not proceed until the write has completed. Read-around-write is enabled by bit 7 of offset 53h.

## 5.2 Cache Controller

The AMD-640 system controller supports direct-mapped cache systems with data sizes ranging from 128 Kbytes to 2 Mbytes. It can accommodate both synchronous and asynchronous data SRAMs to provide flexibility for system trade-offs between cost and performance. Either writeback or writethrough cache schemes are available, and writeback can be implemented with or without a modify bit. If no modify bit is used in a writeback scheme, all lines are treated as modified. This scheme offers a larger cacheable region (compare Table 5-1 and Table 5-2) but does not perform as well as one with a modify bit.

### 5.2.1 Cache Organization

The configuration of tag lines TA[9:0] determines the L2 cache size and address range. Most L2 cache schemes employ 8-bit tags, in which case only the lower eight tag lines are used. The size of the cache determines the particular address lines to which TA[7:0] correspond. Table 5-1 shows some typical 8-bit tag configurations.

**Table 5-1. Common 8-Bit Tag Configurations**

| Cache Size | TA[7:0] | Tag Size | Tag RAM Address | Cacheable Region |
|---|---|---|---|---|
| 256 Kbytes | A[25:18] | 8Kx8 | A[17:5] | 64 Mbytes minus cache size |
| 512 Kbytes | A[26:19] | 16Kx8x2 | A[18:5] | 128 Mbytes minus cache size |
| 1 Mbyte | A[27:20] | 32Kx8 | A[19:5] | 256 Mbytes minus cache size |
| 2 Mbytes | A[28:21] | 32Kx8x2 | A[20:5] | 512 Mbytes minus cache size |

Figure 5-3 shows how the AMD-640 system controller connects to a typical 8-bit tag cache.

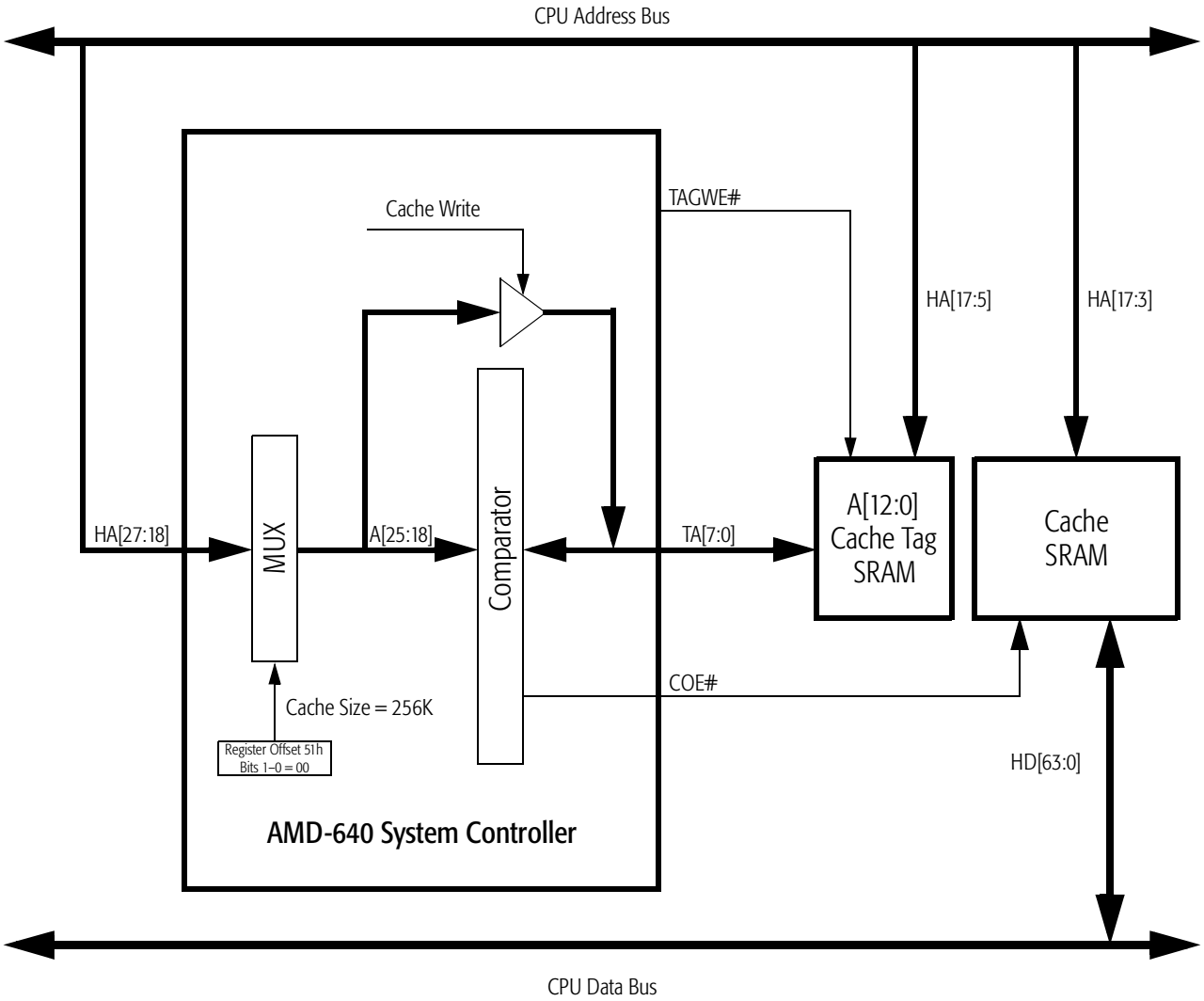



**Figure 5-3. 8-Bit Tag Cache Connections**

The AMD-640 system controller can support a 9-bit or 10-bit tag RAM by enabling TA9 and TA8 in configuration register 50h. Refer to section 7.4.1 on page 7-11. TA8 extends the cacheable region to one gigabyte. TA9 extends it to two gigabytes. Alternatively, TA7 can be programmed to function as a modify bit, as shown in Table 5-2.

**Table 5-2. Writeback Configurations for 7-Bit Tag with Modify Bit**

| Cache Size | TA[6:0] | Tag Size | Tag RAM Address | Cacheable Region |
|---|---|---|---|---|
| 256 Kbytes | A[24:18] | 8Kx8 | A[17:5] | 32 Mbytes minus cache size |
| 512 Kbytes | A[25:19] | 16Kx8x2 | A[18:5] | 64 Mbytes minus cache size |
| 1 Mbyte | A[26:20] | 32Kx8 | A[19:5] | 128 Mbytes minus cache size |
| 2 Mbytes | A[27:21] | 32Kx8x2 | A[20:5] | 256 Mbytes minus cache size |

## 5.2.2 Cache Operation

The AMD-640 system controller contains an integrated 10-bit cache tag comparator which is active during every cache access cycle from either the processor or a PCI initiator. It compares the command address with the tag SRAM to determine if the cycle is a cache hit or cache miss.

### Cache Hits

The action taken on a cache read hit is the same for all cache schemes, but varies for different schemes on a cache write hit. In the writethrough scheme the AMD-640 system controller writes data to DRAM immediately when a cache line is modified. In the writeback scheme employing a modify bit, the AMD-640 system controller merely sets the modify bit of the altered cache line when a line is modified.

On a PCI cycle, the AMD-640 system controller snoops the processor's L1 cache. If it contains the desired PCI data and it has been modified, the cache line must be written back. However, the writeback forwarding feature allows the PCI initiator to read the cache data before the writeback takes place. Processor writeback cycles are handled as normal processor write cycles.

Table 5-3 shows the actions taken by the AMD-640 system controller on a cache hit cycle.

**Table 5-3. Cache Hit Action Taken**

| Cycle Type | Action Taken |
|---|---|
| Processor read | 1. Data (all four bytes) are read from cache.<br>2. Cache data, tag, and modify bits are unchanged. |
| Processor write | 1. Data with active byte enables are written to the cache.<br>2. The tag is unchanged.<br>3. The modify bit is set (writeback/modify bit scheme only).<br>4. The data is also written to DRAM (writethrough scheme only). |
| PCI read | 1. The processor is snooped to write back modified internal cache line.<br>2. Data (all four bytes) are read from cache.<br>3. Cache data, tag, and modify bits are unchanged. |
| PCI write | 1. The processor is snooped to write the back modified internal cache line.<br>2. Data with active byte enables are written to the cache.<br>3. The tag is unchanged.<br>4. The modify bit is set (writeback/modify bit scheme only).<br>5. The data is also written to DRAM (writethrough scheme only). |

## Cache Misses

Table 5-4 shows the action taken on a cache miss cycle. The action taken during a cache write miss cycle is identical for most cache schemes, but varies for different schemes on a read miss cycle. A cache line is allocated on a read miss only, not on a write miss.

With a writethrough cache, no writeback action is required because DRAM-cache coherency is always maintained. With a writeback cache, the existing cache line must be written back to DRAM if its modify bit is set. If no modify bit is employed, the line is assumed to be modified and writeback action is required unless the line is in a write-protected region.

Processor writeback cycles are handled as normal processor write cycles. On a PCI cycle, the AMD-640 system controller snoops the processor's L1 cache. If it contains the desired PCI data and it has been modified, the cache line must be written back. Writeback forwarding allows the PCI to read the modified data before it is written back.

**Table 5-4. Cache Miss Action Taken**

| Cycle Type | Action Taken |
|---|---|
| Processor read | 1. The line currently in the cache is written back to DRAM if no modify bit is used or the modify bit is set.<br>2. The entire data line is read from DRAM and written to the cache.<br>3. The tag is updated.<br>4. The modify bit is reset (writeback/modify bit scheme only).<br>5. The requested data is returned to the processor. |
| Processor write | 1. The data is written to DRAM.<br>2. Cache data, tag, and modify bits are unchanged. |
| PCI read | 1. The processor is snooped to write back the modified internal cache line.<br>2. Data (all four bytes) are read from DRAM.<br>3. Cache data, tag, and modify bits are unchanged. |
| PCI write | 1. The processor is snooped to write back the modified internal cache line.<br>2. Data is written to the cache.<br>3. Cache data, tag, and modify bits are unchanged. |

## Protocol

To simplify system design, the AMD-640 system controller uses only one cache control bit (the modify bit) rather than the two bits employed in the MESI (modified, exclusive, shared, invalid) protocol. In writeback mode there are only three cache states—invalid, valid, and modified. The modify bit indicates whether the cache line is valid (cleared) or modified (set), except when all active tag lines are set, which indicates the invalid state. Table 5-5 summarizes bit conditions for the various cache states.

**Table 5-5. Cache States vs. Bit Conditions**

| Cache State | Modify Bit | Other Tag Lines |
|---|---|---|
| Valid | 0 | Not all 1's |
| Modified | 1 | Not all 1's |
| Invalid | x | All 1's |

Figure 5-4 shows how cache state transitions occur.

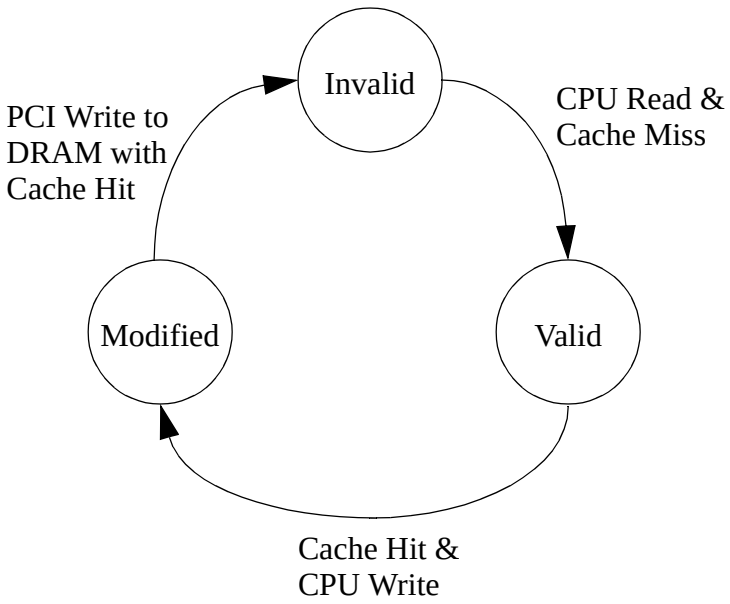



**Figure 5-4. Cache State Transitions**

## Operating Modes

The cache controller has three operating modes: enabled, disabled, and initialization. In the enabled mode, the cache controller functions normally. In the disabled mode, all read and write cycles are passed to the DRAM controller with no change to the cache data and tag bits.

The initialization mode puts the cache into a defined state after power-up. The BIOS puts the cache controller into the initialization state by writing 01 to offset 50h, bits 7-6, then reads from memory to fill the cache with valid data. The reads should start at xx00000 (xx depends on the cache size) and end at the cache limit. This requirement forces all lines to be valid. Finally, the BIOS enables the cache by clearing bits 7-6.

The BIOS normally performs this initialization before the cache is enabled. Any software that tests this feature after the cache has been enabled should reside in non-cacheable memory to prevent a system crash.

Figure 5-5 shows a burst read from PBSRAM. The assertion of COE# indicates an L2 cache hit. CADV# allows the address to increment on each clock. The completion of the data G access is not shown.

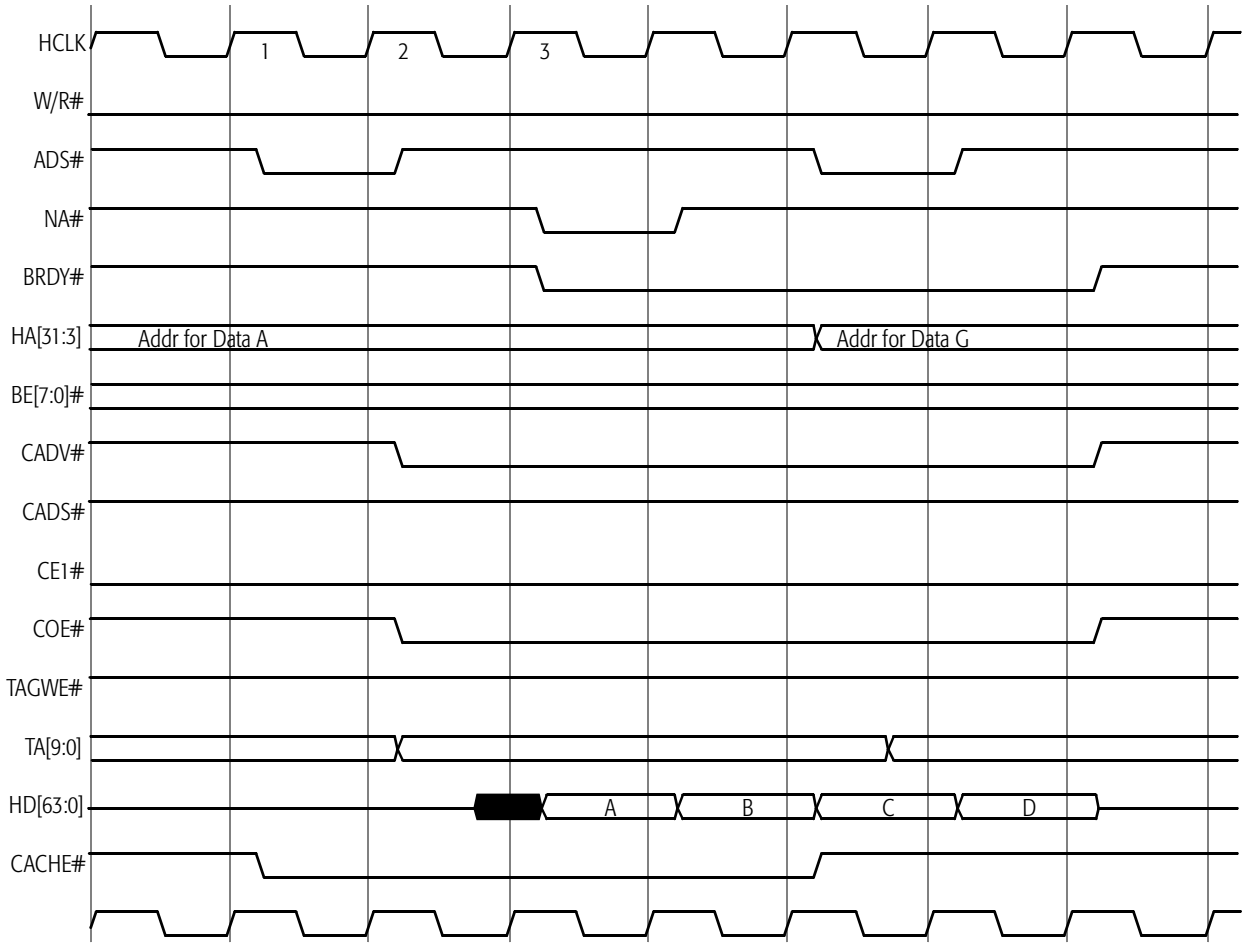



**Figure 5-5. Pipelined Burst Read Cycle**

Figure 5-6 shows a burst write to PBSRAM. Note that CADV#, which allows the address to increment, is high for one clock following ADS#. The completion of the data G access is not shown

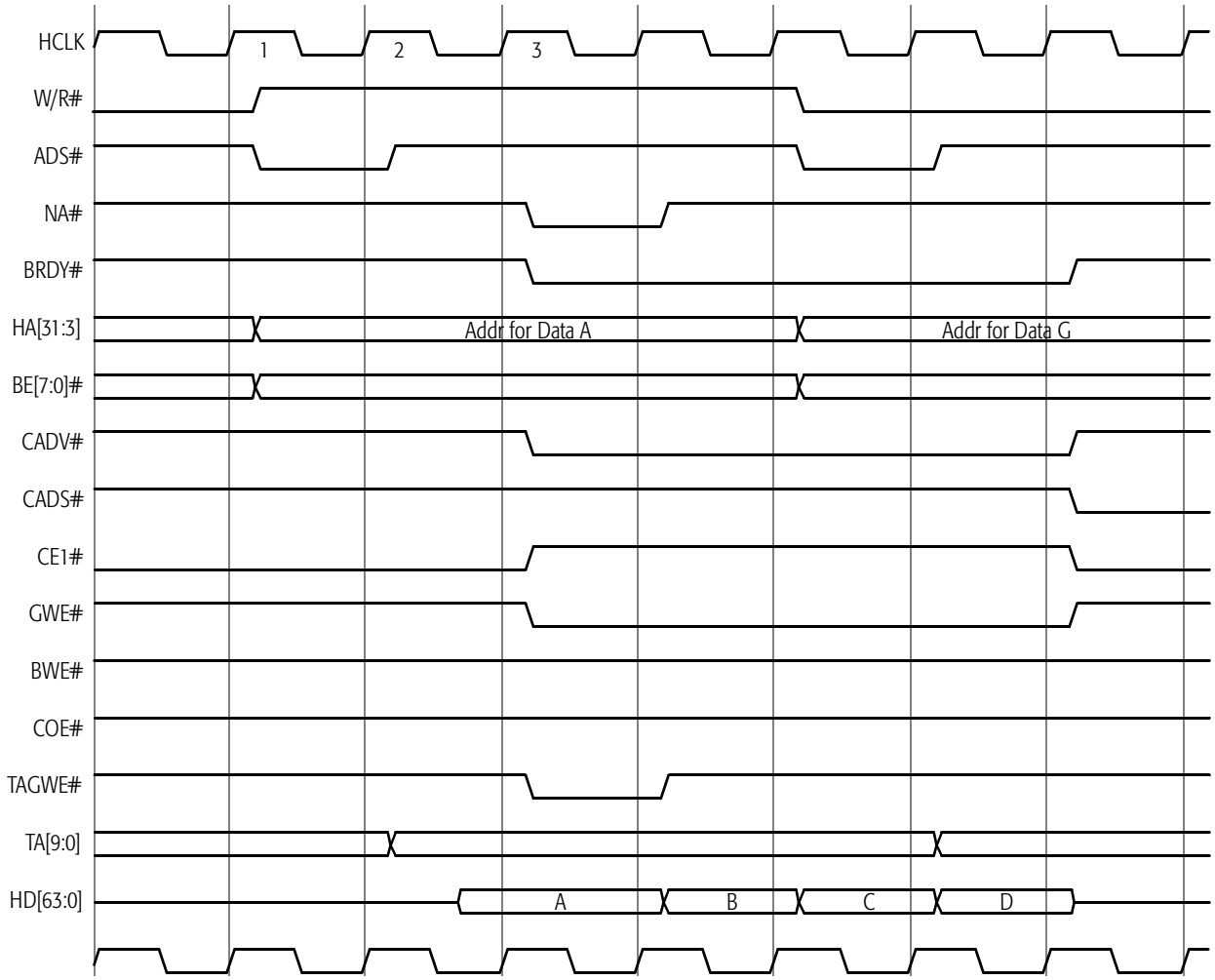



**Figure 5-6. Pipelined Burst Write Cycle**

### 5.2.3 Write Buffers

The AMD-640 system controller includes CPU-to-DRAM and PCI-to-DRAM write buffers to improve performance during cache read and write miss cycles.

On a cache read hit to a modified line, the buffers allow subsequent cache lines to be read while the altered line is written back to DRAM. On cache write misses, the AMD-640 system controller asserts the BRDY# line, enabling the processor to start the next cycle while the buffered data is written to DRAM.

The AMD-640 system controller also allows reads to bypass pending writes (see Section 5.1.3, page 5-5).

### 5.2.4 Cacheable Region

Only DRAM attached to the AMD-640 system controller is cacheable. The cacheable region is further limited by the following factors:

- The size of the DRAM and cache
- The number of tag lines enabled by the Cache Control 1 configuration register, offset 50h (page 7-11)
- The settings in the Non-Cacheable Region configuration registers, offsets 54h–57h (page 7-15)
- The cacheability of video and system BIOS as determined by the Shadow Ram Control configuration registers, offsets 61h–63h (page 7-21)

The normal cacheable region is the lesser of the DRAM size and 256 times the cache size (512 or 1024 if 9 or 10 tag bits, respectively, are used). The normal cacheable region is decoded automatically and does not require setting any configuration registers.

Within the normal cacheable region, two noncacheable areas can be specified by the Non-Cacheable Region configuration registers (page 7-15).

The upper memory region (A0000h to FFFFFh) is noncacheable by default because it corresponds to the memory-mapped I/O ports. However, the video and system

BIOS (C0000h-C7FFFh and E0000h-EFFFFh, respectively) can be made cacheable and write-protected by programming the Shadow Ram Control configuration registers.

The KEN# and EADS# signals maintain consistency in the cacheable region between the L1 and L2 caches. KEN# alerts the processor if data being read is cacheable. The EADS# signal, which the AMD-640 system controller uses to snoop the L1 cache on PCI-DRAM cycles, is only asserted for data in the cacheable region.

### 5.2.5 Cache Parameters

Data transactions with the L2 cache SRAM may differ depending on the type of SRAM selected. This variance implies that a specific SRAM speed may be required for different bus speeds. Examples of SRAM used for various bus speeds are shown in Table 5-6.

**Table 5-6. SRAMs vs. Bus Speeds**

| Type of SRAM | 60 Mhz | 66Mhz | Wait States |
|---|---|---|---|
| Asynchronous | 6.7 ns | 5 ns | 0 |
| Synchronous | 15 ns | 12 ns | 0 |
| Sync Pipelined Burst | | 66 MHz | 0 |

### 5.2.6 Cache Snooping

**Snoop Filtering**

Snoop filtering increases processor bandwidth by reducing the number of snoop cycles (also called inquire cycles) on the local bus. When a PCI cycle causes a snoop, the AMD-640 system controller retains the number of the cache line. If a subsequent access addresses the same line, no snoop cycle is generated.

**Snoop Ahead Read**

This feature prevents stalling a PCI burst transfer to fetch data from DRAM. In a PCI read cycle, the AMD-640 system controller snoops the cache and reads data from there rather than DRAM if it is present. The controller would then commence a PCI burst cycle. However, if the next data line were not in the cache, the controller would have to stall the burst in order to fetch the next line from DRAM. With snoop

ahead read, the controller looks at the next cache line before starting the PCI burst, thus avoiding this potential loss of time.

## 5.3 DRAM Controller

The AMD-640 system controller supports up to six 64-bit banks of DRAM with a capacity of up to 768 Mbytes. Each bank can contain 1-, 2-, 4-, or 16-Mbit by 32- or 64-bit DRAMs, in any combination of FPM, EDO, or SDRAM. FPM and EDO DRAMs can be 72-pin SIMMs with either 36 bits if Error Correcting Code (ECC) is required, or 32 bits (no ECC). SDRAM can be 168-pin DIMMs with either 72 bits (ECC) or 64 bits (no ECC).

Bank 1 is enabled by RAS0#, Bank 2 is enabled by RAS1#, and so on. Single-banked memory modules require one RAS# signal and occupy a single row. Dual-banked memory modules occupy two rows of memory and require two RAS# signals.

The DRAM banks are grouped into three pairs. Each pair can have zero, one, or both banks populated. The only constraint is that if both banks in a pair are populated they must be of the same type (size, SIMM configuration, and mode).

All of the DRAM parameters are programmed in configuration registers 58h–6Fh. See pages 7-16 through 7-31.

### 5.3.1 Mixing Memory

The AMD-640 system controller can accommodate different memory sizes or types in different banks, but not within the same bank or bank pair. Configuration registers 58h–6Fh are used to program the ending address, column address size, DRAM type, error correction, timing, refresh interval, drive strength, and width for each bank.

Rules for populating DRAM are as follows:

1. Pairs must be of the same type (FPM or EDO DRAM).
2. If 64-bit mode is used, the banks must be paired. They need not be paired in 32-bit mode.
3. They must be populated in order. i.e., 0,1,2,3,4,5.

Figure 5-7 shows how EDO DRAM connects to the AMD-640 system controller. Figure 5-8 shows connections to SDRAM.

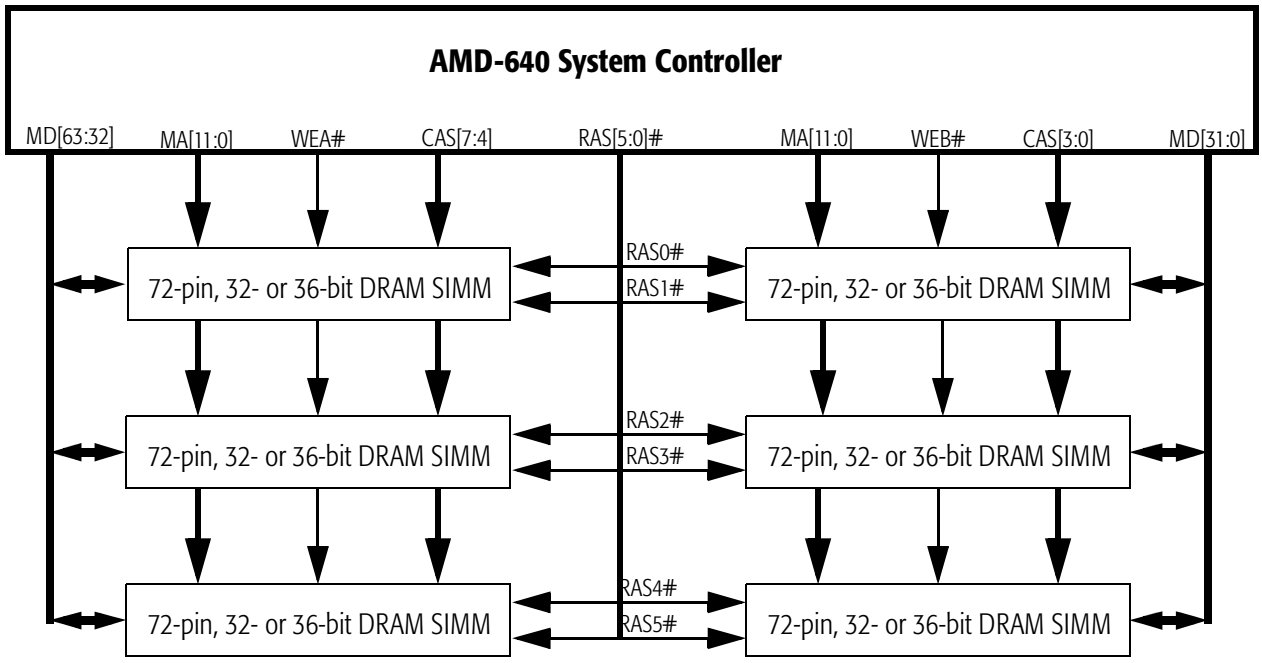



**Figure 5-7. EDO DRAM Interface Example**

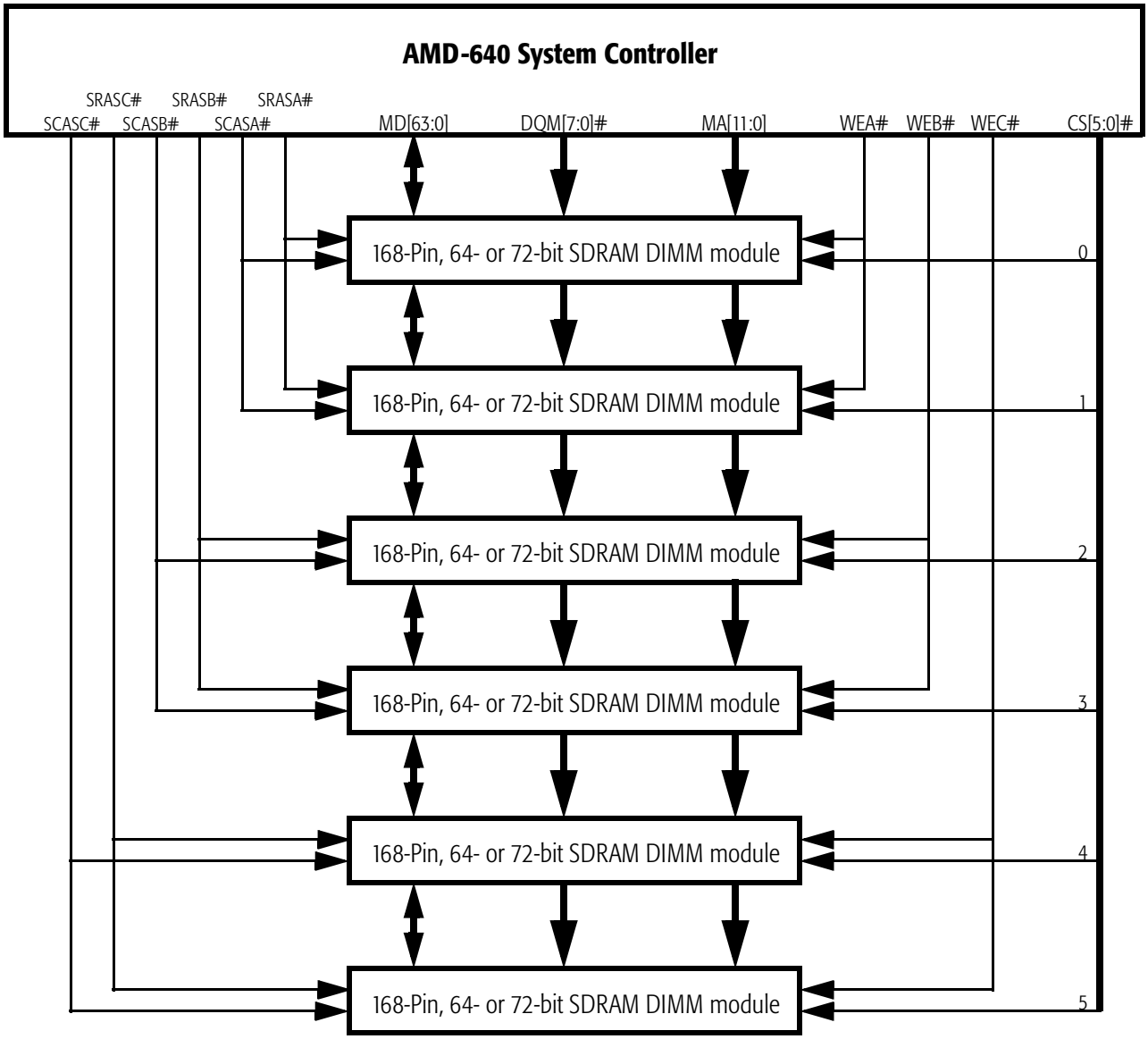



**Figure 5-8. SDRAM Interface Example**

**Memory Detection**

The AMD-640 system controller can accommodate different memory sizes or types in different banks, but not within the same bank or bank pair. A software or firmware mechanism can be integrated into the BIOS that automatically detects the type and size of the DRAM device in each bank. The mechanism sets the Last Bank Populated to Bank 0 (offset 59h bits 2-0, page 7-19), determines its type and size, sets the Bank 0 ending address (offset 5Ah, page 7-19), increments the Last Bank Populated register, and performs the test on bank 1, making sure the address range for bank 1 is above the range determined for bank 0. The cycle is repeated for all populated banks.

To determine the type of device, the mechanism configures the target DRAM bank as EDO, enables it, writes data to it, reads the data back, and compares the results. A match indicates the presence of EDO DRAM, because standard DRAM does not respond properly to the faster EDO access cycles. If the comparison fails, the mechanism configures the bank as FPM DRAM and performs a similar test. If this test also fails, a somewhat more complex test can be run to determine if SDRAM is present or a bank is empty. Refer to the BIOS guide for more details. This procedure should be performed on eight consecutive bytes to determine if different types of memory devices are installed within a row. Any row containing differing memory types should be disabled.

To determine the size of the device, the mechanism sets the start address either at 0 or somewhere in the upper memory area. Then, using the memory column size bits in the corresponding configuration register (offsets 58h or 59h), the mechanism selects the largest size and tests the memory at all possible boundaries.

## 5.3.2 Error Correction Code

The AMD-640 system controller supports error correction code (ECC) to check the integrity of transactions with system memory. ECC, also referred to as Hamming code, corrects single-bit and double-bit errors as well as some triple-bit errors. ECC is enabled in offset 6Eh (page 7-30). The memory modules must have parity bits to implement ECC.

ECC operation requires that system memory be initialized. In this procedure, the BIOS writes to every memory location, generating valid ECC that is stored in the DRAM parity bits. If this procedure is not performed, errors will occur when writing data smaller than a 64-bit doubleword.

Memory types cannot be mixed when ECC is employed. All memory must be of the same type—SDRAM, FPM, or EDO. Also, some timing delay should be added to allow for the delay added by the ECC logic. Systems using SDRAM, should set offset 6Eh, bit 3 (see page 7-30). With EDO or FPM DRAM, the CAS# width in offset 64h, bits 3–2 should be programmed 1T more than normal (see page 7-23). In some cases, the half-cycle delay added by setting offset 65h, bit 3 (see page 7-24) is sufficient.

### 5.3.3 DRAM Refresh

The AMD-640 system controller provides DRAM refresh that is transparent to the rest of the system. Normal, burst, or CAS-before-RAS (CBR) refresh can be selected through offset 6Bh (page 7-27). Accesses to the read and posted write buffers are allowed during a refresh period. RAS# pulses to the six memory banks are staggered one HCLK apart to minimize switching noise during memory refresh, as shown in Figure 5-9.

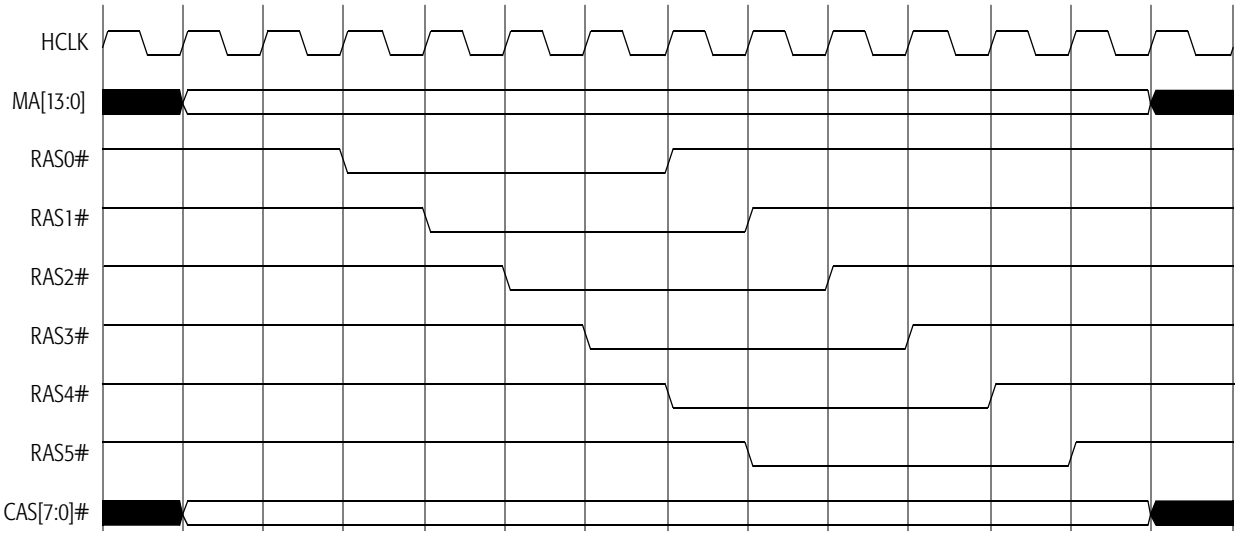



**Figure 5-9. Refresh Timing**

The AMD-640 system controller also contains a refresh counter that provides 4096 refresh cycles on MA[11:0]. This permits the use of DRAMs up to 16 Mbits in size. The refresh period is derived by dividing HCLK by 16 (four-bit prescale) and a refresh divisor based on the 8-bit value in offset 6Ah. The refresh divisor can be calculated by multiplying the DRAM's required refresh period by the prescaled clock rate. For example, in a system with HCLK = 66 MHz and 4-Mbit DRAM requiring a refresh interval of 1024 cycles every 16 ms, the refresh divisor would be calculated as follows:

Refresh divisor = refresh period * prescaled clock

= $(16x10^{-3}$sec/1024 cycles) * (66 x $10^6$ /16)
= 64.5 (decimal)
= 42h (hexadecimal)

**Page Mode DRAM**

The AMD-640 system controller generates DRAM addresses based on the processor address and type of DRAM. Row and column addresses are multiplexed on the same MA bus. For non-page mode operations and page misses, the AMD-640 system controller sequentially generates a row address and column address. On page hits, only a column address is generated during the DRAM access.

DRAM cycles normally operate in page mode. In this mode, RAS# is held active after a DRAM access has finished in anticipation of the next access. RAS# is brought high to precharge the DRAM only when a subsequent cycle to the same bank accesses a different DRAM page or an asynchronous event such as a RAS# time-out occurs.

With Fast Page Mode DRAMs, the column address is latched on the falling edge of CAS#. FPM DRAMs require CAS# to stay active throughout the entire cycle, because their drives turn off when CAS# goes high. While a page cycle continues within a row, RAS# remains active while CAS# is toggled as the address (column) changes. Fast Page Mode DRAMs are enabled or disabled for each bank pair in offset 60h (page 7-24).

DRAM cycles for all processor accesses are generated synchronously with the CPU clock (HCLK). Critical DRAM timing parameters including RAS# precharge time and pulse width, CAS# and write pulse widths, and column address-to-

CAS# delay can be individually programmed in configuration register 64h.

### 5.3.4 Shadow RAM

The AMD-640 system controller supports shadowing of system, video, and other BIOS functions to accelerate access. The BIOS normally resides in Read Only Memory (ROM) to prevent altering the content of this crucial system code. Because ROM is substantially slower than RAM, most systems provide for copying the ROM contents to the upper memory area of RAM and making that area read only. The portion of RAM containing the BIOS copy is referred to as Shadow RAM.

The AMD-640 system controller provides three control registers (offset 61h–63h, starting on page 7-21) to select the portions of upper memory for shadowing and to control read/write access to those areas. The granularity is 16 Kbytes in the address range C0000h–DFFFFh and 64 Kbytes in the address range E0000h–FFFFFh. Read and write access can be enabled independently in each region. Further system performance enhancement can be achieved by programming address ranges C0000h–C7FFF and E0000h–FFFFFh to be cacheable and write-protected.

To copy the ROM code into the targeted shadow memory, set the access control bits for that area to Write Only, then copy the ROM code (read address) to the shadow memory (write address) with the source and destination pointed to the same physical address. After completion of the copy process, adjust the access control bits to Read Only.

If shadow RAM is not enabled, addresses A0000h–FFFFFh can be relocated to the top of local DRAM, increasing memory size by 384 Kbytes. If only the C segment (C0000h–CFFFFh) and/or the F segments are used for shadowing, memory is increased by 256 Kbytes. No DRAM can be relocated if either the D or E segment is used for shadow memory. Addresses A0000h–BFFFFh can be reserved for the system management memory map by setting register 63h bit 1 and cannot be relocated again. In this case, memory increases are reduced to 256 Kbytes (no shadowing) and 128 Kbytes (C and/or F segments used for shadowing).

### 5.3.5 EDO DRAM

EDO DRAM can increase system speed because most EDO accesses take one clock cycle less than standard FPM devices. The AMD-640 system controller generates the appropriate clock cycles for FPM or EDO based on the information in configuration registers 60h and 64h. EDO memory allows shorter page cycle times by keeping the output drivers on when CAS# goes inactive. (Fast-page DRAMs require CAS# to stay active throughout the entire cycle.) The basic characteristics of EDO memory are as follows:

- The column-address is latched when CAS# falls.
- The output drivers remain on when CAS# goes high, and only turn off when both CAS# and RAS# are deasserted.
- Data is valid until either the next falling edge of CAS# or the next rising edge of OE#.

Figure 5-10 shows a pipelined burst read from EDO memory, as indicated by the assertion of NA#. The first access is 5-2-2-2 and the second is 3-2-2-2. The five clock cycles in the first access indicate a page hit. On a page miss the first access is 11 clock cycles to allow for precharging the row and strobing the new row address into the controller.

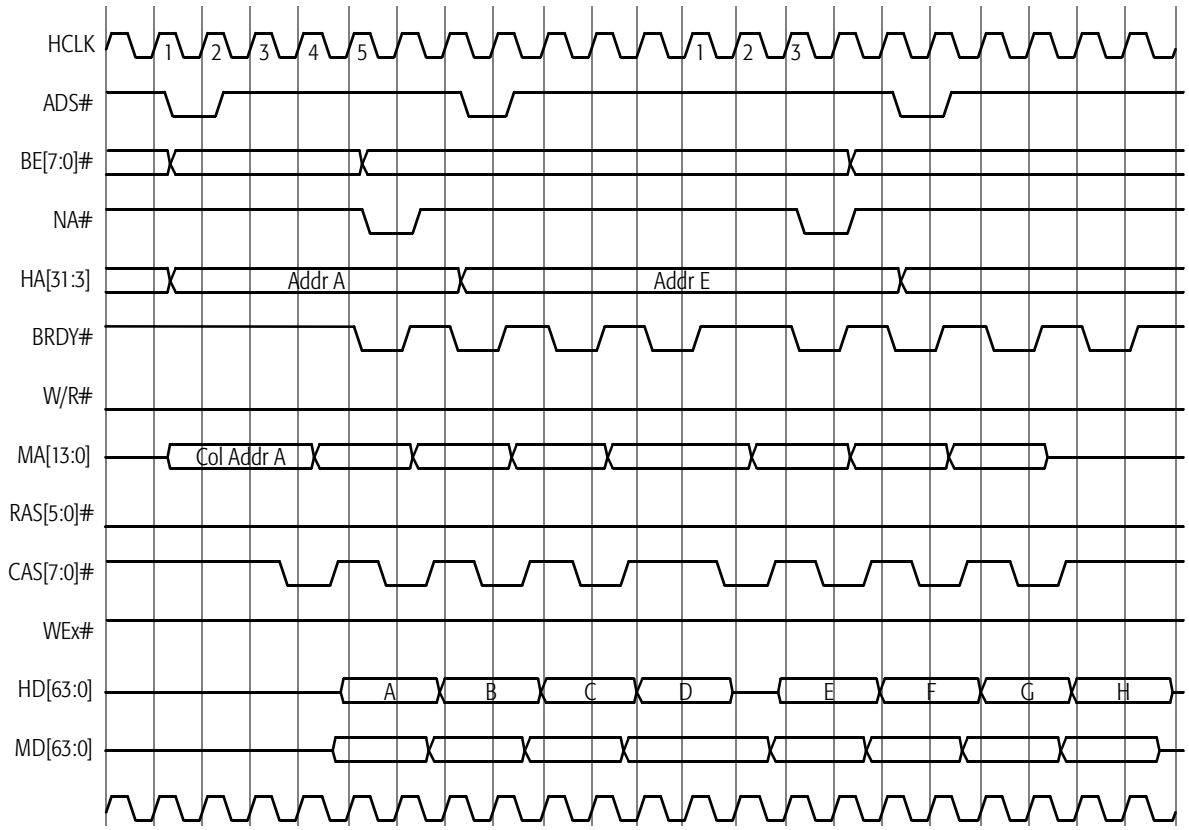



**Figure 5-10. Pipelined EDO Read (5-2-2-2, 3-2-2-2)**

Figure 5-11 shows a write to address A followed by a pipelined read request from address G. The pipelined read is not shown. The write from the processor to the write buffer is a 3-1-1-1 cycle. The transfer from the buffer to DRAM does not begin until the entire line is written to the write buffer. The write buffer allows the processor to continue processing much earlier than if the processor was required to wait for the write to DRAM to complete.

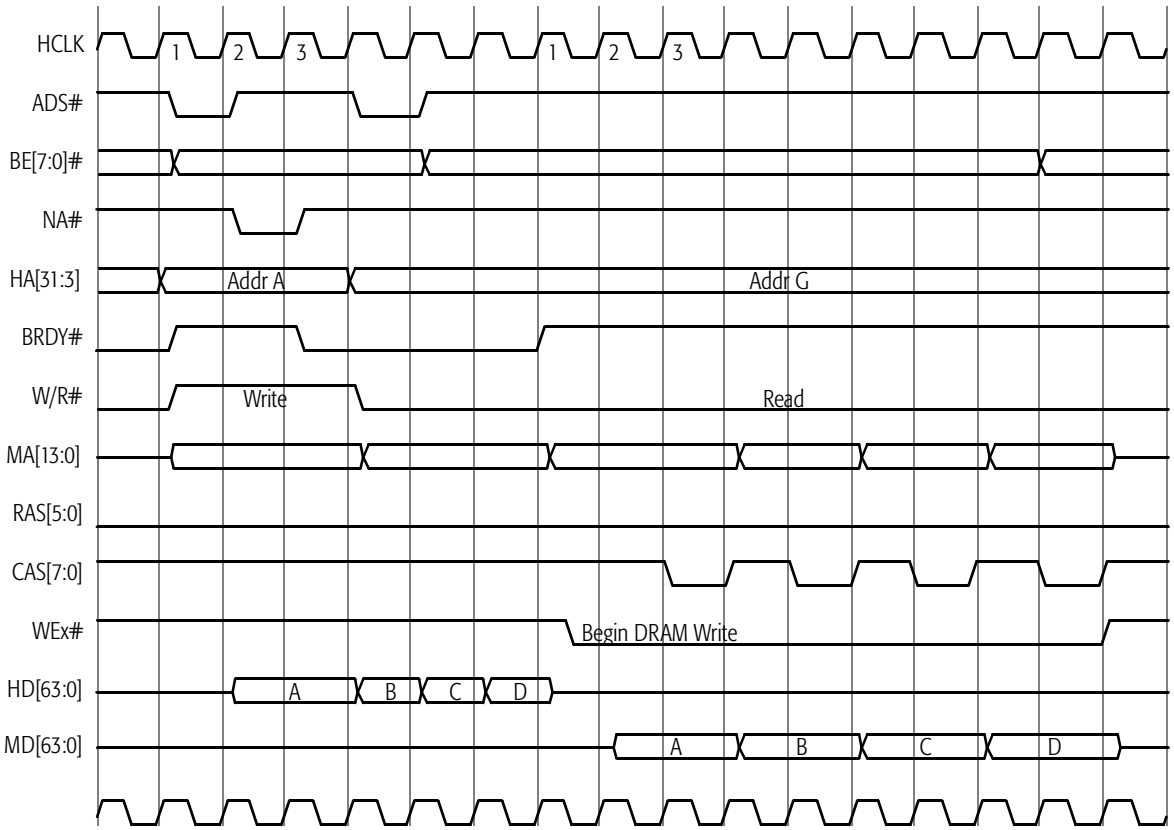



**Figure 5-11. EDO Posted Write (2-2-2-2)**

### 5.3.6 Synchronous DRAM

Synchronous DRAM is the most recent innovation in the evolution of main memory systems to meet the higher bandwidth needs of today's processors. SDRAMs use a clock to synchronize address and data rather than row and column strobes. They can also be programmed to select the burst length, write mode, and type of burst (sequential or linear). The net effect is to achieve performance approaching SRAM. SDRAM is rated by operating frequency rather than access time. Currently, SDRAMs are available in 66-MHz, 83-MHz, and 100-MHz speeds.

SDRAM memory does not toggle CAS# to get new data, but simply increments a counter to supply the address for succeeding cycles, thus substantially reducing bus delays. The basic characteristics of SDRAM are as follows:

- The clock is enabled when either RAS# or CAS# is first sampled asserted.
- The output drivers remain on when CAS# is negated. They are turned off when WEx# or CS# goes high.
- Read data is valid until the next rising clock edge.
- Write data is sampled on each rising clock edge.
- DQM[7:0]# determine which bytes are read or written.
- Control signals need only be valid during CS#.

The BIOS configures the Memory Controller for SDRAM memory operation for each bank of memory by programming offset 60.

Figure 5-12 shows an SDRAM burst read, followed by a single read to address G, then a read access to address X in the other bank. The bank change is indicated by MA11 changing from low to high.

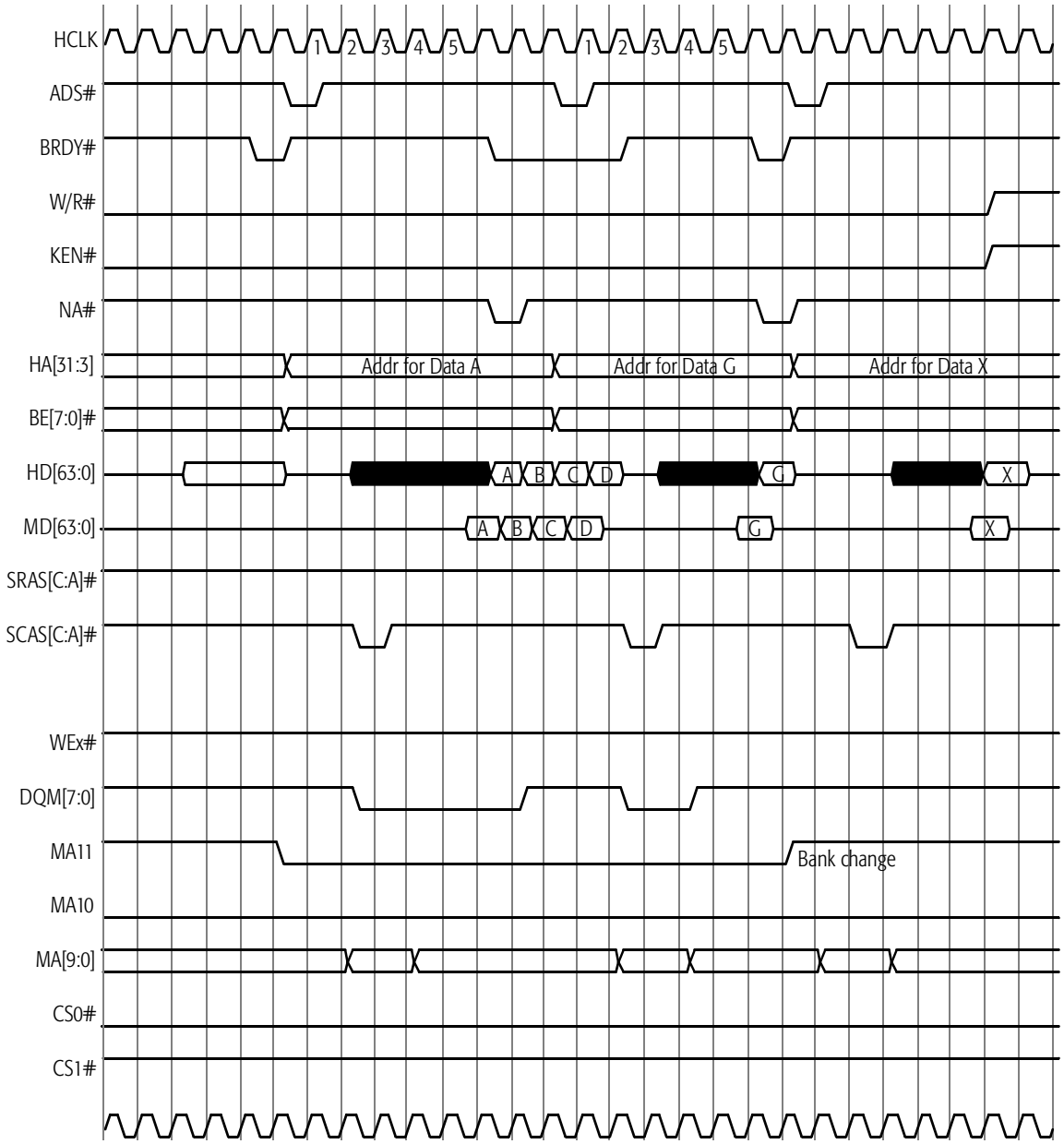



**Figure 5-12. SDRAM Burst Read Cycle**

Figure 5-13 shows an SDRAM burst write, followed by a single write to address G, then a write access to address J in the other bank. The bank change is indicated by MA11 changing from low to high.

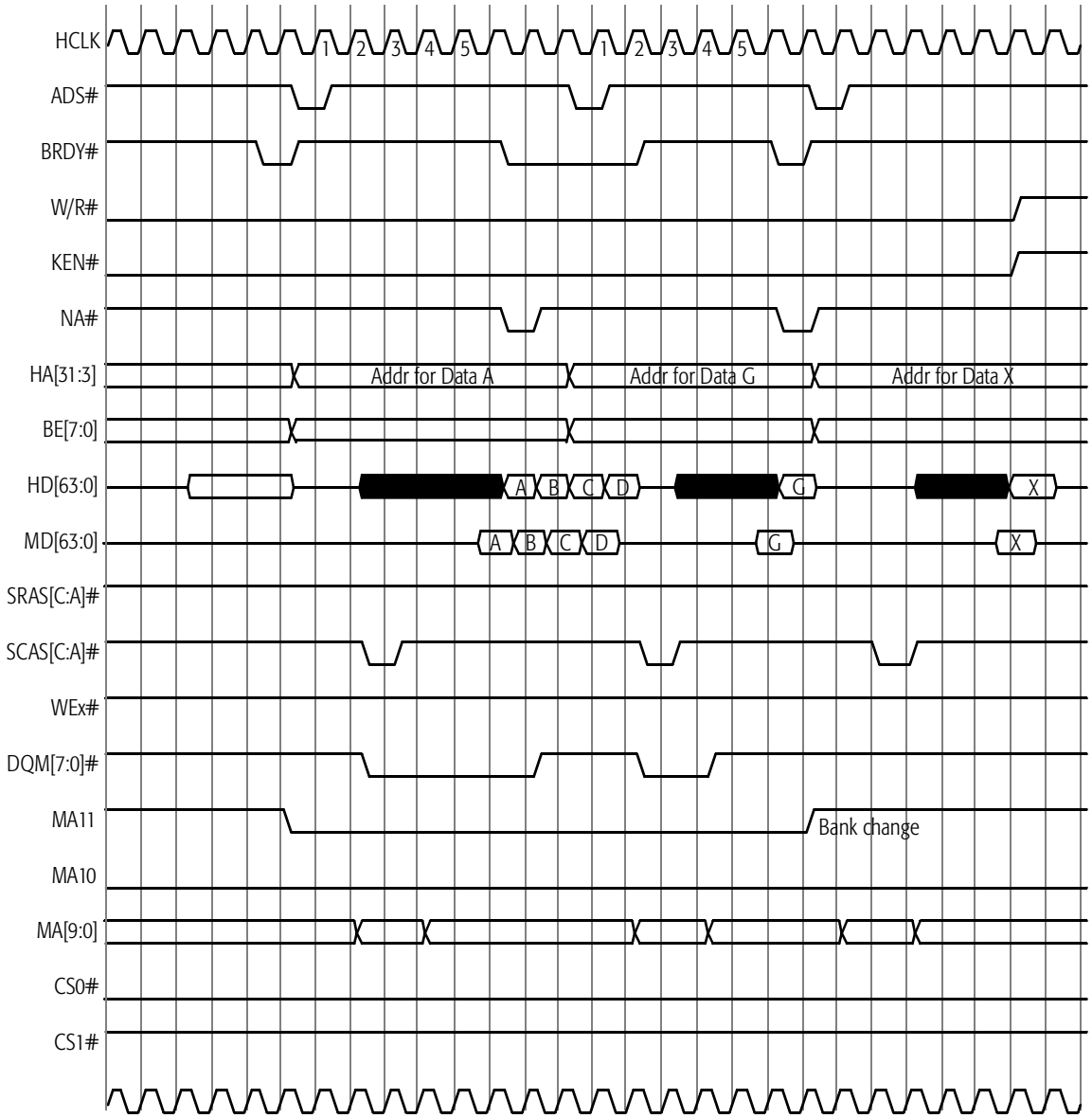



**Figure 5-13. SDRAM Write Cycle**

Figure 5-14 shows a CPU read miss. As the processor reads data from DRAM, the cache controller captures the data and stores it in the L2 cache, updating the tag to reflect a new line. Note the first GWE# is wider, allowing Data A and B to be written sequentially.

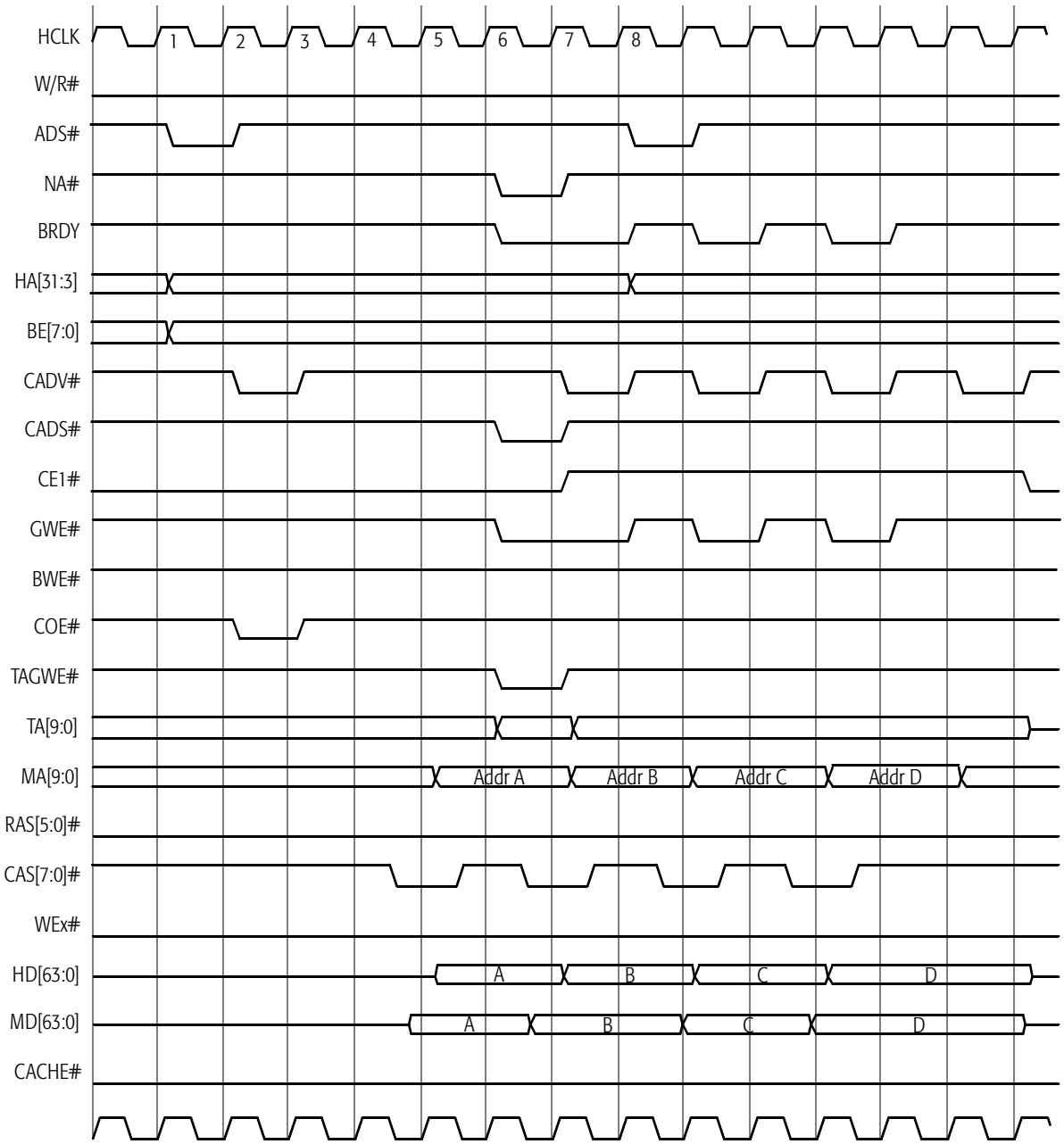



**Figure 5-14. CPU Read Miss**

Figure 5-15 shows a read miss with a modified L2 cache line, which must be written back. During the initial read, the cache data (I, J, K, L) is written into the DRAM write buffer. Next, the processor reads data from the DRAM (A, B, C, D). The cache controller captures the data as it passes to the processor and writes it in the L2 cache, updating the tag to reflect the new line. Finally, the data in the write buffer (G, H, I, J) is written to DRAM.

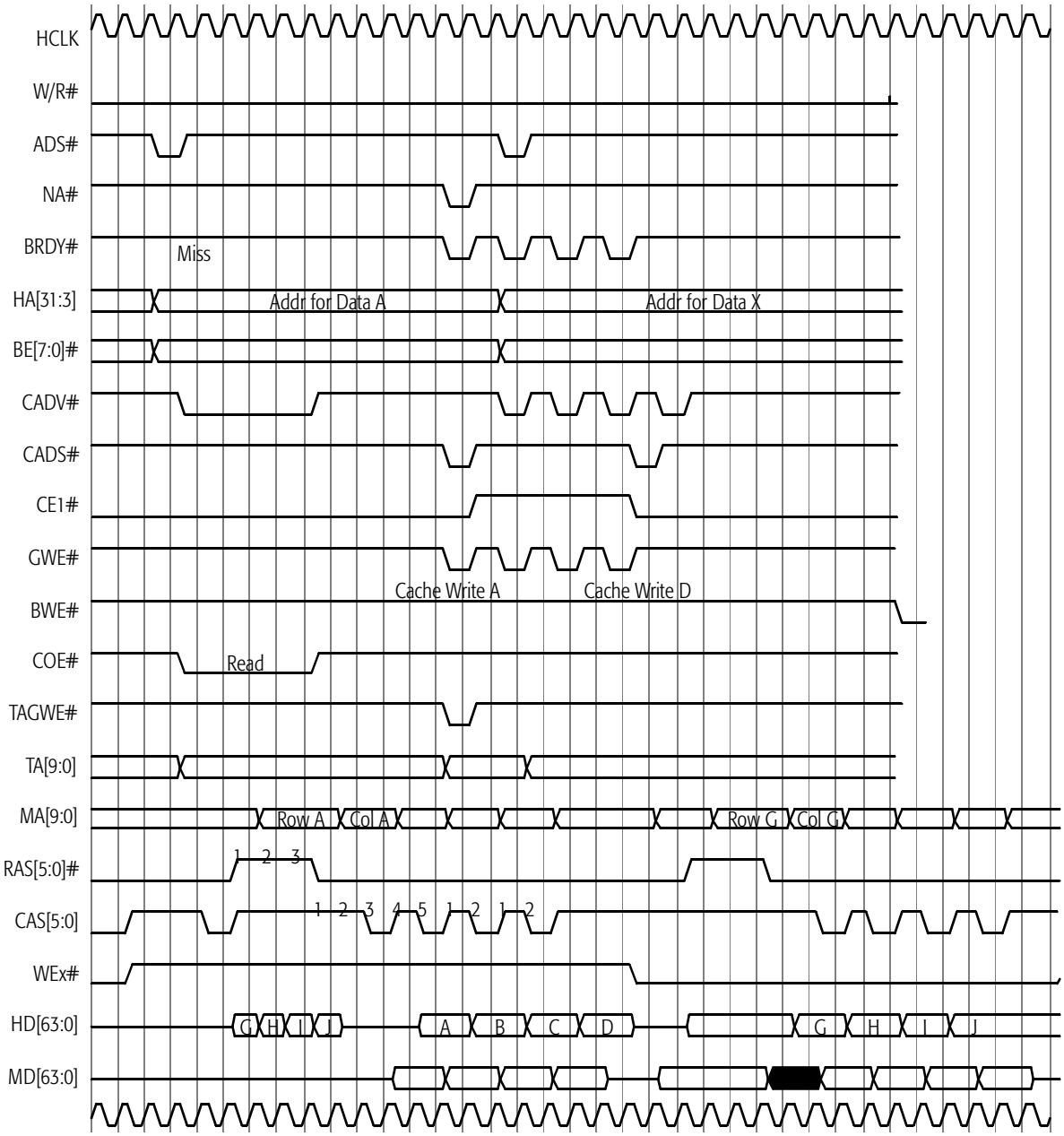



**Figure 5-15. Read Miss With Modified L2 Cache Line**

# 5.4 PCI Bus Controller

The AMD-640 system controller drives the 32-bit PCI bus synchronously with the PCI clock (PCLK), which is a buffered processor clock (HCLK) divided by two. It converts 64-bit processor data to 32-bit PCI data and regenerates commands with minimum overhead. A five-doubleword CPU-to-PCI post write buffer enables the processor and PCI to operate concurrently. The AMD-640 system controller converts consecutive processor addresses to burst PCI cycles, employing byte merging for optimal CPU-to-PCI throughput. Its unique integration of PCI controller and DRAM controller functions on one chip provides a fast 32-bit data link, crucial in achieving zero-wait state buffer movement and sophisticated, upgradeable buffer management schemes such as byte merging. A 48-doubleword PCI-to-DRAM post write buffer and a 26-doubleword DRAM-to-PCI prefetch buffer enable concurrent PCI bus and DRAM/cache accesses during PCI initiator transactions. 2-1-1-1 cache hit and 3-1-1-1 cache miss timing provide a typical PCI bus initiator transfer rate of greater than 100 Mbytes per second.

When the processor drives an I/O cycle to an address other than the AMD-640 system controller's configuration register addresses, the controller passes the I/O cycle to the PCI bus. The AMD-640 system controller posts the I/O cycle in one of its write buffers. The controller does not respond to I/O cycles driven by PCI initiators on the PCI bus. It allows these cycles to complete on the PCI bus.

Transactions on the PCI bus consist of an address/control phase followed by one or more data phases. Three signals provide fundamental control of all PCI data transfers. FRAME# is asserted by the initiator to indicate the beginning and end of a transaction. IRDY# is asserted by the initiator to indicate that it is ready to complete the current data phase. TRDY# is asserted by the target to indicate that it is ready to complete the current data phase.

When FRAME# and IRDY# are both inactive, the PCI bus is idle. A transaction begins with an address phase, in which an initiator simultaneously asserts FRAME# and issues the address and bus command. The first data phase begins on the

following clock edge. Data is transferred between the initiator and target on each clock edge for which both IRDY# and TRDY# are asserted. Either the initiator or target can insert wait states by delaying the assertion of IRDY# or TRDY#, respectively.

### 5.4.1 PCI-to-CPU (Read) Transactions

The AMD-640 system controller contains an eight-byte read buffer which assembles two 32-bit PCI read cycles into one 64-bit quadword for the CPU data bus. The buffers are also used when any read crosses a 32-bit boundary. Aligned byte/word/dword processor reads are passed on to the PCI bus by the AMD-640 system controller as such. The read buffer is always enabled.

When the processor reads from the PCI bus, the AMD-640 system controller acts as a PCI initiator. The controller responds to the read with data from one of its internal buffers or with data obtained by performing a read operation on the PCI bus. Figure 5-16 depicts a PCI read initiated by the AMD-640 system controller. On the first PCLK of the read transaction the controller initiates the address phase by asserting FRAME#, driving the PCI bus command on BE[3:0]#, and driving the address on AD[31:0]. (FRAME# remains asserted until either the data phase for the last transaction begins or the cycle is preempted. Figure 5-16 depicts a single-transfer read, so FRAME# is only asserted for one PCLK.) On the second PCLK, the controller releases AD[31:0] in what is known as the turnaround phase, in which ownership of AD[31:0] changes from the initiator to the target device. The AMD-640 system controller also begins driving the byte enables on BE[3:0]# during the second PCLK to indicate which data paths will be used for the transfer, and asserts IRDY# to indicate it is ready to accept data. During the third PCLK, the target device asserts DEVSEL# to indicate that its address matches the one driven for the cycle and that it is ready to begin returning data. In addition to DEVSEL#, the target device drives the requested data on AD[31:0] and asserts TRDY# to indicate the data is available. On the rising edge of the fourth PCLK, the AMD-640 system controller samples IRDY#, TRDY#, and AD[31:0]. Since IRDY# and TRDY# are both sampled active, the system controller accepts the data on AD[31:0]. The controller either forwards this data on to the

processor or stores it in one of its read buffers. If the target device needs to insert wait states before returning data, it does so by delaying the assertion of TRDY#.

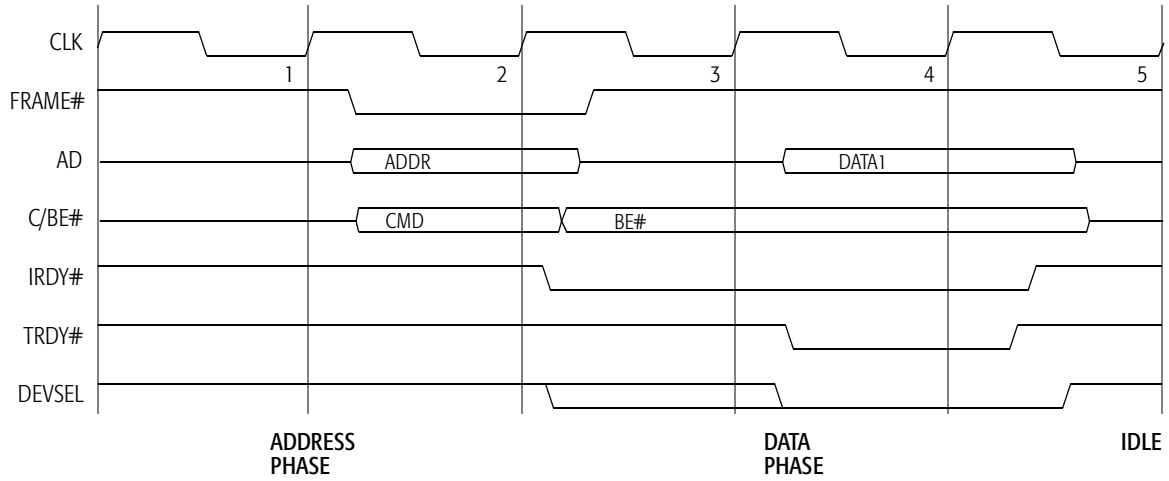



**Figure 5-16. Basic PCI Read Operation**

Figure 5-17 depicts a PCI burst read, which requires four data transfers, initiated by the AMD-640 system controller. In this example, the target inserts a wait state before the fourth data transfer by deasserting TRDY# for one PCLK and then reasserting TRDY# when it is ready to supply the data for the fourth transfer.

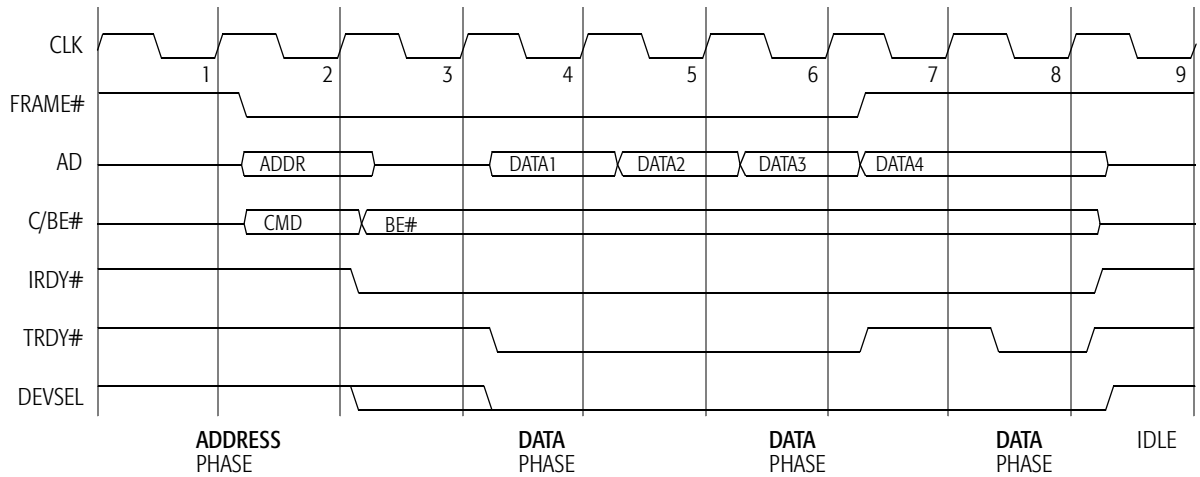



**Figure 5-17. PCI Burst Read Operation**

## 5.4.2 CPU-to-PCI (Write) Transactions

The AMD-640 system controller converts a full 64-bit (quadword) CPU-to-PCI write into two consecutive 32-bit (doubleword) PCI write cycles. It also features byte merging (grouping smaller, consecutive CPU writes into doublewords) and burst transactions (writing up to four doublewords in a single PCI transaction). These features in combination significantly reduce the bus bandwidth required to complete PCI writes.

The AMD-640 system controller contains a five-doubleword post write buffer between the processor and the PCI bus. Every CPU-to-PCI write is stored in the buffer unless it is full, allowing the processor to begin its next operation without having to wait for the write to complete. When the PCI bus is available, the AMD-640 system controller performs up to five 32-bit PCI writes to complete the transaction.

**Byte Merging**

Byte merging combines multiple CPU write cycles into a single PCI transfer. The AMD-640 system controller monitors address and byte enable signals to combine consecutive cycles containing 1, 2, and 4-byte writes into a single 8-byte buffer. The AMD-640 system controller does not allow non-contiguous byte merging. To merge bytes, the second write must be to a subsequent byte location in the 8 byte line. For example, if the first write is a byte write to byte location 3, only subsequent writes to byte locations 4-7 can be merged. If a write is made to locations 0-2, it will be posted to the next write buffer. In addition, the AMD-640 system controller does not allow re-ordering or over-writing merges. This is necessary to maintain support for strong write ordering, in which writes are placed on the PCI bus in the order they are received from the processor.

The AMD-640 system controller also supports byte merging for writes to the video/frame buffer area.

**Burst Cycles**

The AMD-640 system controller writes all of its buffer contents in a single PCI transaction when the bus becomes available. In this way, consecutive CPU-to-PCI writes, whether two full quadwords or several smaller transactions combined through byte merging, are performed in a single PCI transaction.

### Writes To PCI

When the processor writes to the PCI bus, the AMD-640 system controller acts as a PCI initiator. Figure 5-18 depicts a write to PCI initiated by the AMD-640 system controller. The controller drives FRAME#, AD[31:0], and BE[3:0]# to initiate the write transaction during the first PCLK. FRAME# remains asserted until the data phase for the last transaction begins or the cycle is preempted. (Figure 5-18 depicts a single transfer write, so FRAME# is only asserted for one PCLK.) AD[31:0] contains the address for the target of the write and BE[3:0]# contain the bus command (transaction type) information. During the second PCLK, the AMD-640 system controller begins driving the data for the write on AD[31:0] and the corresponding byte enables on BE[3:0]#. There is no need for a turnaround phase because the controller drives AD[31:0] during both the address and data phases. Also during the second PCLK, the AMD-640 system controller asserts IRDY# to indicate it is driving the data and is ready to complete the transaction. In this example the target is able to decode the address and drive DEVSEL# in the second clock to indicate that its address matches the one driven for the cycle, and it drives TRDY# to indicate it is ready to accept data. On the rising edge of the third PCLK, the target samples IRDY#, TRDY#, and AD[31:0]. Because both IRDY# and TRDY# are sampled active, the target accepts the data written on AD[31:0]. This is a zero- wait state write transaction. In most cases, the target device will require additional time to decode the address and complete the write. In this case, the target delays the assertion of DEVSEL#. If the target requires additional time to accept the data and complete the write, it delays the assertion of TRDY# as well.

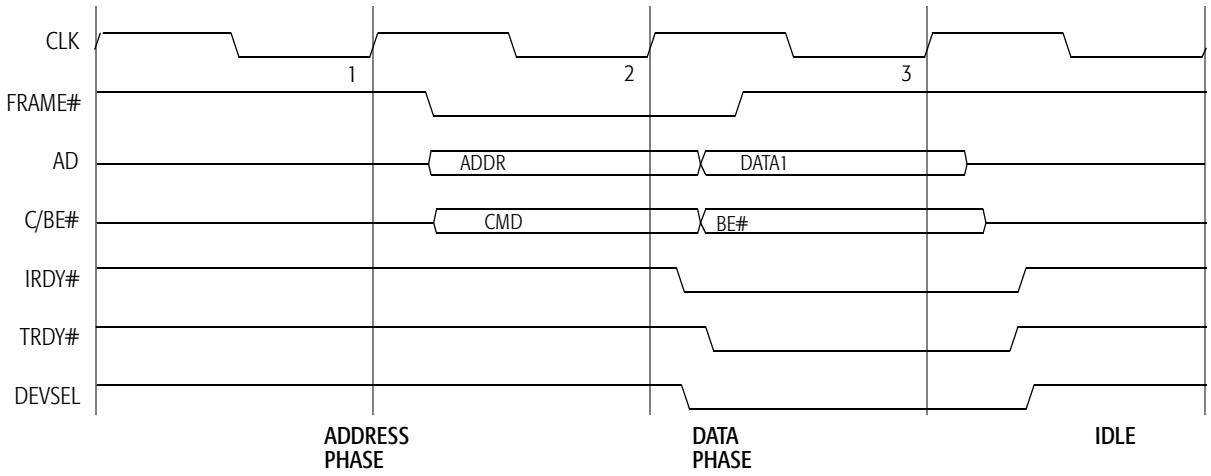



**Figure 5-18. PCI Write**

Figure 5-19 depicts a burst write with four data transfers on the PCI bus initiated by the AMD-640 system controller. This example also includes a wait state inserted by the target for both the first and the third data transfers. The target inserts the wait state by delaying the assertion of TRDY# for the first transfer. To insert a wait state in the third transfer, the target deasserts TRDY# for one PCLK, then reasserts TRDY# when it is ready to receive the third data transfer.

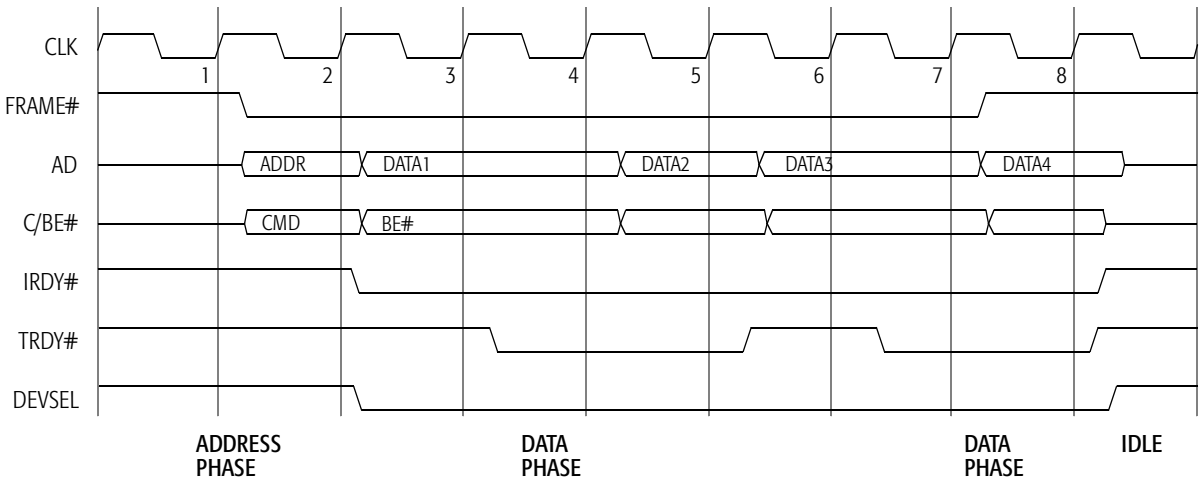



**Figure 5-19. PCI Burst Write**

### 5.4.3 PCI Arbitration

The AMD-640 system controller contains the arbitration logic that allocates ownership of the PCI bus among itself, the AMD-645 peripheral bus controller, and four other PCI initiators. For added flexibility, the AMD-640 system controller allows system designers to select several arbitration mechanisms. Two mechanisms are controlled by bit 7 in offset 75h (see page 7-39). These mechanisms can be disabled and replaced by four other choices in offset 76h (page 7-40). The adjustments include setting priority weight of the processor over other PCI arbiters, selecting REQ# or FRAME# as the trigger for new arbitration, and selecting the bus timeout period.

The PCI bus arbiter implements resource locking, which is selected via configuration register offset 73h, bit 1 (page 7-37).

When there are no requests for the bus, ownership defaults to the processor via the AMD-640 system controller. "Parking" the bus in this way is sometimes referred to as CPU-centric arbitration.

### 5.4.4 PCI Configuration

The AMD-640 system controller uses PCI configuration mechanism #1 to select all of the options available for interaction with the processor, DRAM, L2 cache, and the PCI bus. This mechanism is defined in the *PCI Local Bus Specification Revision 2.1* and described on page 7-1. All configuration functions for the AMD-640 system controller are performed via two I/O-mapped configuration registers, IO_CNTRL (I/O address 0CF8h) and IO_DATA (I/O address 0CFCh).

These two registers are used to access all other internal configuration registers of the AMD-640 system controller. The AMD-640 system controller decodes accesses to these two I/O addresses and handles them internally. A read to a non-existent configuration register returns a value of FFh. Accesses to all other I/O addresses are forwarded to the PCI bus as regular I/O cycles.

Read and write cycles involving the AMD-640 system controller configuration registers are functionally the same as other I/O read and write cycles. Both require five PCI clock cycles to complete. Configuration timing is illustrated in Figure 5-20 and Figure 5-21.

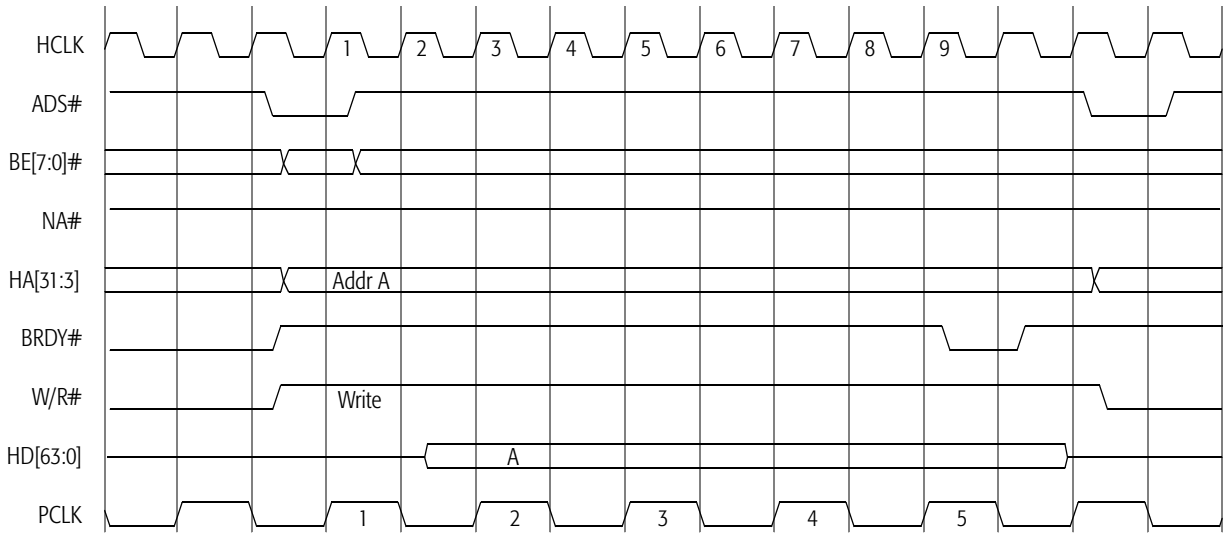



**Figure 5-20. Configuration Write**

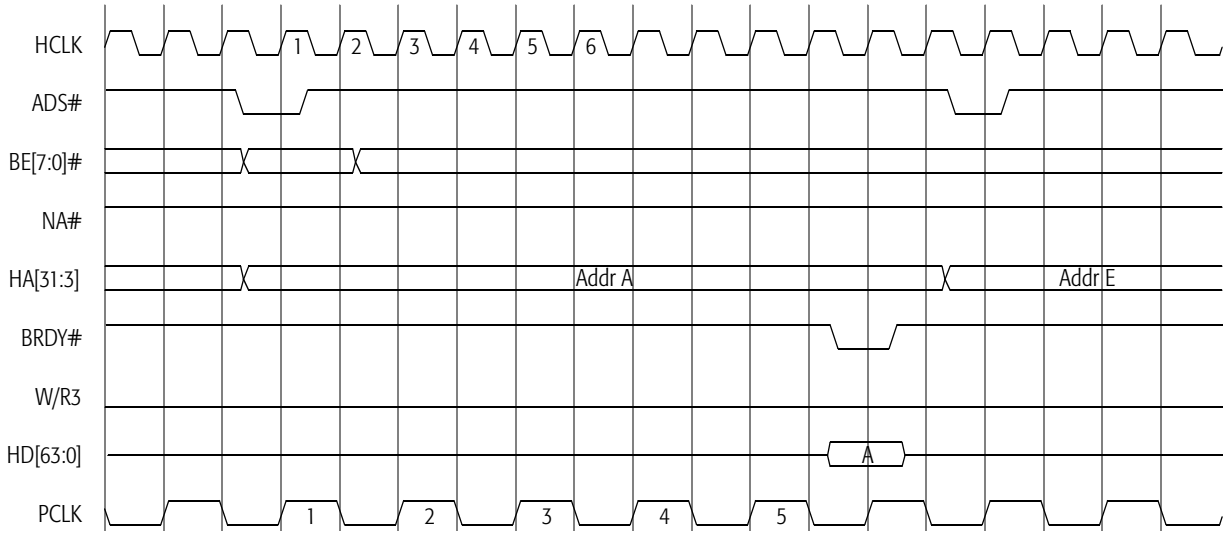



**Figure 5-21. Configuration Read**

## 5.4.5 PCI Transaction Examples

**CPU Read from PCI Target**

Figure 5-22 shows the processor reading from a target on the PCI bus. There is a six-clock latency from the PCI bus to the CPU bus. The single wait state on the PCI bus is included as an example and is not required.

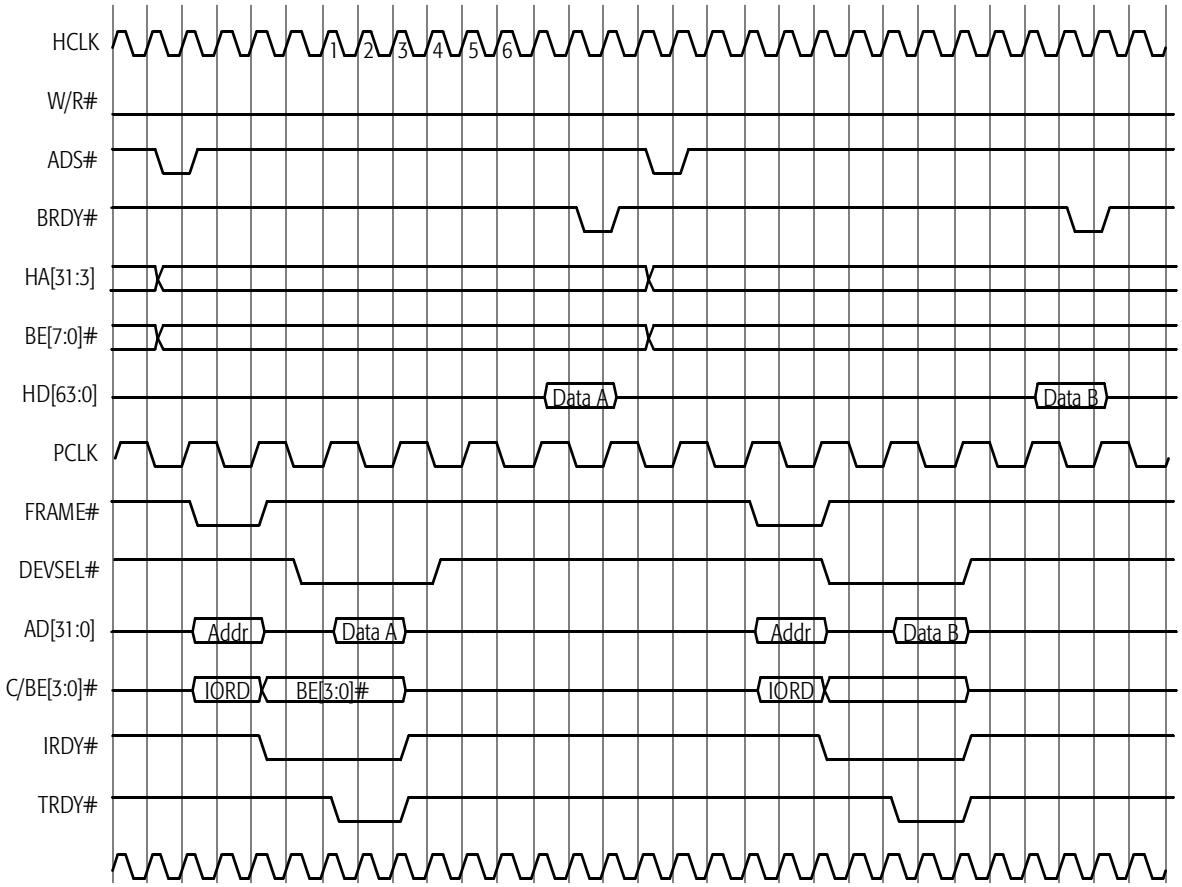



**Figure 5-22. Processor Read from PCI Target**

### CPU Write to PCI Target

Figure 5-23 shows the processor writing to a target on the PCI bus. The AMD-640 system controller stores the processor data in the PCI write buffer and controls the transfer from the write buffer to the PCI bus. There is a seven-clock latency from the processor to the PCI bus.

The PCI write buffer allows the processor to run fast back-to-back cycles. Note that NA# must be enabled by setting the CPU-to-PCI post write bit in the PCI Buffer Control Register, offset 70h, bit 7 (see page 7-32). Also, back-to-back cycles must be enabled in the Command Register, offset 05h, bit 9 (see page 7-6).

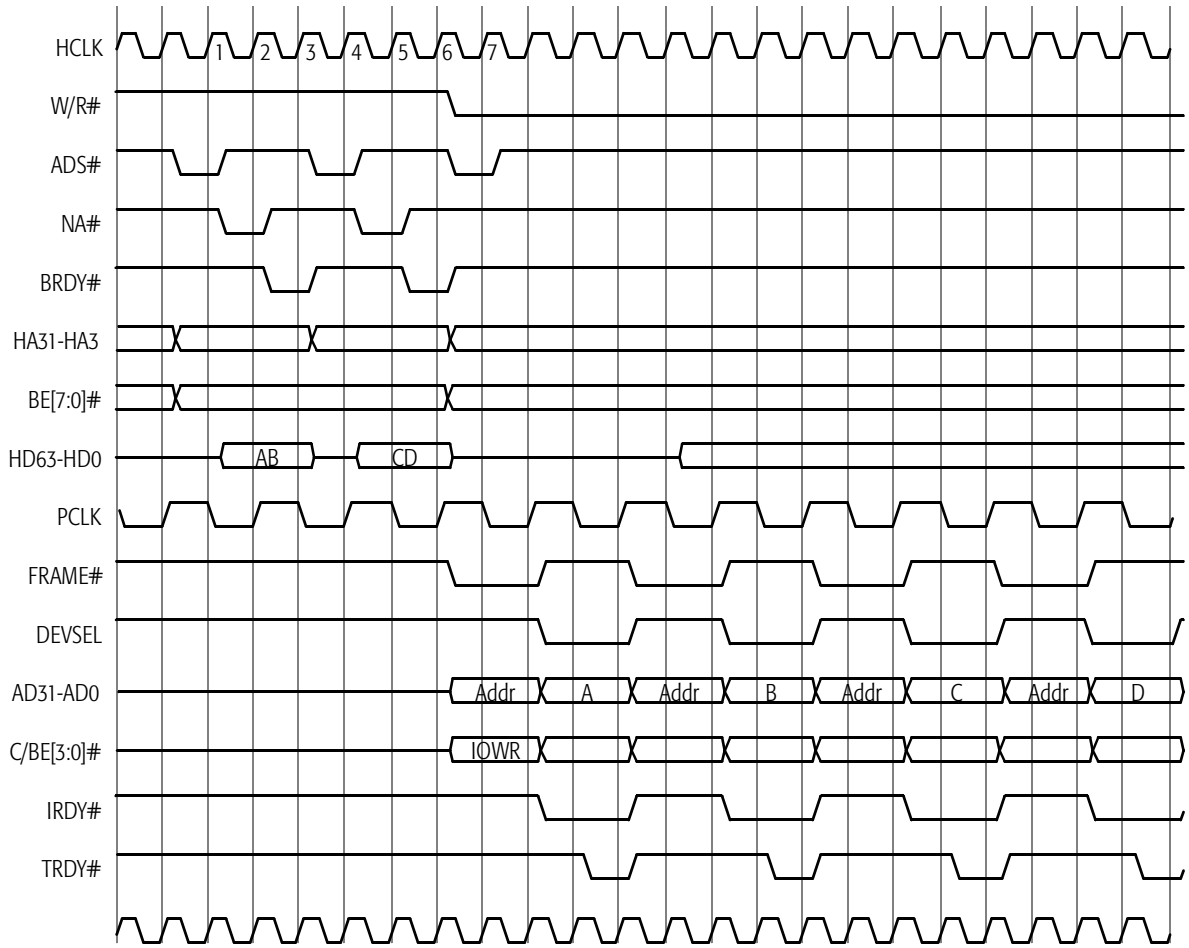



**Figure 5-23. Processor Write to PCI Target**

### PCI Bus Initiator Read: Cache Miss

Figure 5-24 shows a PCI bus initiator reading from memory. A page miss is indicated by RAS# toggling high, then low, to strobe in the new page address. The page miss accounts for the latency shown. Note that an entire cache line (A, B, C, D, E, F, G, H) is read from DRAM even though only four 32-bit words (A, B, C, D) are requested (FRAME# is negated at C). Also note that the DRAM data is echoed on the processor bus, indicating that there is no concurrence during this transfer. The numbers on the clocks are for reference, i.e., there are 3 clocks from RAS# to CAS#, 5 clocks from RAS# to data, and 11 PCLKs from FRAME# to the first data word on the PCI bus in this example.

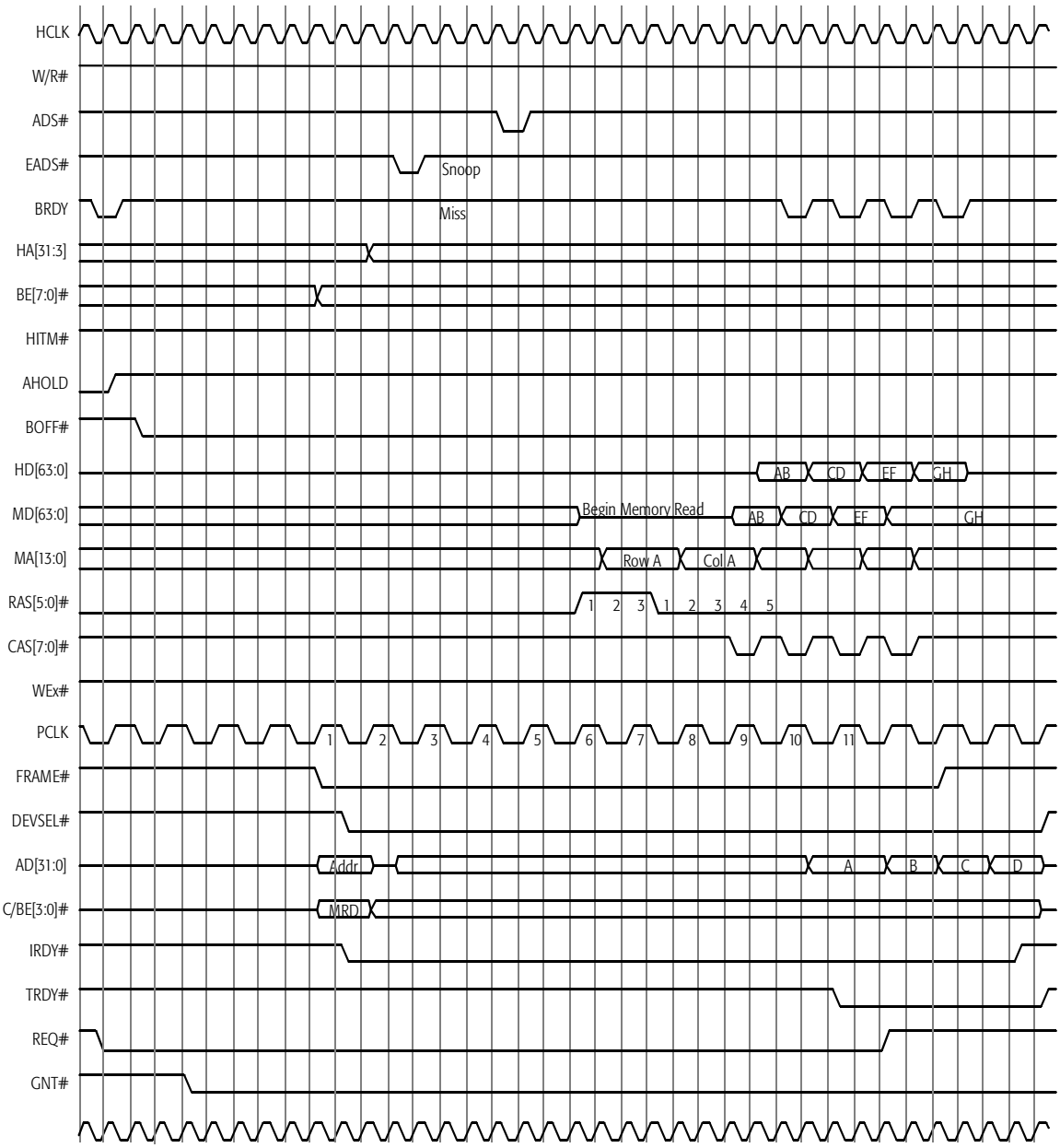



**Figure 5-24. PCI Bus Initiator Read: Cache Miss**

### PCI Bus Initiator Read: Modified L1 Hit, L2 Miss

Figure 5-25 shows a PCI initiator read. L1 and L2 are snooped for the data. L2 misses, but L1 hits a modified line (indicated by HITM#). The L1 cache controller writes the data to the DRAM write buffer and the PCI read buffer via the PCI forward mechanism. The PCI initiator reads the data from the PCI read buffer while the DRAM controller writes the data to DRAM. The L2 cache is not updated because the line is not present in L2.

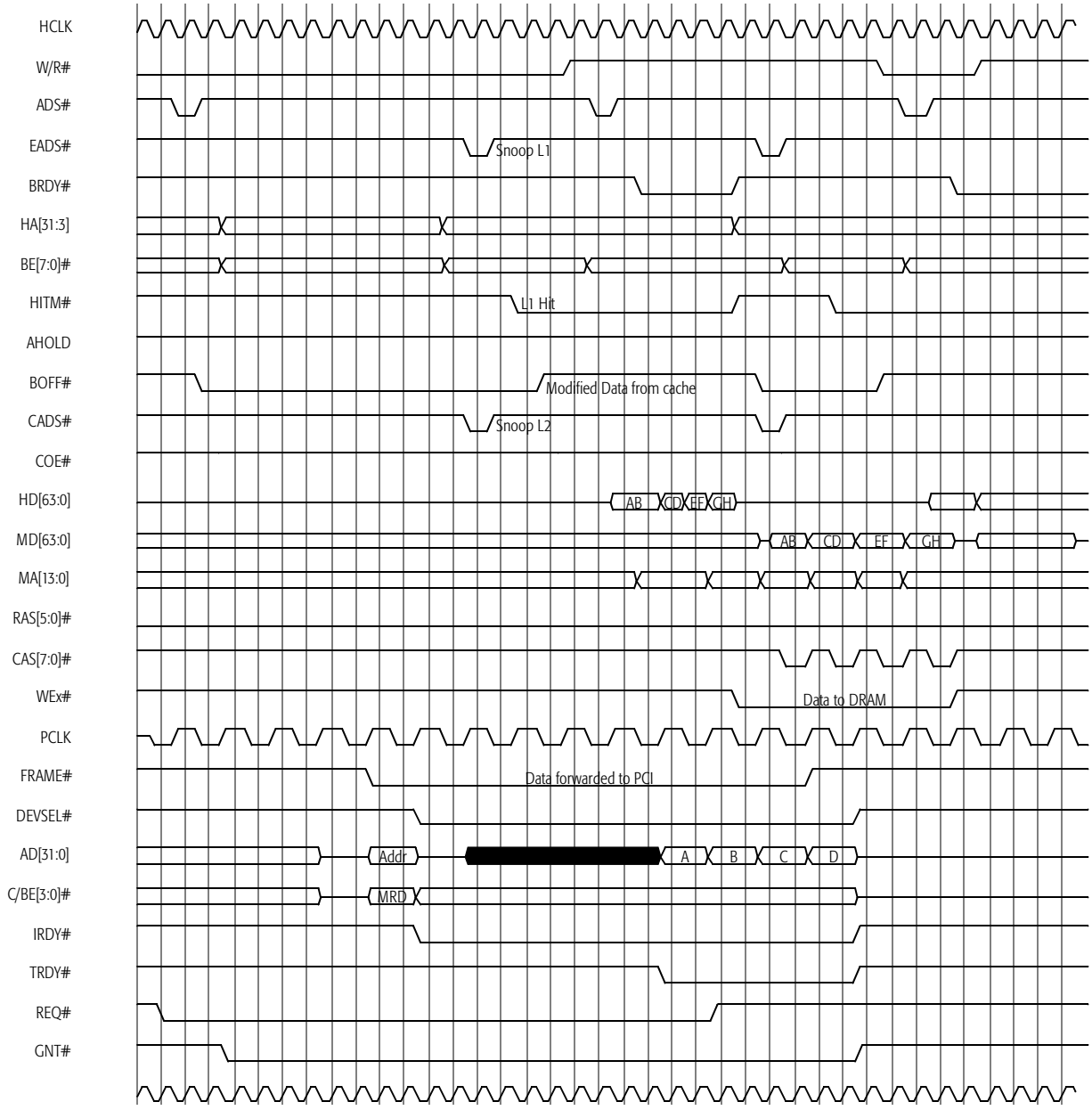



**Figure 5-25. PCI Bus Initiator Read: Modified L1 Hit, L2 Miss**

**PCI Bus Initiator Read: L1 Miss, Unmodified L2 Hit**

Figure 5-26 shows a PCI initiator read. L1 and L2 are snooped. L1 misses but L2 hits. There is no write to DRAM because the line is not modified. The L2 cache data is forwarded to the PCI bus. Note that the entire cache line is read, and if a successive PCI read addresses this data, it will be supplied directly from the AMD-640 system controller.

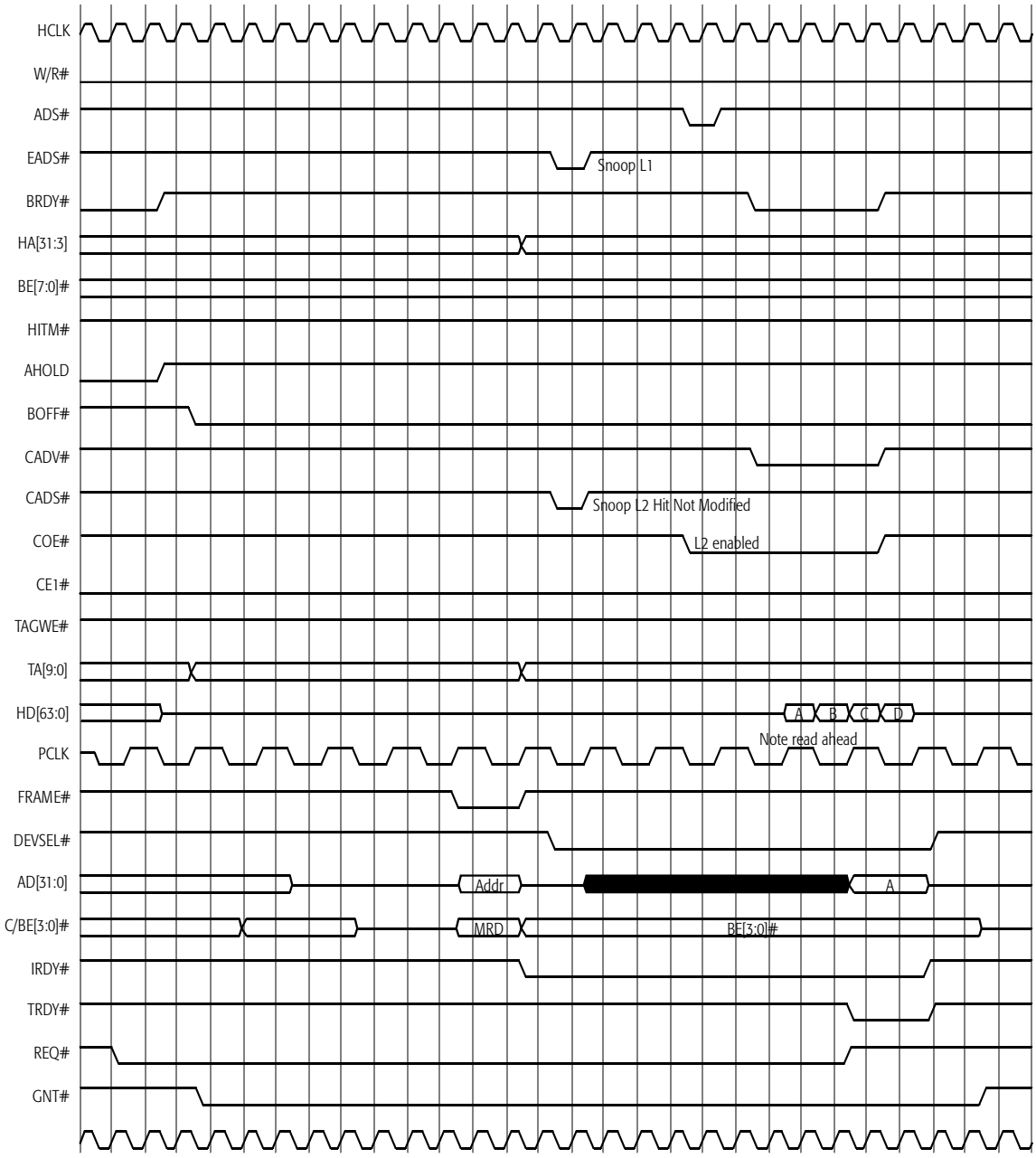



**Figure 5-26. PCI Bus Initiator Read: L1 Miss, Unmodified L2 Hit**

**PCI Bus Initiator Read: Modified L1 Hit**

Figure 5-27 shows another PCI initiator read. L1 and L2 are snooped. L1 has a hit on a modified line (HITM# is asserted). L2 also has a hit. The L1 data is written to both the L2 cache and the PCI read buffer.

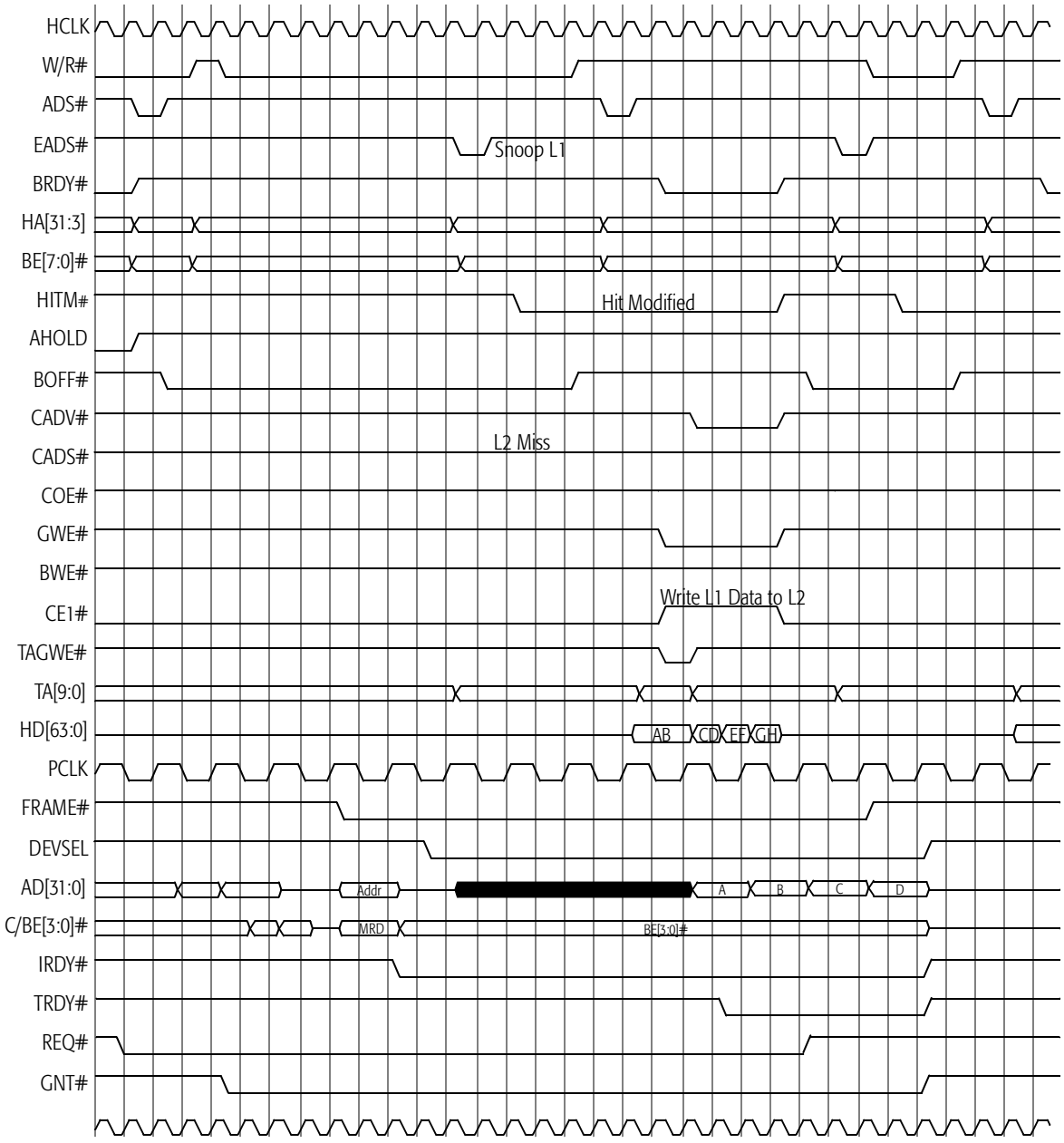



**Figure 5-27. PCI Bus Initiator Read: Modified L1 Hit**

**PCI Bus Initiator Write: Cache Miss**

Figure 5-28 shows a PCI write to DRAM. L1 and L2 are snooped, and both miss. The AMD-640 system controller stores the PCI data into its write buffer and subsequently writes this data (A, B, C, D) to DRAM.

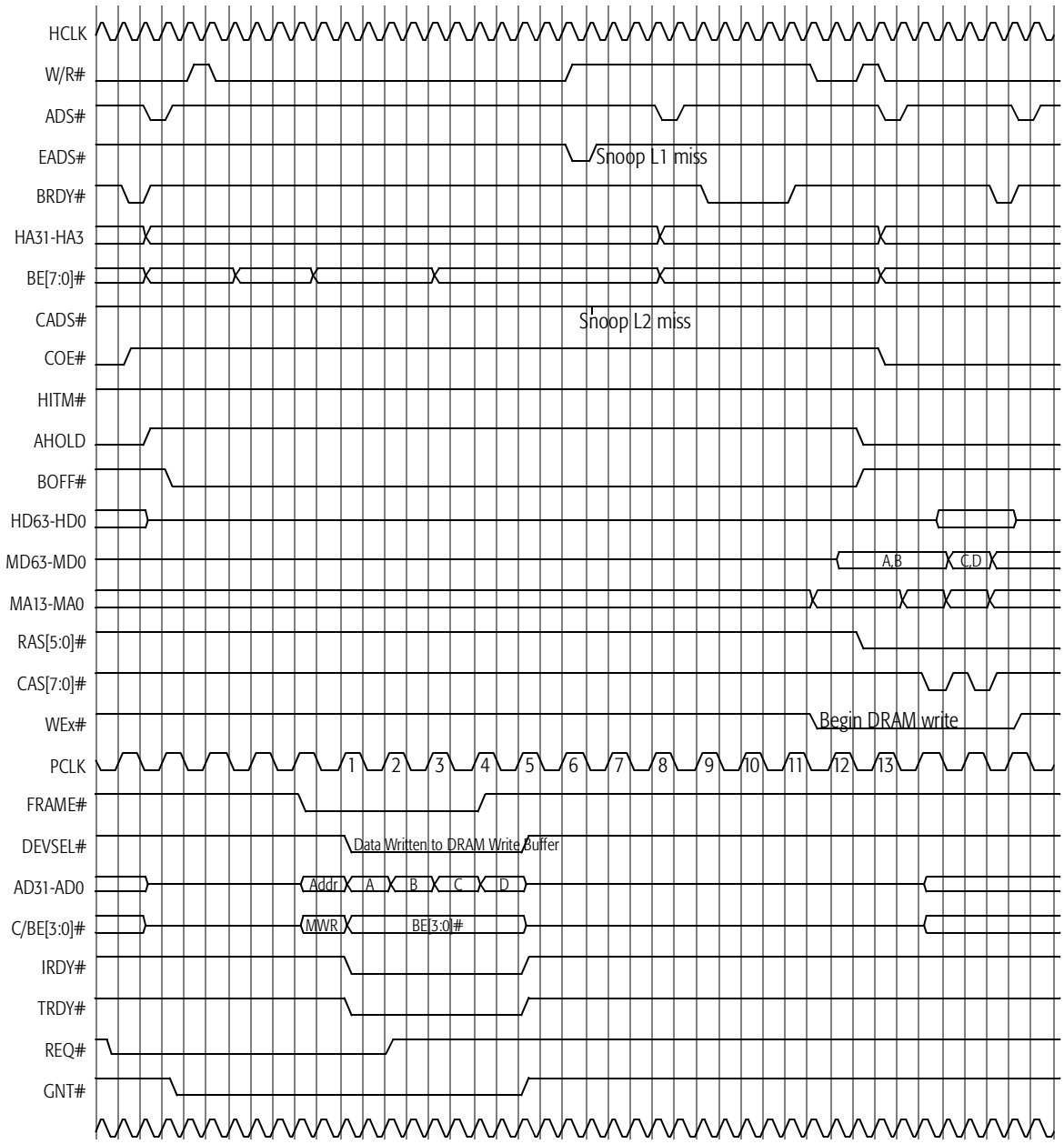



**Figure 5-28. PCI Bus Initiator Write: Cache Miss**

### PCI Bus Initiator Write: L1 Hit, L2 Miss

Figure 5-29 shows a PCI write to DRAM. L1 and L2 are snooped. L2 is a miss and L1 is a hit. The L1 cache controller writes data (G, H, I, J, K, L, M, N) to the DRAM write buffer. This data is merged with the data from the PCI bus (A, B, C, D) and written into DRAM. The cache line boundaries are assumed. There are many possible variations to the example shown.

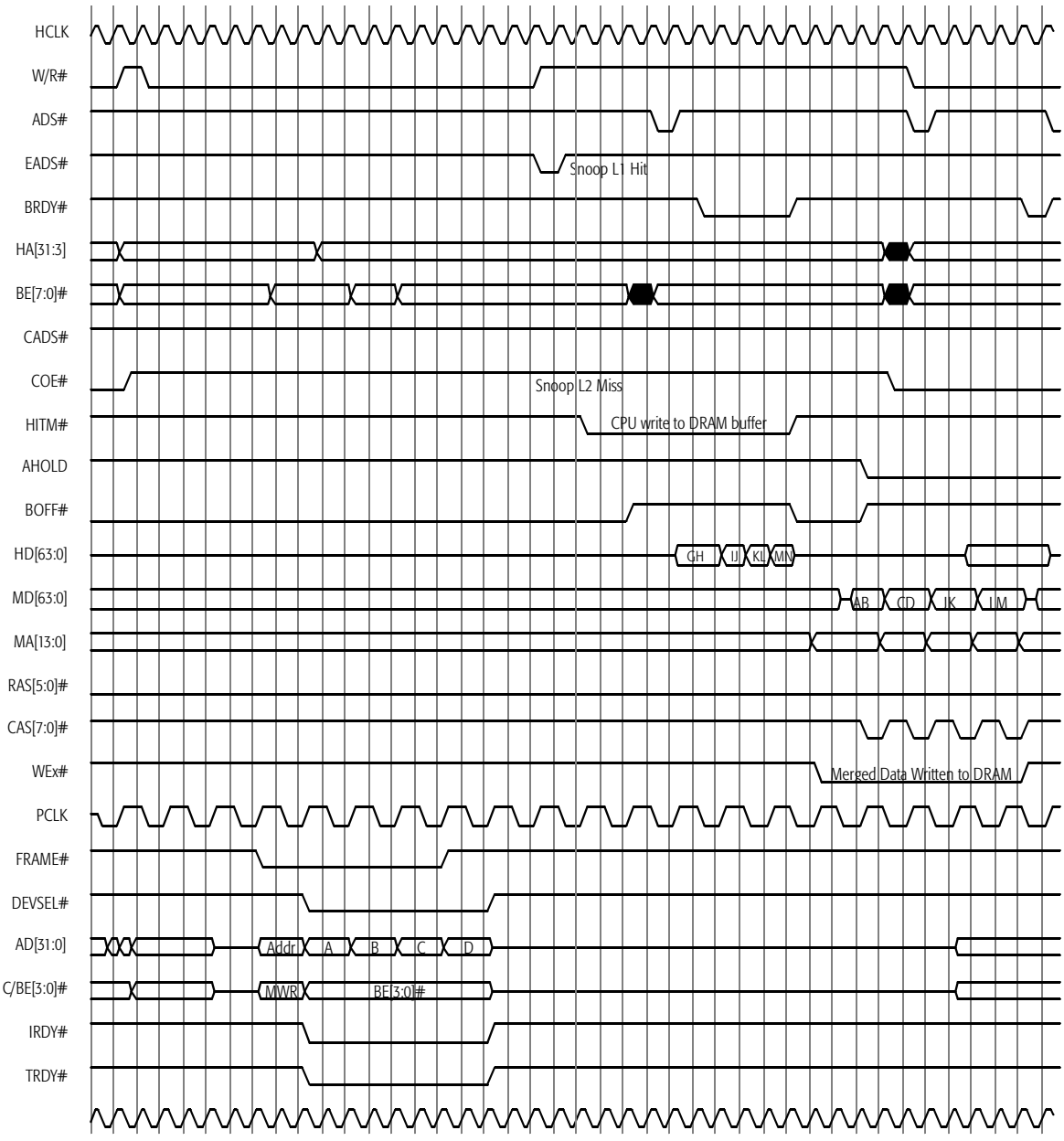



**Figure 5-29. PCI Bus Initiator Write: L1 Hit, L2 Miss**

**PCI Bus Initiator Write: L1 Miss, Unmodified L2 Hit**

Figure 5-30 shows a PCI write to DRAM. L1 and L2 are snooped. L1 is a miss and L2 is a hit. The L2 line is not written back because it is not modified. The line is simply marked invalid because new data is written into the DRAM.

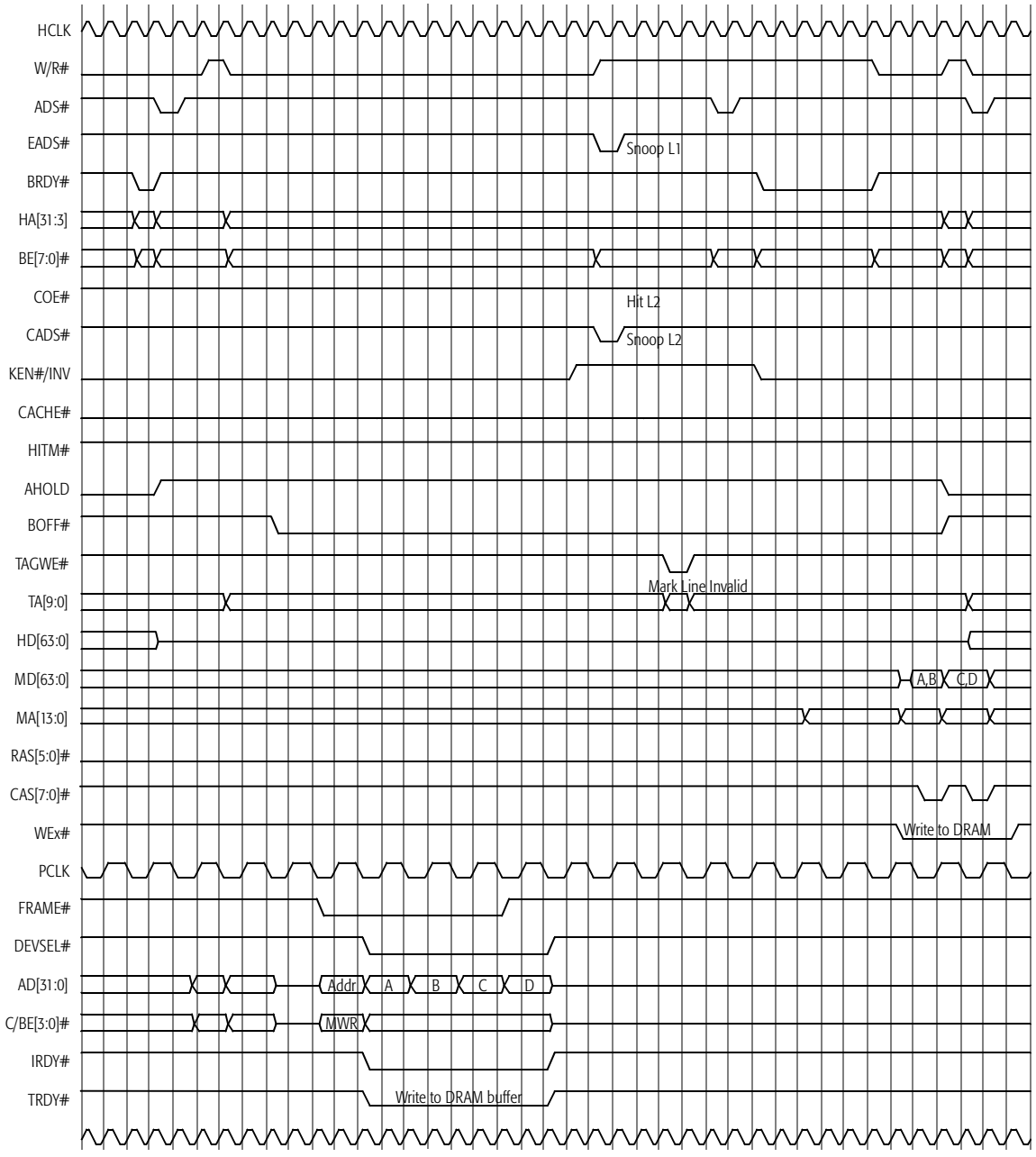



**Figure 5-30. PCI Bus Initiator Write: L1 Miss, Unmodified L2 Hit**

### PCI Bus Initiator Write: Modified L1 Hit, L2 Hit

Figure 5-31 shows a PCI initiator write to DRAM. L1 and L2 are snooped. The processor asserts HITM# indicating that the line in the L1 cache is modified. L2 also indicates a hit. The L1 data is written back to the DRAM buffer. The controller then merges the cache data with the PCI data and writes the merged data to DRAM.

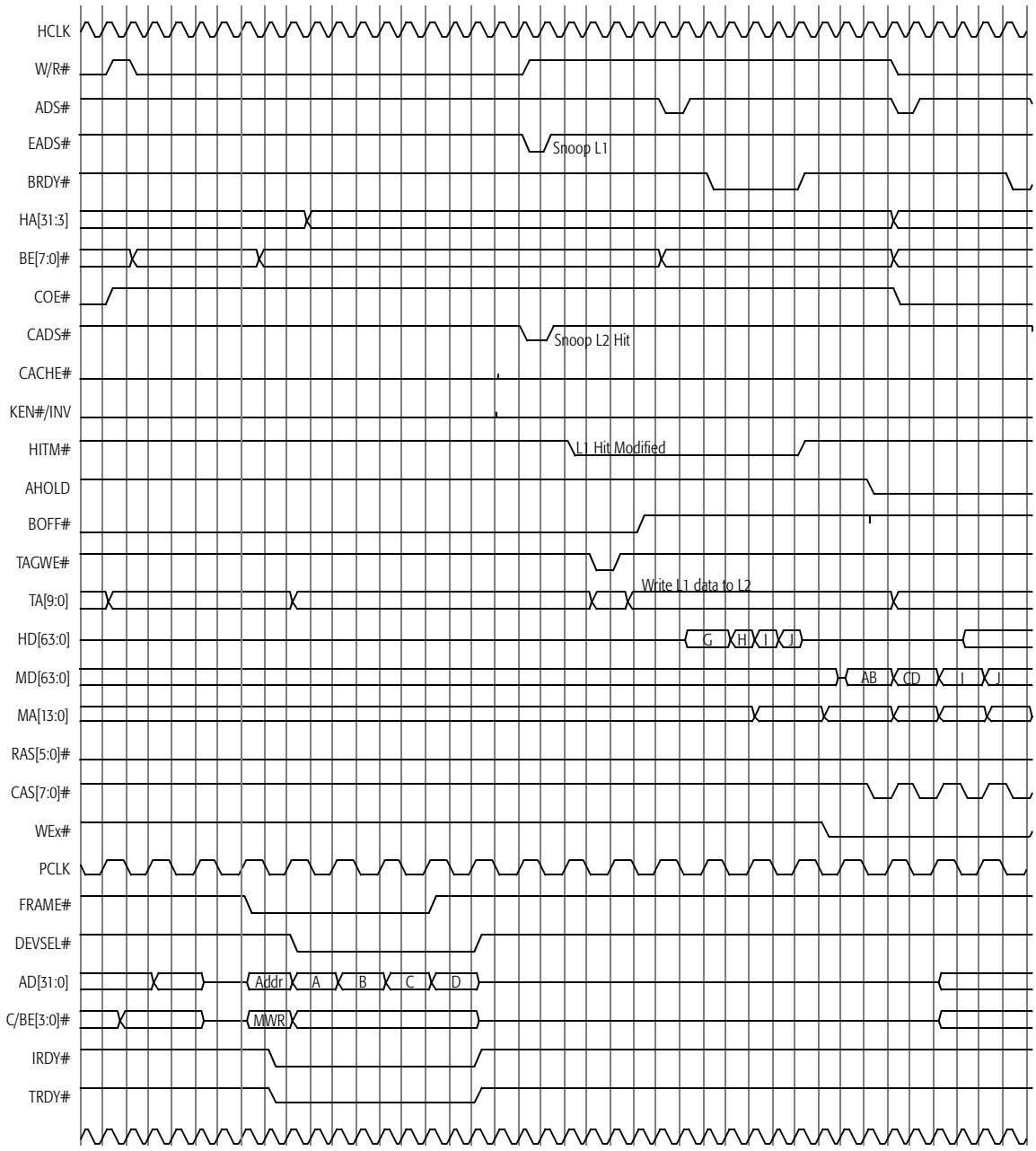



**Figure 5-31. PCI Bus Initiator Write: Modified L1 Hit, L2 Hit**

### PCI Bus Initiator Write: L1 Miss, Modified L2 Hit

Figure 5-32 shows a PCI initiator write to DRAM. L1 and L2 are snooped. L1 is a miss but L2 is a hit and is modified. The controller reads the modified L2 line into the DRAM write buffer, merges it with the PCI data, and writes the merged data to DRAM. TAGWE# invalidates the L2 cache entry by writing FFh. The first CAS# assertion indicates that a DRAM read access is started in parallel with interrogating the cache. The DRAM read aborts when the write cache hit is recognized.

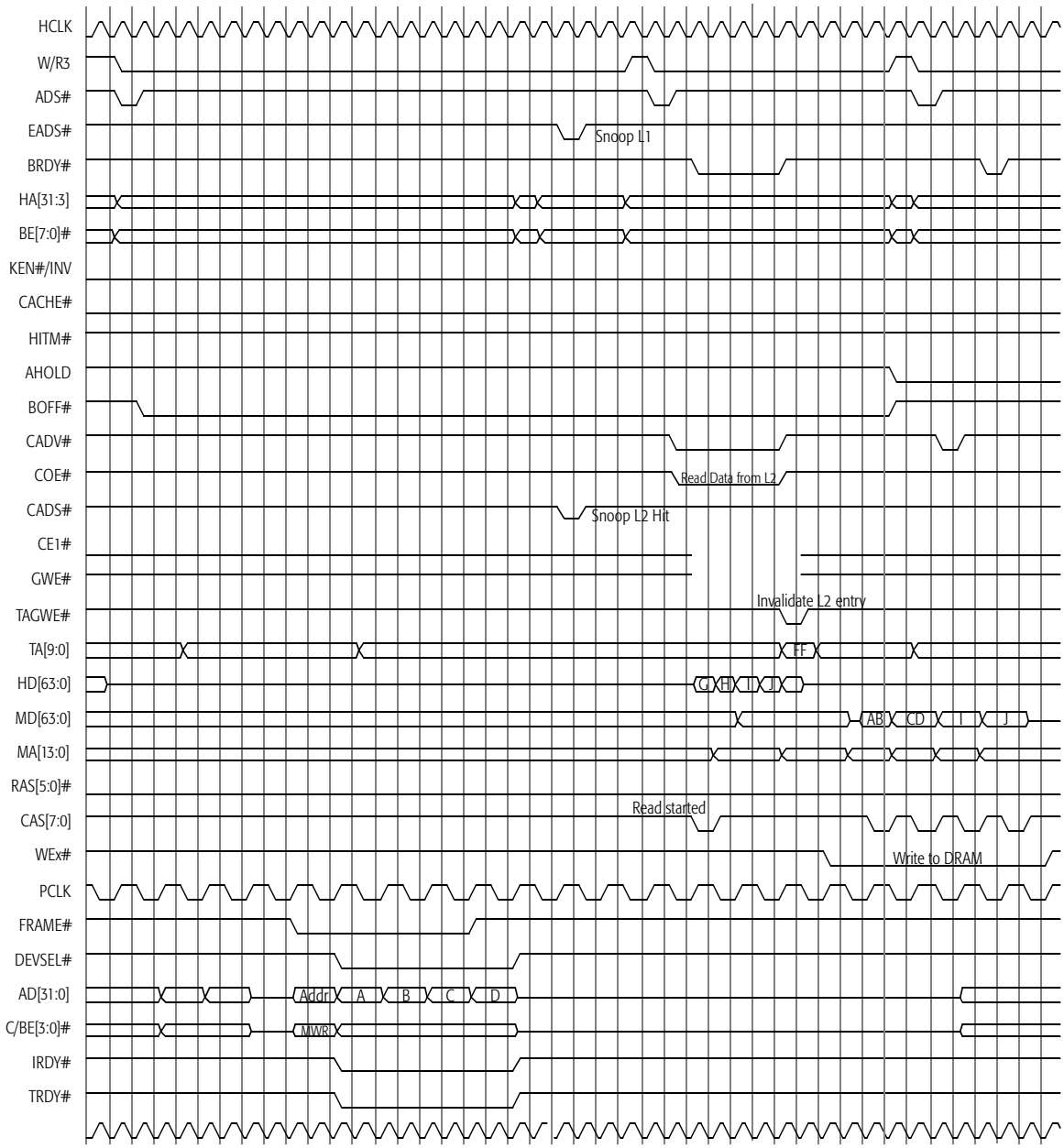



**Figure 5-32. PCI Bus Initiator Write: L1 Miss, Modified L2 Hit**

### 5.4.6 PCI Accesses by Another initiator

A PCI initiator begins a memory read or write cycle by asserting FRAME# and placing the memory address on AD[31:0]. The AMD-640 system controller decodes the address. If the address is within the domain of the host or memory, the AMD-640 system controller accepts the cycle and responds as a PCI target by asserting DEVSEL#. (If the address is not within controller or processor domain, the AMD-640 system controller ignores the cycle and allows it to complete on PCI.)

**PCI Reads**

In a PCI read, the AMD-640 system controller combines the 16-doubleword PCI buffer with the 10-doubleword DRAM read buffer, effectively forming a 26-doubleword PCI read buffer. The controller initiates a memory prefetch starting at the address sent by the initiator, reading data sequentially until the PCI read buffer is full. When the first doubleword of data is available, the controller supplies the data on AD[31:0] and asserts TRDY#, as shown in Figure 5-27 on page 5-45. As space becomes available, more data is prefetched until the cycle is complete. When the entire read is completed, the buffers are automatically invalidated to prevent stale data from being put out on a subsequent PCI initiator read.

If a read operation crosses a memory page boundary, the AMD-640 system controller initiates a target disconnect on the PCI bus at a line (32-byte) boundary.

Each address is passed to the processor bus to snoop the primary and secondary caches. If the address hits a cache entry, the data is supplied from the cache rather than from DRAM. To maintain data coherency, the AMD-640 system controller completes PCI initiator writes to memory before starting a PCI read.

**PCI Writes**

In a PCI write, the AMD-640 system controller combines the 16-doubleword PCI buffer with the 32-doubleword DRAM write buffer, effectively forming a 48-doubleword PCI write buffer. The AMD-640 system controller asserts TRDY# during the same clock it asserts DEVSEL#. On the next rising clock edge the controller samples the write data and posts it in the PCI write buffer, as shown in Figure 5-28 on page 5-46. Posting the write frees the PCI bus so that the next operation is not stalled waiting for completion of the write to memory.

If the buffer fills up before the write is complete, the controller will deassert TRDY# until the buffer has written some of its contents to memory and space is available for more data. The write is completed when the memory bus becomes available.

The L1 and L2 caches are snooped during the write to maintain cache integrity. If the address hits a cache entry the cache data is written and merged with the PCI data. The cache line is also invalidated. To maintain data coherency, the write buffer snoops PCI reads from memory. If read data hits the write buffer, the read stalls by negating TRDY# until the write is completed.

### 5.4.7 PCI Fast Back to Back cycles

The PCI specification allows fast back-to-back cycles to the same target or to different targets. In the AMD-640 system controller, this feature is controlled by the command register (offset 05h–04h) for reads and the PCI configuration register (offset 71h), bit 7 for writes. Offset 73h, bit 7 must be set for slow decode if fast back-to-back is selected.

On same-target back-to-back cycles, the initiator is responsible for preventing contention on TRDY#, DEVSEL#, STOP#, and PERR#. The AMD-640 system controller deletes the idle cycle prior to FRAME# and guarantees it will not produce any contention when it is driving the PCI bus.

On different-target back-to-back cycles, the target is responsible for preventing contention on TRDY#, DEVSEL#, STOP#, and PERR#. When this option is selected the AMD-640 system controller will capture the address without an intervening idle cycle. The AMD-640 system controller will delay assertion of TRDY#, DEVSEL#, STOP#, and PERR# by one clock to avoid contention. Avoiding contention in this mode is more difficult because the capabilities of all targets in the system must be known.

### 5.4.8 PCI Sideband Signals

The AMD-640 system controller supports one pair of PCI sideband signals, PREQ# and PACK#, to connect to a bridge device such as an ISA/EISA bridge. They are generally used when an alternate bus device, typically an ISA master or DMA device, requires ownership of the system's main memory. The alternate bus device asserts PREQ# to request the bus. The AMD-640 system controller grants the request after all of its write buffers have been flushed by asserting PACK#.

### 5.4.9 Power Management

The AMD-640 system controller supports the Advanced Power Management Specification, version 2.1. The counters required for this feature are contained in the AMD-645 peripheral bus controller companion device. SMIACT controls selection of the SMM memory space.

To initialize the SMM memory, the BIOS writes 01 to offset 63h, bits 1-0. This enables it to copy the SMM code into the SMM memory. The BIOS then writes 00, then 10 to these bits. (The program should not change the bits from 01 to 10 directly.) This action enables the controller to redirect processor accesses to the appropriate address range if SMIACT# is active.

To avoid cache coherence problems in SMM mode, connect the SMIACT# signal to FLUSH# on the processor.

# 6 Initialization

All programmable features in the AMD-640 system controller are controlled by the PCI configuration registers, which are normally written to only during system initialization. This section summarizes the register functions, default values, access types, and addresses (offset numbers). For more detailed descriptions of the configuration registers, see Section 7.

Recommended values are shown for FPM, EDO, and SDRAM. These values ensure acceptable performance but may not be optimal for a particular system. Refer to the BIOS guide for sample code.

Access types are indicated as follows:

R/W Read/Write
R/O Read Only
RWC Read, Write 1's to Clear individual bits

**Table 6-1. Configuration Space Header Registers**

| Offset | PCI Header | Default | Access |
|---|---|---|---|
| 01h–00h | Vendor ID | 1106h | RO |
| 03h–02h | Device ID | 0595h | RO |
| 05h–04h | Command | 0007h | RW |
| 07h–06h | Status | 02A0h | RWC |
| 08h | Revision ID | nn (note 1) | RO |
| 09h | Program Interface | 00h | RO |
| 0Ah | Bus Class Code | 00h | RO |
| 0Bh | Base Class Code | 06h | RO |
| 0Ch | Cache Line Size | 00h | RO |
| 0Dh | Latency Timer | 00h | RW |
| 0Eh | Header Type | 00h | RO |
| 0Fh | Built-In Self Test (BIST) | 00h | RO |
| 10h–3Fh | Reserved | 00h | – |

***Note:***

1. *nn changes for each device revision. Rev D = 02h is the current revision as of publication of this document (Rev E = 03h, Rev F = 04h).*

**Table 6-2. Configuration Space Cache Control Registers**

| Offset | Cache Control | Default | Recommended | | Access |
|---|---|---|---|---|---|
| | | | Setting | Result | |
| 50h | Cache Control 1 | 00h | 83h | Normal operation, PBSRAM | RW |
| 51h | Cache Control 2 | 00h | 01h | 1 bank; 512 kbytes | RW |
| 52h | Non-Cacheable Control | 02h | 96h | L1=L2=WB | RW |
| 53h | System Performance Control | 00h | 78h | PCI concurrency | RW |
| 55h–54h | Non-Cacheable Region #1 | 0000h | | | RW |
| 57–56h | Non-Cacheable Region #2 | 0000h | | | RW |

**Table 6-3. Configuration Space DRAM Control Registers**

| Offset | Cache Control | Default | Recommended | | Access |
|---|---|---|---|---|---|
| | | | Setting | Result | |
| 58h | DRAM Configuration Register #1 | 40h | 44h | 10-bit column | RW |
| 59h | DRAM Configuration Register #2 | 05h | 03h | banks 0-3 populated | RW |
| 5Ah | DRAM Bank 0 Ending [HA29-22] | 01h | 10h/02h | 64 Mbytes/8 Mbytes | RW |
| 5Bh | DRAM Bank 1 Ending [HA29-22] | 01h | 20h/04h | 64 Mbytes/8 Mbytes | RW |
| 5C | DRAM Bank 2 Ending [HA29-22] | 01h | 30h/06h | 64 Mbytes/8 Mbytes | RW |
| 5Dh | DRAM Bank 3 Ending [HA29-22] | 01h | 40h/08h | 64 Mbytes/8 Mbytes | RW |
| 5Eh | DRAM Bank 4 Ending [HA29-22] | 01h | 50h/08h | 64 Mbytes/no RAM | RW |
| 5Fh | DRAM Bank 5 Ending [HA29-22] | 01h | 60h/08h | 64 Mbytes/no RAM | RW |
| 60h | DRAM Type | 00h | 00h<br>05h | Fast Page Mode<br>EDO mode banks 0-3 | RW |
| 61h | Shadow RAM Control Register #1 | 00h | CAh | Video BIOS | RW |
| 62h | Shadow RAM Control Register #2 | 00h | 00h | Disable | RW |
| 63h | Shadow RAM Control Register #3 | 00h | 22h | Main BIOS | RW |
| 64h | DRAM Timing | ABh | FFh<br>44h<br>57h | Slowest initially<br>60 ns EDO<br>60 ns FP | RW |
| 65h | DRAM Control Register #1 | 00h | A8h | Page open<br>Fast decode<br>Latch delay | RW |
| 66h | DRAM Control Register #2 | 00h | 01h | SDRAM | RW |
| 67h | 32-Bit DRAM Width Control Register | 00h | 00h | 64-bit DRAM | RW |
| 69h–68h | Reserved | | | | – |
| 6Ah | DRAM Refresh Counter | 00h | 42h | 15 µsec | RW |
| 6Bh | DRAM Refresh Control Register | 00h | 80h | CBR | RW |
| 6Ch | SDRAM Control Register | 00h | 00h | | RW |
| 6Dh | DRAM Drive Strength Control Register | 00h | 4Fh | 24-ma drive | RW |
| 6Eh | ECC Control Register | 00h | 00h | | RW |
| 6Fh | ECC Status Register | 00h | 00h | | RWC |

**Table 6-4. Configuration Space PCI Control Registers**

| Offset | Cache Control | Default | Recommended | | Access |
|---|---|---|---|---|---|
| | | | Setting | Result | |
| 70h | PCI Buffer Control 1 | 00h | E0h | Enable write buffers & prefetch | RW |
| 71h | Processor to PCI Flow Control #1 | 00h | DEh | Post writes to DRAM & Merge | RW |
| 72h | Processor to PCI Flow Control #2 | 02h | ECh | Reduce FRAME | RWC |
| 73h | PCI Target Control | 00h | 8Dh | STOP control | RW |
| 74h | PCI Initiator Control | 00h | C0h | Enhance commands | RW |
| 75h | PCI Arbitration Control #1 | 00h | 00h | | |
| 76h | PCI Arbitration Control #2 | 00h | 80h | | RW |

# 7 Configuration Registers

All of the many options available on the AMD-640 system controller are selected by writing to its configuration registers. These registers are usually set during system initialization and are not accessed during normal operation. However, some registers may require specific programming sequences during power-up to detect the type and size of installed memory.

This section contains a description of the mechanism used to access the AMD-640 system controller's configuration registers and describes the location and definition of each register.

## 7.1 PCI Configuration Mechanism

The AMD-640 system 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–0CFBh to specify the target address and locations 0CFCh–0CFFh for data to or from the target address. The target address includes the PCI bus, device, function, and register numbers of the 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 | | |

To specify the AMD-640 system controller, set bit 31, the enable bit. If bit 31 is cleared, the AMD-640 system controller passes the data through as an I/O transaction. The bus number, device number, and function number of the AMD-640 system controller are all 00h.

**Note:** In the AMD-640 system controller, IDSEL is internally connected to AD11. Other PCI devices in a system must connect their IDSEL lines to a unique line in AD[31:12], and cannot use AD11.

The PCI specification calls for 256 configuration registers in each target device to be organized on doubleword boundaries. Each register is numbered as an "offset" from zero. To access a particular register, the most significant six bits of the offset are written to bits 7-2 of the target address to specify the register's doubleword boundary, while the PCI byte enable lines C/BE[3:0]# select the byte represented by the least two significant bits of the offset. Some registers are described as 16- or 32-bit entities, in which case two or four byte enable lines, respectively, are asserted. For example, the Status Register is described as residing at offset 07h–06h. 07h = 00000111b and 06h = 00000110b. To access this register, write the six most significant bits (000001 in this example) of either byte to bits 7 2 of I/O address 0CF8h to specify the doubleword (all multi-byte registers reside within the same doubleword), and assert byte enables 3 and 2 (clear C/BE3# and C/BE2#) corresponding to the two least significant bits of 07h and 06h, respectively.

Table 7-1 summarizes the I/O ports involved in PCI configuration.

**Table 7-1. Configuration Port Register Summary**

| Register Name | I/O Address | Type | Default Value | Size |
|---|---|---|---|---|
| IO_CNTRL | 0CF8h | R/W | 0000 0000h | 32 |
| IO_DATA32 | 0CFCh | R/W | 0000 0000h | 32 |
| IO_ODD_DATA16 | 0CFCh | R/W | 0000h | 16 |
| IO_EVEN_DATA16 | 0CFEh | R/W | 0000h | 16 |
| IO_0_DATA8 | 0CFCh | R/W | 00h | 8 |
| IO_1_DATA8 | 0CFDh | R/W | 00h | 8 |
| IO_2_DATA8 | 0CFEh | R/W | 00h | 8 |
| IO_3_DATA8 | 0CFFh | R/W | 00h | 8 |

# 7.2 Register Overview

Tables 7-2 through 7-5 summarize the AMD-640 system controller configuration register offsets, functions, default values, and access types. Access types are indicated as follows:

R/W Read/Write
R/O Read Only
RWC Read, Write 1's to Clear individual bits

**Table 7-2. Configuration Space Header Registers**

| Offset | PCI Header | Default | Access |
|---|---|---|---|
| 01h–00h | Vendor ID | 1106h | RO |
| 03h–02h | Device ID | 1595h | RO |
| 05h–04h | Command | 0007h | RW |
| 07h–06h | Status | 02A0h | RWC |
| 08h | Revision ID | nn (Note 1) | RO |
| 09h | Program Interface | 00h | RO |
| 0Ah | Bus Class Code | 00h | RO |
| 0Bh | Base Class Code | 06h | RO |
| 0Ch | Cache Line Size | 00h | RO |
| 0Dh | Latency Timer | 00h | RW |
| 0Eh | Header Type | 00h | RO |
| 0Fh | Built-In Self Test (BIST) | 00h | RO |
| 10h–3Fh | Reserved | 00h | – |

***Notes:***

1. *nn changes for each device revision. Rev D = 02h was the current revision as of publication of this document (Rev E = 03h, Rev F = 04h).*

**Table 7-3. Configuration Space Cache Control Registers**

| Offset | Cache Control | Default | Access |
|---|---|---|---|
| 50h | Cache Control 1 | 00h | RW |
| 51h | Cache Control 2 | 00h | RW |
| 52h | Non-Cacheable Control | 02h | RW |
| 53h | System Performance Control | 00h | RW |
| 55h–54h | Non-Cacheable Region #1 | 0000h | RW |
| 57h–56h | Non-Cacheable Region #2 | 0000h | RW |

**Table 7-4. Configuration Space DRAM Control Registers**

| Offset | Cache Control | Default | Access |
|---|---|---|---|
| 58h | DRAM Configuration Register #1 | 40h | RW |
| 59h | DRAM Configuration Register #2 | 05h | RW |
| 5Ah | DRAM Bank 0 Ending | 01h | RW |
| 5Bh | DRAM Bank 1 Ending | 01h | RW |
| 5Ch | DRAM Bank 2 Ending | 01h | RW |
| 5Dh | DRAM Bank 3 Ending | 01h | RW |
| 5Eh | DRAM Bank 4 Ending | 01h | RW |
| 5Fh | DRAM Bank 5 Ending | 01h | RW |
| 60h | DRAM Type | 00h | RW |
| 61h | Shadow RAM Control Register #1 | 00h | RW |
| 62h | Shadow RAM Control Register #2 | 00h | RW |
| 63h | Shadow RAM Control Register #3 | 00h | RW |
| 64h | DRAM Timing | ABh | RW |
| 65h | DRAM Control Register #1 | 00h | RW |
| 66h | DRAM Control Register #2 | 00h | RW |
| 67h | 32-Bit DRAM Width Control Register | 00h | RW |
| 68h | Reserved | 00h | RW |
| 69h | Reserved | 00h | RW |
| 6Ah | DRAM Refresh Counter | 00h | RW |
| 6Bh | DRAM Refresh Control Register | 00h | RW |
| 6Ch | SDRAM Control Register | 00h | RW |
| 6Dh | DRAM Drive Strength Control Register | 00h | RW |
| 6Eh | ECC Control Register | 00h | RW |
| 6Fh | ECC Status Register | 00h | RWC |

**Table 7-5. Configuration Space PCI Control Registers**

| Offset | PCI Bus Control | Default | Access |
|---|---|---|---|
| 70h | PCI Buffer Control Register | 00h | RW |
| 71h | Processor-to-PCI Control Register#1 | 00h | RW |
| 72h | Processor-to-PCI Control Register#2 | 00h | RWC |
| 73h | PCI Initiator Control Register #1 | 00h | RW |
| 74h | PCI Initiator Control Register #2 | 00h | RW |
| 75h | PCI Arbitration Control Register #1 | 00h | RW |
| 76h | PCI Arbitration Control Register #2 | 00h | RW |
| 77h–FFh | Reserved | 00h | RW |

# 7.3 PCI Configuration Space Registers

## 7.3.1 Vendor ID (Offset 01h–00h)

| | 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 |

This read-only value is defined as 1106h.

## 7.3.2 Device ID (Offset 03h–02h)

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

This read-only value of 1595h represents the AMD-640 system controller.

## 7.3.3 Command (Offset 05h–04h)

| | Bits 15–10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Reserved | FBBCE | SERRE | STEP | Reserved | VGAPS | MWIC | SCMON | INITEN | MEMSPC | IOSPC |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |

**Bits 15–10 Reserved (always reads 0)**

**Bit 9 Fast Back-to-Back Cycle Enable (RW)**—See Section 5.4.7 on page 5-52 for a discussion of back-to-back cycles.

0 = Fast back-to-back transactions only allowed to the same agent (default)
1 = Fast back-to-back transactions allowed to different agents

**Bit 8 SERR# Enable (RW)**—This bit does not affect setting of bit 14 in offset 07h–06h.

0 = SERR# driver disabled (default)
1 = SERR# driver enabled

***Note:*** *If a system error occurs, SERR# may be asserted by a PCI master or by the AMD-645 peripheral bus controller.*

**Bit 7 Address/Data Stepping (always reads 0)**

0 = Device never does stepping

**Bit 6 Reserved (RW)**—This bit must remain at the default value of 0.

**Bit 5 VGA Palette Snoop (always reads 0)**

0 = Palette accesses generate normal PCI cycles

**Bit 4 Memory Write and Invalidate Command (always reads 1)**—This feature increases overall performance by eliminating cache writebacks when a PCI initiator writes to the address of a modified line. The AMD-640 system controller invalidates the cache line rather than writing it back to DRAM.

1 = Bus initiators may generate Memory Write and Invalidate

**Bit 3 Special Cycle Monitoring (always reads 0)**

0 = Special cycles not monitored

**Bit 2 Initiator Enable (always reads 1)**

1 = AMD-640 system controller can behave as bus initiator

**Bit 1 Memory Space (always reads 1)**

1 = Responds to memory space

**Bit 0 I/O Space (always reads 1)**

1 = Responds to I/O space

## 7.3.4 Status (Offset 07h–06h)

| | Bit 15 | 14 | 13 | 12 | 11 | 10–9 | 8 | 7 | 6 | 5 | Bits 4–0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Rsvd | SSE | SIA | RTA | STA | DEVSEL# Timing | DPED | FBBC | Reserved | 66 MHz | Reserved |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 1 | 0 | 1 | 0 | 1 | 0 0 0 0 0 |

**Bit 15** **Reserved (always reads 0)**

**Bit 14** **Signaled System Error (always reads 0)**

0 = No error detected
1 = ECC error detected. (ECC errors are reported in this status bit but the SERR# pin is not asserted.)

**Bit 13** **Signaled Initiator Abort (RO)**—This bit is set by a PCI initiator when its transaction is terminated with Initiator Abort.

0 = PCI transactions proceeding normally
1 = The AMD-640 system controller, acting as PCI initiator, has terminated a transaction before completion

**Bit 12** **Received Target Abort (RWC)**—The target issues a target abort when it detects a fatal error or cannot complete a transaction. This bit is set by simultaneously deasserting DEVSEL# and asserting STOP#.

0 = No abort received
1 = Transaction aborted by target

**Bit 11** **Signaled Target Abort (RO)**—This bit is set when a PCI initiator accesses DRAM and the AMD-640 system controller cannot respond, or when the AMD-640 system controller accesses a target device and the target cannot respond.

0 = PCI transactions proceeding normally
1 = Transaction aborted by target

**Bits 10–9** **DEVSEL# Timing (always reads 01)**—This field indicates that the slowest DEVSEL# timing will be medium.

00 = Fast
01 = Medium (AMD-640 system controller only implements this timing)
10 = Slow
11 = Reserved

**Bit 8** **Data Parity Error Detected (RO)**—The AMD-640 system controller samples the PCI PAR line to check for parity errors during a PCI read.

0 = No parity error
1 = Parity error was detected

**Bit 7** **Fast Back-to-Back capable (always reads 1)**—The AMD-640 system controller can accept fast back-to-back transactions, including those from different agents.

**Bit 6** **User-Defined Features (always reads 0)—**The AMD-640 system controller does not support user-defined features**.**

**Bit 5** **66 MHz Capable PCI Bus (always reads 1)—**The maximum PCI bus operating speed is 66 MHz.

**4–0** **Reserved (always reads 0)**

## 7.3.5 Revision ID (Offset 08h)

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | AMD-640 System Controller Chip Revision Code | | | | | | | |
| Reset | – | – | – | – | – | – | – | – |

**Bits 7–0** **AMD-640 System Controller Revision Code (RO)—**04h = Revision F (as of June 1997). 05h = Revision G. 06h = Revision H.

## 7.3.6 Programming Interface (Offset 09h)

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | Programming Interface | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7–0** **AMD-640 System Controller Programming Interface (always reads 00h)—**This register is defined in different ways for each combination of base and subclass codes. It is undefined for this type of device.

## 7.3.7 Sub Class Code (Offset 0Ah)

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | Sub Class Code | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7–0** **Sub Class Code (always reads 00h)**—The PCI-defined sub class code for a host bridge is 00h.

## 7.3.8 Base Class Code (Offset 0Bh)

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | Base Class Code | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |

**Bits 7–0** **Base Class Code (always reads 06h)**—The PCI-defined base class code for a bridge device is 06h.

## 7.3.9 Cache Line Size (Offset 0Ch)

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | Cache Line Size | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7–0** **Cache Line Size (always reads 0)**—The AMD-640 system controller accepts, but does not implement, the PCI Memory Write and Invalidate (MWI) command, for which the cache lines size is a required component.

## 7.3.10 Latency Timer (Offset 0Dh)

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | Latency Timer Values | | | | | Reserved | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7–3** **Latency Timer Value (RW)**—This five-bit binary value specifies the Latency Timer in units of 8 PCI bus clocks.

00000 = 32 x 8 PCI clocks
00001–11111 = (5-bit binary value) x 8 PCI clocks

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

## 7.3.11 Header Type (Offset 0Eh)

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | Header Type | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7–0** **PCI Header Type (RO)**—The AMD-640 system controller PCI Header Type is 00h.

## 7.3.12 Built-In Self Test (BIST) (Offset 0Fh)

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | Built-in Self Test Functions | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7–0** **Built-in Self Test Functions (RO)**—The AMD-640 system controller does not support built-in self-test functions, so this read-only register is 00h.

# 7.4 Cache Control Registers

## 7.4.1 Cache Control Register 1 (Offset 50h)

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | CACHEN | | Reserved | TAGCON | | Reserved | SRMTYP | |
| Reset | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

**Bits 7–6 Cache Enable (RW)**

00 = Cache disabled (default)
01 = Cache initialization—BIOS fills the L2 cache to a known state
10 = Cache enabled—normal operation
11 = Reserved

**Bit 5 Reserved (RW)**—This bit must remain at the default value 1.

**Bits 4–3 Tag Configuration (RW)**

00 = 8+0—8 tag bits, no modify bit (default)
01 = 7+1—7 tag bits, one modify bit
10 = 10+0—10 tag bits, no modify bit
11 = 9+1—9 tag bits, one modify bit

**Bit 2 Reserved (always reads 0)**

**Bits 1–0 Cache SRAM Type (RW)**

00 = No SRAM (default)
01 = Reserved
10 = Burst SRAM
11 = Pipeline Burst SRAM

## 7.4.2 Cache Control Register 2 (Offset 51h)

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | Reserved | | BOP | Reserved | SRAMBNK | Reserved | Cache Size | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

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

**Bit 5** **Backoff Processor (RW)**—Used when register 52h, bit 2 is set for "L2 fill when CACHE# is inactive." This bit should normally be cleared to 0 for best performance, although system-level performance differences are usually negligible.

0 = Defer ready return (i.e., do not assert BRDY#) until L2 is filled (default)
1 = Backoff processor (assert BOFF#) until L2 is filled

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

**Bit 3** **SRAM Banks (RW)**

0 = One bank (default). With no NA# delay, pipelined read hit timing is 3-1-1-1-1-1-1-1 (no bank to switch).
1 = Two banks. With no NA# delay, pipelined read hit timing is 3-1-1-1-2-1-1-1 for bank switch.

**Bit 2** **Reserved (always reads 0)**

**Bits 1–0** **Total L2 Cache Size**

00 = 256 Kbytes (default)
01 = 512 Kbytes
10 = 1 Mbyte
11 = 2 Mbytes

## 7.4.3 Non-Cacheable Control Register (Offset 52h)

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | WPC | WPD | WPE | WPF | Reserved | L2FILL | Reserved | L2WBWT |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

**Bit 7** **L2 Write-Protect / L2 Cacheability for Addresses C0000h–C7FFFh (RW)**
**Bit 6** **L2 Write-Protect / L2 Cacheability for Addresses D0000h–DFFFFh (RW)**
**Bit 5** **L2 Write-Protect / L2 Cacheability for Addresses E0000h–EFFFFh (RW)**
**Bit 4** **L2 Write-Protect / L2 Cacheability for Addresses F0000h–FFFFFh (RW)**

Each of these bits enables its associated L2 cache address range to contain the associated BIOS code, which can improve performance. Setting these bits not only enables BIOS caching but protects the cached BIOS from modification by inadvertent writes. When one of these bits is set, a read access in the associated address range will load data into the cache, and subsequent reads will come from the cache. A write will not affect the L2 cache contents but will be passed to DRAM. (Refer to registers 61h–63h on page 7-28 for DRAM response to writes.)

0 = Address range is neither cacheable nor write-protected (default)
1 = Address range is both cacheable and write-protected.

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

**Bit 2** **L2 Fill (RW)**—Setting this bit forces the requested data to be filled into the L2 cache (provided that L2 cache is enabled), even if the processor does a read cycle with CACHE# deasserted. Although the AMD-640 system controller ignores the non-cacheable settings in the processor when this bit is set, it still adheres to the non-cacheable settings in its configuration registers.

0 = Normal L2 cache fill (default)
1 = Force L2 cache fill

***Note:*** *Setting this bit significantly improves performance.*

**Bit 1** **Reserved (RW)**

1 = (default)

**Bit 0** **L2 Writeback or Writethrough (RW)**—This bit determines if the L2 cache operates as writeback or writethrough.

0 = Writeback (default)
1 = Writethrough

## 7.4.4 System Performance Control Register (Offset 53h)

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | RAW | CRPC | CWPC | DPC | PCIMPC | Reserved | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7** **Read-Around-Write (RW)**—This feature gives read priority over write. If data is queued in the write buffer, a read request will be serviced before the write is completed.

When read-around-write is disabled, read and write requests are serviced in the order they are received.
0 = Disable (default)
1 = Enable (offset 65h, bit 0 must also be set)

**Bit 6** **Cache Read Pipeline Cycle (RW)**
**Bit 5** **Cache Write Pipeline Cycle (RW)**
**Bit 4** **DRAM Pipeline cycle (RW)**

Each of these bits enables the corresponding pipeline cycles when set. Also, NA# is asserted during pipelined cycles, but not otherwise.

0 = Disable (default)
1 = Enable

**Bit 3** **PCI Initiator Peer Concurrence (RW)**

0 = Disabled (default). The arbiter will assign the memory to the PCI port.
1 = Enabled. A PCI initiator can transfer data to a target PCI device without tying up the memory or CPU buses.

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

## 7.4.5 Non-Cacheable Region #1 (Offset 55h–54h)

| | Bit 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | A20 | A19 | A18 | A17 | A16 | R2 | R1 | R0 | A28 | A27 | A26 | A25 | A24 | A23 | A22 | A21 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

## 7.4.6 Non-Cacheable Region #2 (Offset 57h–56h)

| | Bit 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | A20 | A19 | A18 | A17 | A16 | R2 | R1 | R0 | A28 | A27 | A26 | A25 | A24 | A23 | A22 | A21 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15–11,7–0 Base Address (RW)**—Bits 15–11 and bits 7–0 determine the base address of the non-cacheable region. The default value of these bits is zero, specifying the lowest 64 Kbytes of the address range.

**Bits 10–8 Address Range R2–R0 (RW)**—Bits 10–8 specify the size of the cache area to be non-cacheable, starting from the base address. Note that the non-cacheable region must be "region aligned". For example a 128-Kbyte range must be on aligned on a 128 Kbyte boundary, a 1-Mbyte range on a 1- Mbyte boundary, etc.

000 = Disabled (default)
001 = 64 Kbytes
010 = 128 Kbytes
011 = 256 Kbytes
100 = 512 Kbytes
101 = 1 Mbyte
110 = 2 Mbytes
111 = 4 Mbytes

# 7.5 DRAM Control Registers

## 7.5.1 DRAM Configuration Register #1 (Offset 58h)

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | Memory Address Map Type for Banks 0–1 | | | Reserved | Memory Address Map Type for Banks 2–3 | | | Reserved |
| Reset | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7–5 Memory Address Map Type for Banks 0 and 1 (RW)**

**EDO/FP DRAM**

000 = 8-bit column address
001 = 9-bit column address
010 = 10-bit column address (default)
011 = 11-bit column address
100 = 12-bit column address
101 to 111 = Reserved

**SDRAM**

0xx = 16-Mbit SDRAM (default)
1xx = 64-Mbit SDRAM

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

**Bits 3–1 Memory Address Map Type for Banks 2 and 3 (RW)**

**EDO/FP DRAM**

000 = 8-bit column address (default)
001 = 9-bit column address
010 = 10-bit column address
011 = 11-bit column address
100 = 12-bit column address
101 to 111 = Reserved

**SDRAM**

0xx = 16-Mbit SDRAM (default)
1xx = 64-Mbit SDRAM

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

## 7.5.2 DRAM Configuration Register #2 (Offset 59h)

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | Memory Address Map Type for Banks 4–5 | | | Reserved | | Last DRAM bank populated | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7–5 Memory Address Map Type for Banks 4 and 5 (RW)**

**EDO/FPM DRAM**

000 = 8-bit column address (default)
001 = 9-bit column address
010 = 10-bit column address
011 = 11-bit column address
100 = 12-bit column address
101 to 111 = Reserved

**SDRAM**

0xx = 16-Mbit SDRAM (default)
1xx = 64-Mbit SDRAM
Table 7-6 on page 7-18 shows how the host address bus lines map to memory address bus lines for several DRAM configurations. Note that MA11 selects the bank for 2-bank SDRAMs and MA13 selects the bank for 4-bank SDRAMs. PC will be zero on page hits and 1 on page misses. PC is also 1 if offset 6C selects the all banks precharge command. This command is normally only used during BIOS initialization.

**Bits 4–3 Reserved (always reads 0)**

**2–0 Last DRAM Bank Populated (RW)**

000 = Bank 0
001 = Bank 1
010 = Bank 2
011 = Bank 3
100 = Bank 4
101 = Bank 5 (default)
110 to 111 = Reserved

***Note:*** *The cacheable properties of a memory address vary with its bank number, as follows:*

- *Bank 0, Bank 1 and Bank 5 are cacheable.*
- *Bank 2, Bank 3 and Bank 4 are non-cacheable if tag = 10 + modified bit.*
- *Bank 2, Bank 3 and Bank 4 are cacheable if tag is any other configuration than 10 + modify bit.*

**Table 7-6. Mapping Host Address Lines to Memory Address Lines**

**EDO/FP DRAM**

| Reg 59h Bits 7–5 | | MA13 | MA12 | MA11 | MA10 | MA9 | MA8 | MA7 | MA6 | MA5 | MA4 | MA3 | MA2 | MA1 | MA0 | Row:Col |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 000 | Row | | 23 | 22 | 21 | 11 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 12:8, 13:8 |
| | Column | | | | | | | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | |
| 001 | Row | | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 10:9, 12:9, 13:9 |
| | Column | | | | | | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | |
| 010 | Row | | 25 | 24 | 23 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11:10, 12:10, 13:10 |
| | Column | | | | | 22 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | |
| 011 | Row | | 26 | 25 | 23 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 12:11, 13:11 |
| | Column | | | | 24 | 22 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | |
| 100 | Row | | 27 | 25 | 23 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 13:12 |
| | Column | | | 26 | 24 | 22 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | |

**SDRAM**

| Reg 59h Bits 7–5 | | MA13 | MA12 | MA11 | MA10 | MA9 | MA8 | MA7 | MA6 | MA5 | MA4 | MA3 | MA2 | MA1 | MA0 | Row:Col |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0xx 16 Mbit | Row | | | 11 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11:10, 11:9, 11:8 |
| | Column | | | 11 | PC | 24 | 23 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | |
| 1xx 64 Mbit Rev C | Row | 12 | 13 | 25 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 24 | 23 | x4 (14:10) x8 (14:9) |
| | Column | 12 | 13 | 11 | PC | 26 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | |
| 1xx 64 Mbit Rev D | Row | 25 | 12 | 13 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 24 | 23 | x4 (14:10) x8 (14:9) |
| | Column | 25 | 12 | 13 | PC | 26 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | |

## 7.5.3 DRAM Bank 0 Ending Address (Offset 5Ah)
## DRAM Bank 1 Ending Address (Offset 5Bh)
## DRAM Bank 2 Ending Address (Offset 5Ch)
## DRAM Bank 3 Ending Address (Offset 5Dh)
## DRAM Bank 4 Ending Address (Offset 5Eh)
## DRAM Bank 5 Ending Address (Offset 5Fh)

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | HA29 | HA28 | HA27 | HA26 | HA25 | HA24 | HA23 | HA22 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

**Bits 7–0** **DRAM Bank Ending Address Bits 29–22 (RW)**—These registers are used to steer RAS#/CAS# lines to the correct memory banks. Each of these registers defaults to 01h. Table 7-7 illustrates several examples.

**Table 7-7. Ending Address Register Settings**

| Memory Bank | Offset | Example 1 | | Example 2 | | Example 3 | |
|---|---|---|---|---|---|---|---|
| | | Memory Size | Register Values | Memory Size | Register Values | Memory Size | Register Values |
| Bank 0 | 5Ah | 4 Mbytes | 01h | 4 Mbytes | 01h | 8 Mbytes | 02h |
| Bank 1 | 5Bh | 4 Mbytes | 02h | 8 Mbytes | 03h | 8 Mbytes | 04h |
| Bank 2 | 5Ch | 4 Mbytes | 03h | 4 Mbytes | 04h | 16 Mbytes | 08h |
| Bank 3 | 5Dh | | 03h | | 04h | | 08h |
| Bank 4 | 5Eh | | 03h | | 04h | | 08h |
| Bank 5 | 5Fh | | 03h | | 04h | | 08h |

***Notes:***

1. *The BIOS must fill the ending address register for each bank whether or not the bank is populated. The endings must be in incremental order.*

## 7.5.4 DRAM Type (Offset 60h)

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | Reserved | | Dram Type For Banks 5–4 | | Dram Type For Banks 3–2 | | Dram Type For Banks 1–0 | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

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

**Bits 5–4 DRAM Type for Banks 5–4 (RW)**
**Bits 3–2 DRAM Type for Banks 3–2 (RW)**
**Bits 1–0 DRAM Type for Banks 1–0 (RW)**

00 = Fast Page Mode(FPM) DRAM (default)
01 = Extended Data Out (EDO) DRAM
10 = Reserved
11 = Synchronous DRAM (SDRAM)

## Shadow Control Registers (Offsets 61h–63h)

A memory read/write is considered shadowed when the accessed memory segment(s) in lower memory are intercepted by the AMD-640 system controller and redirected to data copies in the upper memory area. Each 16-Kbyte segment in the UMA (64-Kbyte segments in addresses E0000h–FFFFFh) can be enabled for shadowing by setting at least one of its corresponding two bits in offsets 61h–63h. Shadowing can be enabled for read only, write only, or both.

### 7.5.5 Shadow RAM Control Register #1 (Offset 61h)

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | CC000–CFFFFh | | C8000–CBFFFh | | C4000–C7FFFh | | C0000–C3FFFh | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7–6** **Shadow RAM Control for Addresses CC000–CFFFFh (RW)**
**Bits 5–4** **Shadow RAM Control for Addresses C8000–CBFFFh (RW)**
**Bits 3–2** **Shadow RAM Control for Addresses C4000–C7FFFh (RW)**
**Bits 1–0** **Shadow RAM Control for Addresses C0000–C3FFFh (RW)**

### 7.5.6 Shadow RAM Control Register #2 (Offset 62h)

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | DC000–DFFFFh | | D8000–DBFFFh | | D4000–D7FFFh | | D0000–D3FFFh | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7–6** **Shadow RAM Control for Addresses DC000–DCFFFFh (RW)**
**Bits 5–4** **Shadow RAM Control for Addresses D8000–DBFFFh (RW)**
**Bits 3–2** **Shadow RAM Control for Addresses D4000–D7FFFh (RW)**
**Bits 1–0** **Shadow RAM Control for Addresses D0000–D3FFFh (RW)**

Each pair of bits controls the accessibility of its corresponding address range as follows:

00 = Shadowing (read and write) disabled (default)
01 = Write enabled
10 = Read enabled
11 = Read and write enabled

## 7.5.7 Shadow RAM Control Register #3 (Offset 63h)

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | E0000–EFFFFh | | F0000–FFFFFh | | Memory Hole | | SMI Redirect | VGA DRAM |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7–6 Shadow RAM Control for Addresses E0000–EFFFFh (RW)**
**Bits 5–4 Shadow RAM Control for Addresses F0000–FFFFFh (RW)**

Each pair of bits controls the accessibility of its corresponding address range as follows:

00 = Shadowing (read and write) disabled (default)
01 = Write enabled
10 = Read enabled
11 = Read and write enabled

**Bits 3–2 Memory Hole (RW)**

00 = None (default)
01 = 512 kbytes - 640 kbytes (80000–BFFFFh)
10 = 15 Mbytes - 16 Mbytes (F0000–FFFFFh)
11 = 14 Mbytes - 16 Mbytes (E0000–FFFFFh)

***Note:*** *The memory hole is used on certain legacy ISA boards and is generally not used in PCI systems.*

**Bit 1** **SMI Redirect (RW)**—Setting this bit redirects RAM accesses from graphics memory to other areas of system memory as follows:

30000h–3FFFFh is redirected to B0000h–BFFFFh.
40000h–4FFFFh is redirected to A0000h–AFFFFh.

0 = Disable redirection (default)
1 = Enable redirection.

**Bit 0** **Access to DRAM Addresses A0000–BFFFFh (RW)**—Addresses A0000h–BFFFFh are reserved for use by VGA controllers for system access to the VGA frame buffer. Setting this bit directs accesses in this range to the corresponding memory addresses in system DRAM rather than to the PCI bus for VGA frame buffer access. This feature is used to initialize B0000h–BFFFFh for SMM mode.

0 = Disable read and write to A0000h–BFFFFh (default)
1 = Enable read and write to A0000h–BFFFFh

***Note:*** *SMIACT# should be connected to the processor FLUSH# signal to avoid cache coherency problems in SMM mode.*

## 7.5.8 DRAM Timing (Offset 64h)

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | RAS# Precharge Time | | RAS# Pulse Width | | CAS# Pulse Width | | Write PW | CACAS#D |
| Reset | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |

**Bits 7-6 RAS# Precharge Time (RW)**

00 = 2T (50 ns DRAM)
01 = 3T (60 ns DRAM)
10 = 4T (70 ns DRAM) (default)
11 = 6T

**Bits 5-4 RAS# Pulse Width (RW)**

00 = 3T (50 ns DRAM)
01 = 4T (60 ns DRAM)
10 = 5T (70 ns DRAM) (default)
11 = 6T

**Bits 3-2 CAS# Pulse Width (RW)**

These two bits determine the number of CAS cycles for fast page mode and EDO DRAM, as shown in Table 7-8.

**Table 7-8. CAS# Pulse Width**

| Bits 3–2 | FPM Cycles | EDO cycles |
|---|---|---|
| 00 | 1T | 4T |
| 01 | 2T | 1T |
| 10 | 3T | 2T |
| 11 | 4T | 3T |

**Bit 1 Write Pulse Width (RW)**

0 = 1T
1 = 2T (default)

**Bit 0 Column Address to CAS# Delay (RW)**

0 = 1T
1 = 2T (default)

***Note:*** *T = 1 HCLK period.*

## 7.5.9 DRAM Control Register #1 (Offset 65h)

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | Page Mode Control | | FDDE | EDLCR | DDLD | Reserved | | DRCD |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7–6 Page Mode Control (RW)**

00 = Page closes after access (default)
01 = Reserved
10 = Page stays open after access until page time out or page miss
11 = Page closes if processor is idle, i.e., there has been no DRAM access for 8 CPU cycles

**Bit 5** **Fast DRAM Decoding Enable (RW)**—This bit should be enabled to reduce DRAM leadoff time. The timings in the diagrams presented in Section 5 refer to operations with this bit set.

0 = Disable fast DRAM decoding (default)
1 = Enable fast DRAM decoding

**Bit 4** **EDO DRAM Leadoff Cycle Reduction (RW)**—Set this bit only if system bus is 50 MHz or slower. Bit 4 has no effect unless bit 5 is set.

0 = Normal EDO DRAM leadoff cycle (default)
(Normal leadoff is 6T)
1 = Reduce EDO DRAM leadoff cycle by 1T

**Bit 3** **DRAM Data Latch Delay (RW)**—Systems that use ECC can set this bit to increase data setup time.

0 = Latch DRAM data normally (default)
1 = Delay DRAM data latch by 1/2 clock

**Bits 2–1 Reserved (always reads 0)**

**Bit 0** **DRAM Read Cycle Delay (RW)**—This bit must be set if read-around-write is enabled (offset 53h, bit 7).

0 = No delay (default)
1 = Delay DRAM read cycle 1T when write buffer is not empty

***Note:*** *T = 1 HCLK period.*

## 7.5.10 DRAM Control Register #2 (Offset 66h)

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | EDOTME | Reserved | | | | MHFC | SRPR | SRCDR |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7** **EDO Test Mode Enable (RW)**—This bit is enabled by the BIOS to detect if the DRAM is FPM or EDO. The controller delays BRDY# for 15 µsec after CAS# goes high. If the DRAM is FPM, the data on the bus will discharge during this time and read incorrectly. If it is EDO the data will be valid**.**

0 = Normal mode (default)
1 = EDO test mode

**Bits 6–3** **Reserved (always reads 0)**

**Bit 2** **Memory Data to Host Data FIFO Control (RW)**—This bit is set only if a layout problem exists on the processor bus.

0 = 0 wait states for memory data-to-host data pop (default)
1 = 1 wait state for memory data-to-host data pop

**Bit 1** **SDRAM RAS# Precharge Reduction (RW)**—This bit is only set for SDRAM. It has no effect if SDRAM is not selected in offset 60h.

0 = Normal RAS# precharge (set by bits 6 and 7 in register 64h) (default)
1 = Reduce RAS# precharge 1T for SDRAM

**Bit 0** **SDRAM RAS#-to-CAS# Delay Reduction (RW)**—This bit is only set for SDRAM. It has no effect if SDRAM is not selected in offset 60h.

0 = Normal RAS#-to-CAS# delay (2T) (default)
1 = Reduce RAS# (Active) to CAS# (Command) delay for SDRAM

***Note:*** *T = 1 HCLK period.*

## 7.5.11 32-Bit DRAM Width Control Register (Offset 67h)

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | RCAD | NA Delay | Bank 5 width | Bank 4 width | Bank 3 width | Bank 2 width | Bank 1 width | Bank 0 width |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7 RAS#-to-Column Address Delay (RW)**

0 = 1T (default)
1 = 2T

**Bit 6 NA# Delay (RW)**

0 = No NA# delay (default).
1= Delay NA# 1T.
The timing for a read hit depends on both the NA# delay and the number of PBSRAM banks, indicated by offset 51h, bit 3 (page 7-12) as shown in Table 7-9.

**Table 7-9. PBSRAM Timing**

| Offset 67h Bit 6 | Offset 51h Bit 3 | PBSRAM Read Hit Timing |
|---|---|---|
| 0 | 0 | 3-1-1-1-1-1-1-1 |
| 0 | 1 | 3-1-1-1-2-1-1-1 |
| 1 | 0 | 3-1-1-1-2-1-1-1 |
| 1 | 1 | 3-1-1-1-3-1-1-1 |

**Bit 5 Bank 5 Width (RW)**
**Bit 4 Bank 4 Width (RW)**
**Bit 3 Bank 3 Width (RW)**
**Bit 2 Bank 2 Width (RW)**
**Bit 1 Bank 1 Width (RW)**
**Bit 0 Bank 0 Width (RW)**

These bits should all be cleared.

***Note:*** *T = 1 HCLK period.*

## 7.5.12 DRAM Refresh Counter (Offset 6Ah)

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | Refresh Counter | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7–0** **Refresh Counter (RW)**—This 8-bit binary number represents the number of time units between DRAM refresh cycles. Each time unit equals 16 processor clocks. The default value is zero.

At 66 MHz, 16 clocks = 240 ns. For example, to set the refresh interval to 15.625 µsec, set this register to (15.625 µsec/240 ns) = 65 decimal = 41 hex.

***Note:*** *When this register is set to zero, DRAM refresh is disabled.*

## 7.5.13 DRAM Refresh Control Register (Offset 6Bh)

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | CBR Refresh | Burst Refresh | Reserved | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7** **CAS#-Before-RAS# Refresh (RW)**

0 = Disabled (default) (RAS#-only refresh)
1 = Enabled

**Bit 6** **Burst Refresh (RW)**

0 = Disable burst refresh (1 row refreshed every 15 µsec) (default)
1 = Enable burst refresh (4 rows refreshed every 60 µsec)

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

## 7.5.14 SDRAM Control Register (Offset 6Ch)

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | 64MBSI | SBW | SBIE | Reserved | SCL | SOMS | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7** **64 Mbit SDRAM Interleave (RW)**—This bit is relevant only for 64-Mbit SDRAM with bit 5 set.

0 = 2-bank interleave (default)
1 = 4-bank interleave

**Bit 6** **SDRAM Burst Write (RW)**

0 = Disabled (default)
1 = Enabled

**Bit 5** **SDRAM Bank Interleave Enable (RW)**—This feature increases performance by reducing the number of clocks required.

0 = Disabled (default). Timing for a 3-line burst is 8-1-1-1-3-1-1-1-3-1-1-1.
1 = Enabled. Timing for a 3-line burst is 8-1-1-1-1-1-1-1-1-1-1-1.

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

**Bit 3** **SDRAM CAS# Latency (RW)**

0 = Cycle latency is 2 (default)
1 = Cycle latency is 3

***Note:*** *It is possible to program CAS latencies of 1, 2, or 3 into any SDRAM bank regardless of the value set by bit 3. Programming a different CAS latency value in memory than the value implemented by this register can result in miscommunication.*

**Bits 2–0** **SDRAM Operation Mode Select (RW)**—These commands are used in the SDRAM detection algorithm. Refer to the BIOS porting guide.

000 = normal SDRAM mode (default)
001 = NOP command enabled
010 = CPU-to-DRAM cycles are converted to All Banks Precharge commands
011 = CPU-to-DRAM cycles are converted to commands driven on MA[11:0]. The BIOS selects an appropriate host address for each memory row of memory to generate the appropriate commands.
100 = CAS#-Before-RAS# cycle enable
101 to 111 = Reserved

## 7.5.15 DRAM Drive Strength Control Register (Offset 6Dh)

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | BDT | MA[1:0] Drive | MA/RAS# | FORCE SMM | SDRAMCD | MA&WED | CAS# Drive | RAS# Drive |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7 Bank Decoding Test (RW)**

0 = Normal operation (default)
1 = For production test only. DO NOT SET.

**Bit 6 MA[1:0] Drive (RW)**

0 = 12 mA (default)
1 = 24 mA

**Bit 5 MA[1:0]/RAS[5:4]# (RW)**—This pin determines whether pins N17 and M17 function as RAS[5:4]# or secondary drivers for MA[1:0] as shown in Table 7-10.

**Table 7-10. Functions of Pins N17 and M17**

| Bit 5 | Pin N17 Function | Pin M17 Function | Drive Control |
|---|---|---|---|
| 0 | RAS5# | RAS4# | Reg. 6Dh, bit 0 |
| 1 | MA1 | MA0 | Reg. 6Dh, bit 6 |

**Bit 4 Force SMM Mode (RW)**—When this bit is set the AMD-640 system controller responds as if the SMIACT# pin was asserted.

0 = SMM mode not forced (default)
1 = SMM mode forced.

**Bit 3 SDRAM command Drive (SRAS#, SCAS#, WEx#) (RW)**

0 = 12 mA (default)
1 = 24 mA

**Bit 2 MA[13:2] and WEx# Drive (RW)**

0 = 12 mA (default)
1 = 24 mA

**Bit 1 CAS# Drive (RW)**

0 = 12 mA (default)
1 = 24 mA

**Bit 0 RAS# Drive (RW)**

0 = 12 mA (default)
1 = 24 mA

## 7.5.16 ECC Control Register (Offset 6Eh)

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | ENMS | Reserved | ESEE | ESEE | ECTPDR | EEE5/4 | EE3/2 | EEE1/0 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7 ECC/Normal Mode Select (RW)**

0 = Parity (default)
1 = ECC

**Bit 6 Reserved (always reads 0)**

**Bit 5 Enable SERR# on ECC (Multi-bit) Error (RW)**

0 = Don't assert SERR# for ECC errors (default)
1 = Assert SERR# for ECC errors

**Bit 4 Enable SERR# on ECC (Single-bit) Error (RW)**

0 = Do not assert SERR# for ECC errors (default)
1 = Assert SERR# for ECC errors

**Bit 3 ECC Cycle Timing for CPU-SDRAM Reads (RW)**

0 = Normal CPU SDRAM reads (default)
1 = Add 1T for CPU SDRAM read cycles with ECC. The extra cycle is required if ECC mode is enabled.

**Bit 2 ECC Enable — Bank 5/4 (RW)**
**Bit 1 ECC Enable — Bank 3/2 (RW)**
**Bit 0 ECC Enable — Bank 1/0 (RW)**

0 = Disabled (default)
1 = Enabled

## 7.5.17 ECC Status Register (Offset 6Fh)

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | MBED | Multi-Bit Error DRAM Bank | | | SBE | Single-Bit Error DRAM Bank | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7** **Multi-bit Error Detected (RWC)**

Write 1 to this bit to clear it.

**Bits 6–4** **Multi-Bit Error DRAM Bank (RWC)**—These bits contain the encoded value of the DRAM bank containing the multi-bit error. Write 1's to these bits to clear them.

000 = Bank 1 (default)
001 = Bank 2
010 = Bank 3
011 = Bank 4
100 = Bank 5
101 = Bank 6

**Bit 3** **Single-Bit ECC Error (RWC)**—Write 1 to this bit to clear it. Single bit errors are corrected but not written back to memory.

**Bits 2–0** **Single-Bit Error DRAM Bank (RWC)**—These bits contain the encoded value of the DRAM bank containing the single-bit error. Write 1's to these bits to clear them.

000 = Bank 1 (default)
001 = Bank 2
010 = Bank 3
011 = Bank 4
100 = Bank 5
101 = Bank 6

# 7.6 PCI Bus Control Registers

## 7.6.1 PCI Buffer Control Register (Offset 70h)

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | CPPW | PIDPW | PIDP | Reserved | | | PRPQWA | NOFLUSH |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7** **CPU-to-PCI Post-Write (RWC)**

0 = Disabled (default)
1 = Enabled

**Bit 6** **PCI Initiator-to-DRAM Post-Write (RW)**

0 = Disabled (default)
1 = Enabled

**Bit 5** **PCI Initiator-to-DRAM Prefetch (RW)**

0 = Disabled (default)
1 = Enabled

**Bits 4–2** **Reserved (always reads 0)**

**Bit 1** **PCI Retry for Processor QW (Quadword) Access (RW)**—By default, the controller backs off the processor if the second doubleword of a PCI access is delayed, and starts over with the first dword, potentially resulting in a system deadlock. Clearing bit 1 avoids this hazard by preventing the controller from asserting BOFF# during the second dword access. This feature should always be enabled.

0 = Enabled (recommended)
1 = Disabled (default)

**Bit 0** **Disable Flush of CPU-to-PCI Buffer (RW)**—By default, the AMD-640 system controller writes all contents in the PCI write buffer (i.e., flushes the buffer) before granting the bus to another PCI initiator. This feature allows the AMD-640 system controller to grant the PCI bus to another initiator before the write buffer is emptied. Enabling this feature reduces grant latency.

0 = Flushing CPU-to-PCI buffer has priority (default)
1 = Grant to another PCI initiator has priority

## 7.6.2 Processor-to-PCI Flow Control Register #1 (Offset 71h)

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | BURST2 | Byte Merge | Reserved | PIOCPW | BURST1 | EPFBBW | EQFG | E1WSPC |
| Reset | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

**Bits 7 & 3** **PCI Burst Control Bits (RW)**—These two bits determine how the AMD-640 system controller processes CPU-to-PCI write transactions, as shown in Table 7-11.

**Table 7-11. PCI Burst Control Bits**

| Bit 7 | Bit 3 | Operation |
|---|---|---|
| 0 | 0 | Every write transaction goes to the write buffer.<br>No burst operations occur.<br>This is the default setting. |
| 0 | 1 | Burst writes are placed in the write buffer.<br>Burst operations are later performed on the PCI bus.<br>Non-burst writes are immediately written to the PCI bus after the write buffers are flushed. |
| 1 | x | Every write transaction is placed in the write buffer.<br>Burst operations are performed on burstable transactions.<br>Regular PCI writes are performed on non-burstable transactions.<br>This is the setting for normal operation. |

**Bit 6** **Byte Merge (RW)**—Setting this bit enables the AMD-640 system controller to collect bytes and write them as a word or doubleword.

0 = Disabled (default)
1 = Enabled

**Bit 5** **Reserved (always reads 1)**

**Bit 4** **PCI I/O Cycle Post Write (RW)**—Enabling this feature allows the CPU to proceed after posting a PCI write. This is the preferred setting. If this feature is disabled, the processor is held in a wait state until the PCI write is completed.

0 = Disabled (default)
1 = Enabled

**Bit 2** **Enable PCI Fast Back-to-Back Write (RW)**—See Section 5.4.7 on page 5-52 for a discussion of fast back-to-back reads.

0 = Disabled (default)
1 = Enabled (generates NA#)

**Bit 1** **Enable Quick Frame Generation (RW)**—When this bit is set, FRAME# is generated one PCI clock earlier than in standard PCI operation. This is the recommended setting.

0 = Disabled (default)
1 = Enabled (recommended)

**Bit 0** **Enable 1-Wait-State PCI Cycles (RW)**—The AMD-640 system controller delays assertion of IRDY# one clock cycle when this bit is set.

0 = Disabled (default)
1 = Enabled

## 7.6.3 Processor-to-PCI Flow Control Register #2 (Offset 72h)

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | Retry Status | RTA | Retry Count and Backoff | | CFDCR | PBPRRF | R1TFG | R1TPRPT |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7** **Retry Status (RWC)**—This bit indicates that a CPU-to-PCI transaction has been retried unsuccessfully either 16 or 64 times (see Bits 5–4). Write a 1 to this bit position to clear the bit.

0 = No retry has occurred (default)
1 = Retry has occurred

**Bit 6** **Retry Timeout Action (RW)**

0 = Retry continuously and record status only (default)
1 = Flush buffer and return FFFFFFFFh for read

**Bits 5–4** **Retry Count and Retry Backoff (RW)**

00 = Retry two times and backoff processor (default)
01 = Retry 16 times and set Retry Status (Bit 7)
10 = Retry 4 times and backoff processor
11 = Retry 64 times and set Retry Status (Bit 7)

**Bit 3** **Clear Failed Data And Continue Retry (RW)**—The post write buffer stores data going from the CPU to the PCI bus. If the target is not ready to accept the data a retry will occur. If the cycle fails to complete after the number of retry attempts specified in bits 5–4, the data will be discarded (popped) if bit 3 is set. This makes room in the post write buffer to accept new data from the processor.

0 = Disable (default)
1 = Flush (pop) the failed data and continue posting when posting retries fail

**Bit 2** **Processor Backoff on PCI Read Retry Failure (RW)**—This feature generates BOFF# when a PCI read retry fails, momentarily boosting priority of the PCI.

0 = Disabled (default)
1 = Enabled

**Bit 1** **Reduce 1T for FRAME# Generation (RW)**—When this bit is set, FRAME# is generated one PCI clock earlier than the setting in register 71h, bit 1. Doing so may cause timing problems and is not recommended.

0 = Disabled (default) (recommended)
1 = Enabled

**Bit 0** **Reduce 1T for Processor Read of PCI Target (RW)**—Setting this bit reduces the delay from TRDY# to BRDY# by one HCLK to speed up system performance.

0 = Disabled (default)
1 = Enabled

## 7.6.4 PCI Target Control Register (Offset 73h)

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | LMD | PT1WSW | PT1WSR | Reserved | ASAWT | ASART | LOCK | PIBTE |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7** **Local Memory Decoding (RW)**—This bit must be set if fast back-to-back cycles are selected in either register 04h-05h or register 71h**.**

0 = Fast (address phase) (default)
1 = Slow (first data phase)

**Bit 6** **PCI Target 1 Wait State Read (RW)**

0 = Zero wait state TRDY# response (default)
1 = One wait state TRDY# response

**Bit 5** **PCI Target 1 Wait State Write (RW)**

0 = Zero wait state TRDY# response (default)
1 = One wait state TRDY# response

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

**Bit 3** **Assert STOP# After Write Timeout (RW)**—Enabling this feature allows the AMD-640 system controller to signal a retry to the initiator by asserting STOP#. This is the recommended setting.

0 = Disabled (default)
1 = Enabled (recommended)

**Bit 2** **Assert STOP# After Read Timeout (RW))**—Enabling this feature allows the AMD-640 system controller to signal a retry to the initiator by asserting STOP#. This is the recommended setting.

0 = Disabled (default)
1 = Enabled (recommended)

**Bit 1** **LOCK# Function (RW)**—When this bit is enabled, the AMD-640 system controller samples the LOCK# pin on the PCI bus and reserves the resource per the PCI specification. The controller does not assert LOCK# during CPU- to-PCI cycles.

0 = Disabled (default)
1 = Enabled

**Bit 0** **PCI Initiator Broken Timer Enable (RW)**—Setting this bit forces the AMD-640 system controller to initiate PCI arbitration in the event that FRAME# has not been asserted 16 PCI clocks after the last GNT# was issued.

0 = Disabled (default)
1 = Enabled

## 7.6.5 PCI Initiator Control Register (Offset 74h)

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | PECS | PISWM | Reserved | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7** **PCI Enhanced Command Support (RW)**—Setting this bit improves performance by enabling the Memory Read Line (MRL), Memory Read Multiple (MRM), and Memory Write and Invalidated (MWI) PCI commands. MRL and MRM make DRAM access more efficient by enabling burst accesses to the DRAM. MWI prevents unnecessary snoops on the processor bus.

0 = Disabled (default)
1 = Enabled

**Bit 6** **PCI Initiator Single Write Merge (RW)**—Enabling this function reduces PCI bus traffic to improve bus utilization. This is accomplished by collecting bytes or words and forming them into one doubleword. For example, if the processor performs four consecutive byte writes, they will be combined into one 32-bit transfer on the PCI bus.

0 = Disabled (default)
1 = Enabled

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

## 7.6.6 PCI Arbitration Control Register #1 (Offset 75h)

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | ARBPRI | ARBMODE | Reserved | | PCI Initiator Bus Time-Out | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7** **Arbitration Priority (RW)**—Setting bit 7 forces fair arbitration between PCI initiators and the processor employing round-robin (rotating) arbitration. By default, the prioity order is fixed as follows:

1. REQ1#
2. REQ2#
3. REQ3#
4. REQ4#
5. PREQ#
6. CPU

0 = Fixed priority (default)
1 = Fair arbitration

**Note:** If register 76h bit 7 is set, the processor has higher priority than the PCI bus, and its operation overrides the priority selected by register 75h bit 7.

**Bit 6** **Arbitration Mode (RW)**

0 = REQ#-based (arbitrate at end of REQ#) (default)
1 = FRAME#-based (arbitrate at end of each FRAME#). This enables a higher priority initiator to preempt a lower priority initiator.

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

**Bits 3–0** **PCI Initiator Bus Timeout (RW)**—Bits 3–0 represent the binary number of idle time periods the AMD-640 system controller allows on the PCI bus before forcing arbitration. Each time period is equal to 32 PCI clock cycles. The default value of 0000h disables this feature.

## 7.6.7 PCI Arbitration Control Register #2 (Offset 76h)

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | IPRE | Reserved | IPRC | | reserved | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7 Initiator Priority Rotation Enable (RW)**

0 = Disabled (arbitration per register 75h, bit 7) (default)
1 = Enabled (arbitration per bits 5–4 below)

**Bit 6 Reserved (always reads 0)**

**Bits 5–4 Initiator Priority Rotation Control (RW)**

00 = Disabled (arbitration per register 75h bit 7) (default)
01 = Grant to processor after every PCI initiator grant. The processor is guaranteed access to the PCI bus after the current PCI initiator completes, regardless of the number or priority of other requesting PCI initiators.
10 = Grant to processor after every two PCI initiator grants. The processor is guaranteed access to the PCI bus after the current PCI initiator and one more PCI initiator complete.
11 = Grant to processor after every three PCI initiator grants. The processor is guaranteed access to the PCI bus after the current PCI initiator and two more PCI initiators complete.

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

# 8 Electrical Data

## 8.1 Absolute Ratings

Long-term reliability and functional integrity of the AMD-640 system controller are guaranteed as long as it is not subjected to conditions exceeding the absolute ratings listed in Table 8-1.

**Table 8-1. Absolute Ratings**

| Parameter | Minimum | Maximum | Comments |
|---|---|---|---|
| $V_{DD}$ | -0.5 V | 5.5 V | Core Supply |
| $V_{DD3}$ | -0.5 V | 3.6 | I/O Supply |
| $V_{PIN}$ (Processor) | -0.5 V | $V_{DD3}$+0.5 V or 4.0 V, whichever is lower | Note 1 |
| $V_{PIN}$ (PCI and DRAM) | -0.5 V | $V_{DD}$ + | Note 2 |
| $T_{CASE}$ (under bias) | -65°C | +110°C | |
| $T_{STORAGE}$ | -65°C | +150°C | |

***Notes:***

1. *The voltage on any I/O pin on the CPU interface 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*
2. *The voltage on any I/O pin on the PCI or DRMA interface must not be greater than 0.5 V above the voltage being applied to $V_{DD3}$*

# 8.2 Operating Ranges

The functional operation of the AMD-640 system controller is guaranteed if the voltage and temperature parameters are within the limits defined in Table 8-2.

**Table 8-2. Operating Ranges**

| Parameter | Minimum | Typical | Maximum | Comments |
|---|---|---|---|---|
| $V_{DD}$ | 4.75 V | 5.0 V | 5.25 V | Core (Note 1) |
| $V_{DD3}$ | 3.135 V | 3.3V | 3.6 V | I/O (Note 1) |
| $T_{CASE}$ | 0°C | | 70°C | |

***Notes:***

*1. $V_{DD}$ and $V_{DD3}$ are referenced from $V_{SS}$.*

# 8.3 DC Characteristics

**Table 8-3. 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 (Processor) | 2.0 V | $V_{DD3}$+0.5 V | Note 1 |
| $V_{IH}$ | Input High Voltage (PCI and DRAM) | 2.0 V | $V_{DD}$+0.5 V | |
| $V_{OL}$ | 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 | | 0.40 A | 66 MHz, Note 2 |
| $I_{DD3}$ | 3 V Power Supply Current | | 0.35 A | 66 MHz, Note 3 |
| $I_{LI}$ | Input Leakage Current | | ±10 µA | Note 4 |
| $I_{LO}$ | Output Leakage Current | | ±20 µA | Note 4 |
| $I_{IL}$ | Input Leakage Current Bias with Pullup | | 40 µA | Note 5 |
| $I_{IH}$ | Input Leakage Current Bias with Pulldown | | -40 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, 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. *$V_{DD2}$ = 5.25 V – The maximum power supply current must be taken into account when designing a power supply.*
3. *$V_{DD3}$ = 3.6 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 $V_{IL}$ = 0.4V.*
6. *Refers to inputs with an internal pulldown and $V_{IH}$ = 2.4V.*

# 8.4 Power Dissipation

Table 8-4 shows typical and maximum power dissipation of the AMD-640 system controller during normal and reduced power states. The measurements are taken with HCLK = 66 MHz, $V_{DD}$ = 5.0 V, and $V_{DD3}$ = 3.3 V.

**Table 8-4. Typical and Maximum Power Dissipation**

| Clock Control State | Typical (Note 1) | Maximum (Note 2) | Comments |
|---|---|---|---|
| Normal (Thermal Power) | 1.8 W | 2.5 W | Note 3 |
| Stop Grant /Halt | | 0.21 W | Note 4 |
| Stop Clock | | 0.175 W | Note 5 |

***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-640 system controller processor.*
4. *The CLK signal and the internal PLL are still running but most internal clocking has stopped.*
5. *The CLK signal, the internal PLL, and all internal clocking has stopped.*

# 9 Switching Characteristics

The AMD-640 system controller signal switching characteristics are presented in Tables 9-1 through 9-7. 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 (HCLK, PCLK, or RESET) passes through 1.5V to the time the target signal passes through 1.5V.
- All signal slew rates are 1 V/ns, from 0V to 3V (rising) or 3V to 0V (falling).
- Parameters are within those listed in "Operating Ranges" on page 8-2.
- The load capacitance ($C_L$) on each signal is 0 pF with the exception of maximum timings for clock, processor, DRAM and cache, where the $C_L$ = 50 pf.

# 9.1 CLK Switching Characteristics

Table 9-1 and Table 9-2 contain the switching characteristics of the HCLK input to the AMD-640 system controller for 66-MHz and 60-MHz CPU bus operation, respectively. Table 9-3 contains the switching characteristics of the PCLK input for 33-MHz PCI bus operation. These timings are all measured with respect to the voltage levels indicated by Figure 9-1 on page 9-3.

The CLK period stability specifies the variance (jitter) allowed between successive periods of the CLK input measured at 1.5V. This parameter must be considered as one of the elements of clock skew between the AMD-640 system controller and the system logic.

**Table 9-1. HCLK Switching Characteristics for 66-MHz Bus Operation**

| Symbol | Parameter Description | Preliminary Data | | Figure | Comments |
|---|---|---|---|---|---|
| | | Min | Max | | |
| | Frequency | 33.3 MHz | 66.6 MHz | | |
| $t_2$ | HCLK high time | 6.0 ns | | 9-1 | |
| $t_3$ | HCLK low time | 6.0 ns | | 9-1 | |
| $t_4$ | HCLK fall time | 0.15 ns | 1.5 ns | 9-1 | |
| $t_5$ | HCLK rise time | 0.15 ns | 1.5 ns | 9-1 | |
| | HCLK period stability | | ± 250 ps | | Note 1 |

***Notes:***
*1. Jitter frequency power spectrum peaking must occur at frequencies greater than (HCLK frequency)/3 or less than 500 KHz.*

**Table 9-2. HCLK Switching Characteristics for 60-MHz Bus Operation**

| Symbol | Parameter Description | Preliminary Data | | Figure | Comments |
|---|---|---|---|---|---|
| | | Min | Max | | |
| | Frequency | 30 MHz | 60 MHz | | |
| $t_2$ | HCLK high time | 6.0 ns | | 9-1 | |
| $t_3$ | HCLK low time | 6.0 ns | | 9-1 | |
| $t_4$ | HCLK fall time | 0.15 ns | 1.5 ns | 9-1 | |

***Notes:***
*1. Jitter frequency power spectrum peaking must occur at frequencies greater than (Frequency of CLK)/3 or less than 500 KHz.*

**Table 9-2. HCLK Switching Characteristics for 60-MHz Bus Operation (continued)**

| Symbol | Parameter Description | Preliminary Data | | Figure | Comments |
|---|---|---|---|---|---|
| | | Min | Max | | |
| $t_5$ | HCLK rise time | 0.15 ns | 1.5 ns | 9-1 | |
| | HCLK period stability | | ± 250 ps | | Note 1 |

***Notes:***
*1. Jitter frequency power spectrum peaking must occur at frequencies greater than (Frequency of CLK)/3 or less than 500 KHz.*

**Table 9-3. PCLK Switching Characteristics for 33-MHz PCI Bus**

| Symbol | Parameter Description | Preliminary Data | | Figure | Comments |
|---|---|---|---|---|---|
| | | Min | Max | | |
| t1 | PCLK cycle | 30 ns | ∞ | | |
| $t_2$ | PCLK high time | 11.0 ns | | 9-1 | |
| $t_3$ | PCLK low time | 11.0 ns | | 9-1 | |
| $t_4$ | PCLK fall time | 1 V/ns | 4V/ns | 9-1 | |
| $t_5$ | PCLK rise time | 1 V/ns | 4V/ns | 9-1 | |
| | PCLK period stability | | ± 250 ps | | Note 1 |

***Notes:***
*1. Jitter frequency power spectrum peaking must occur at frequencies greater than (HCLK frequency)/3 or less than 500 KHz.*

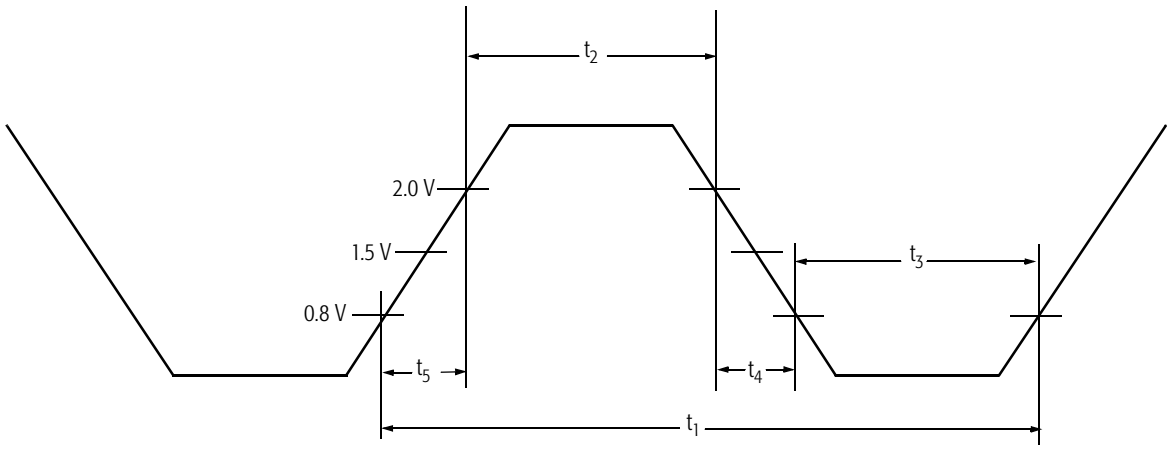



**Figure 9-1. CLK Waveform**

# 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-640 system controller input signals presented here must be met by any device that interfaces with it to assure the proper operation of the AMD-640 system controller.

Figure 9-2 illustrates the relationship between the rising clock edge and setup, hold, and valid data timings.

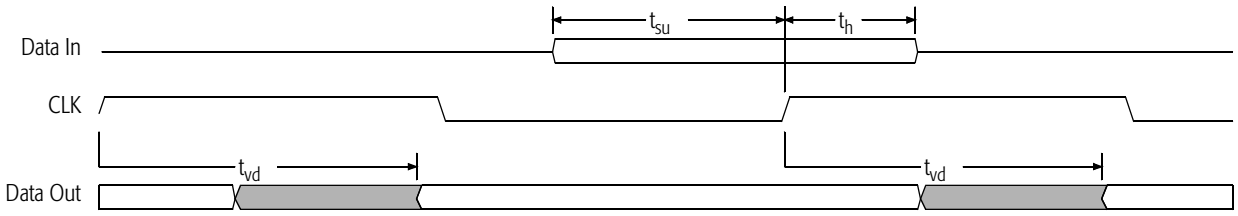



**Figure 9-2. Setup, Hold, and Valid Delay Timing Diagram**

# 9.3 Processor Interface Timing

All of the following timings are relative to HCLK.

Pads 0 to 13 are driver types. For system simulation, select the pad from the appropriate IBIS model.

**Table 9-4. Processor Cycle Timing**

| Symbol | Parameter Description | Preliminary Data | | Figure | Comments |
|---|---|---|---|---|---|
| | | Min | Max | | |
| $t_{su}$ | Setup time for<br>ADS# DC#<br>HLOCK# CACHE#<br>SMIACT# BE[7:0]#<br>HA[31:3] | 5 ns | | 9-2 | |
| $t_{su}$ | W/R# | 5.5 ns | | | |
| $t_{su}$ | Setup time for<br>M/IO# HITM# | 6 | | 9-2 | |
| $t_{su}$ | Setup time for<br>HD[63:0] | 3 ns | | 9-2 | |
| $t_h$ | Hold time for<br>SMIACT# HITM#<br>W/R# CACHE#<br>M/IO# BE[7:0]#<br>D/C# HA[31:3] | 1 ns | | 9-2 | |
| $t_h$ | ADS# hold time | 1.5 ns | | 9-2 | |
| $t_h$ | HLOCK# hold time | 1.0 ns | | 9-2 | |
| $t_h$ | HD[63:0] hold time | 2 ns | | 9-2 | |
| $t_{vd}$ | Valid delay for<br>AHOLD BOFF#<br>KEN#/INV EADS# | 1.5 ns | 7 ns | 9-2 | Pad 0 (note 1) |
| | BRDY#, NA# valid delay | 1.5 ns | 8 ns | 9-2 | Pad 0 (note 1) |
| | HA[31:3] valid delay | 2 ns | 13 ns | 9-2 | Pad 2 (note 1) |
| | HD[63:0] Valid delay | 1.5 ns | 8.5 ns | 9-2 | Pad 1 (note 1) |
| $t_{fd}$ | HA[31:3] float delay | 4 ns | 9 ns | 9-2 | Note 1 |

***Notes:***

1. *Measurements are taken with no load.*

# 9.4 PCI Interface Timing

All of the following timings are relative to PCLK.

**Table 9-5. PCI Interface Timing**

| Symbol | Parameter Description | Preliminary Data | | Figure | Comments |
|---|---|---|---|---|---|
| | | Min | Max | | |
| $t_{su}$ | AD[31:0] setup time | 7 ns | | 9-2 | |
| | PREQ#, REQ[3:0]# setup time | 12 ns | | 9-2 | |
| | Setup time for<br>FRAME# STOP#<br>TRDY# DEVSEL#<br>IRDY# C/BE[3:0]#<br>RESET# | 7 ns | | 9-2 | |
| $t_h$ | AD[31:0] hold time | 0 ns | | 9-2 | |
| | Hold time for<br>FRAME# STOP#<br>TRDY# DEVSEL#<br>IRDY# C/BE[3:0]#<br>PREQ# REQ[3:0]# | 0 ns | | 9-2 | |
| $t_{vd}$ | AD[31:0] Valid Delay (address phase) | 2 ns | 11 ns | 9-2 | Pad 12 (note 1) |
| | AD[31:0] valid delay (data phase) | 2 ns | 11 ns | 9-2 | Pad 12 (note 1) |
| | Valid delay for<br>FRAME# STOP#<br>TRDY# DEVSEL#<br>IRDY# C/BE[3:0]#<br>GNT[3:0]# | 2 ns | 11 ns | 9-2 | Pad 13 (note 1) |
| | PGNT# valid delay | 2 ns | 12 ns | 9-2 | |
| $t_{fd}$ | Float delay for<br>FRAME# STOP#<br>TRDY# DEVSEL#<br>IRDY# C/BE[3:0]# | | 28 ns | 9-2 | (note 1) |
| $t_{pw}$ | RESET# pulse width | 2 clks | | 9-2 | |
| $t_{lat}$ | REQ# to GNT# latency | 3 clks | | 9-2 | |

***Notes:***

1. *Measurements are taken with no load for $t_{min}$, 50 pF for $t_{max}$.*

# 9.5 DRAM Interface Timing

All of the following timings are relative to HCLK.

**Table 9-6. DRAM Interface Timing**

| Symbol | Parameter Description | Preliminary Data | | Figure | Comments |
|---|---|---|---|---|---|
| | | Min | Max | | |
| $t_{su}$ | MD[63:0] setup (SDRAM) | 2 ns | | 9-2 | |
| | MD[63:0] setup (EDO/FP) | 2 ns | | 9-2 | |
| $t_h$ | MD[63:0] hold (SDRAM) | 2.5 ns | | 9-2 | |
| | MD[63:0] hold (EDO/FPM) | 4 ns | | 9-2 | |
| $t_{vd}$ | RAS[5:0]# valid delay (EDO) | 1.5 ns | 8 ns | 9-2 | Pad 5 (note 1) |
| | CS[5:0]# valid delay (SDRAM) | 2 ns | 8 ns | 9-2 | Pad 5 (note 1) |
| | CAS[7:0]# valid delay (EDO) | 1.5 ns | 8 ns | 9-2 | Pad 7 (note 1) |
| | DQM[7:0]# valid delay (SDRAM) | 2ns | 7 ns | 9-2 | Pad 11 (note 1) |
| | SRAS[C:A]# valid delay (SDRAM) | 2 ns | 7 ns | 9-2 | Pad 5 (note 1) |
| | SCAS[C:A]# valid delay (SDRAM) | 2 ns | 7 ns | 9-2 | Pad 5 (note 1) |
| | WE[C:A]# valid delay | 2 ns | 7 ns | 9-2 | Pad 5 (note 1) |
| | MA[13:2] valid delay | 2 ns | 10 ns | 9-2 | Pad 5 (note 1) |
| | MA[1:0] valid delay | 2 ns | 10 ns | 9-2 | Pad 5 (note 1) |
| | MD[63:0] valid delay (SDRAM) | 2 ns | 7 ns | 9-2 | Pad 1 (note 1) |
| | MD[63:0] valid delay (EDO/FPM) | 2 ns | 12 ns | 9-2 | Pad 1 (note 1) |
| $t_{ft}$ | MA[11:0] Flow-through delay from HA for first read cycle | 2 ns | 10 ns | 9-2 | Pad 5 (note 1) |

***Notes:***

1. *Measurements are taken with no load.*

# 9.6 L2 Cache Timing

All of the following timings are relative to HCLK.

**Table 9-7. L2 Cache Timing**

| Symbol | Parameter Description | Preliminary Data | | Figure | Comments |
|---|---|---|---|---|---|
| | | Min | Max | | |
| $t_{su}$ | TA9–TA0 setup time | 6.7 ns | | 9-2 | |
| $t_h$ | TA9–TA0 hold time | 2 ns | | 9-2 | |
| $t_{vd}$ | TA9–TA0 valid delay | 2 ns | 9 ns | 9-2 | Pad 1 (note 1) |
| | Valid delay for CE1 CADS CADV | 1.5ns | 7 ns | 9-2 | Pad 0 (note 1) |
| | COE, TAGWE valid delay | 1.5 ns | 10 ns | | Pad 0 (note 1) |
| | GWE, BWE valid delay | 1.5 ns | 9 ns | | Pad 0 (note 1) |

***Notes:***
1. *Measurements are taken with no load.*

# 10 IBIS Models

All of the AMD-640 system 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. These buffers can be configured to provide the higher drive strength for applications that place a heavier load on these I/O signals.

AMD has developed several I/O buffer models that represent the characteristics of each of the possible drive strength configurations supported by the AMD-640 system controller.

AMD developed the models to allow system designers to perform analog simulations of AMD-640 system 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 the system's signal quality requirements are met. Signal quality measurements include overshoot, undershoot, slope reversal, and ringing.

## 10.1 Selectable Drive Strength

The driver types are specified in the AC table. The model also associates the appropriate driver type to its respective pin. Only the memory drivers are programmable by configuration registers. Hence the designer must select the 12-ma or-24 ma driver, depending on the intended use.

# 10.2 I/O Buffer Model

AMD provides models of the AMD-640 system 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 to model the of I/O buffer behavior accurately.

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 sufficient data points for accurate representation of 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-640 system controller to accommodate simulators that can yield more accurate results based on this wider range.
- Rising and falling ramp rates are specified.
- The min/typ/max $V_{DD3}$ 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.

## 10.3 I/O Model Application Note

The AMD-640 system controller I/O Buffer IBIS Models and their applications can be found in the *AMD-640 System Controller I/O Model (IBIS) Application Note*, order# (tbd).

The Model is available at http://www.amd.com

## 10.4 I/O Buffer AC and DC Characteristics

Refer to Section 9 for the AMD-640 system controller AC timing specifications.

Refer to Section 8 for the AMD-640 system controller DC specifications.

## 10.5 References

Ease System Simulation With IBIS Device Models by Syed Huq, *Electronics Design*, December 2, 1996

*IBIS 2.1 Specification* at http://vhdl.org/

*IBIS Forum I/O Buffer Modeling Cook Book*

# 11 Pin Descriptions

## 11.1 Electrical Considerations

**Reset**

During reset, bidirectional pins are tri-stated, and outputs are driven to their inactive state.

**Configuration**

Certain PCI pins require pullup resistors (recommended value 8.2 Kohms). Their names and pin numbers are as follows:

- DEVSEL# K-2
- FRAME# L-3
- IRDY# K-1
- LOCK# K-3
- PAR# J-3
- REQ0# C-2
- REQ1# E-5
- REQ2# A-1
- REQ3# C-1
- SERR# J-1
- STOP# H-5
- TRDY# K-4

## 11.2 Pin Numbering

The following tables list the AMD-640 system controller pin names and their corresponding pin numbers. Table 11-1 groups the pins by function. Table 11-2 presents the pins as they appear on the 328-pin ball grid array.

## Table 11-1. Functional Grouping

| Host Address | | Host Data | | Host Control | | PCI Addr/Data | | PCI Control | | Cache Interface | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Pin Name | Pin No. | Pin Name | Pin No. | Pin Name | Pin No. | Pin Name | Pin No. | Pin Name. | Pin No. | Pin Name | Pin No. |
| HA3 | V-11 | HD0 | Y-1 | ADS# | T-5 | AD0 | A-15 | C/BE0# | B-11 | BWE# | Y17 |
| HA4 | Y-11 | HD1 | Y-3 | AHOLD | L-5 | AD1 | A-14 | C/BE1# | A-5 | CADS# | W15 |
| HA5 | U-8 | HD2 | Y-2 | | | AD2 | B-13 | C/BE2# | A-8 | CADV# | V15 |
| HA6 | Y-12 | HD3 | Y-4 | BE0# | K-2 | AD3 | A-13 | C/BE3# | B-8 | CE1# | W16 |
| HA7 | Y-8 | HD4 | W-1 | BE1# | K-3 | AD4 | C-12 | | | COE# | U15 |
| HA8 | V-8 | HD5 | W-2 | BE2# | K-4 | AD5 | B-12 | DEVSEL# | E-9 | GWE# | Y16 |
| HA9 | V-7 | HD6 | V-1 | BE3# | L-1 | AD6 | A-12 | FRAME# | E-6 | TAGWE# | V16 |
| HA10 | Y-7 | HD7 | R-1 | BE4# | L-2 | AD7 | C-11 | | | | |
| HA11 | W-7 | HD8 | U-1 | BE5# | L-3 | AD8 | A-11 | GNT0# | D-8 | TA9 | Y19 |
| HA12 | U-7 | HD9 | W-3 | BE6# | L-4 | AD9 | C-10 | GNT1# | D-10 | TA8 | W19 |
| HA13 | Y-6 | HD10 | V-3 | BE7# | M-1 | AD10 | B-10 | GNT2# | D-12 | TA7 | Y18 |
| HA14 | V-6 | HD11 | W-4 | | | AD11 | A-10 | GNT3# | D-14 | TA6 | W18 |
| HA15 | W-6 | HD12 | T-1 | BOFF# | P-5 | AD12 | C-9 | | | TA5 | V18 |
| HA16 | U-6 | HD13 | V-4 | BRDY# | M-5 | AD13 | B-9 | IRDY# | E-7 | TA4 | T14 |
| HA17 | Y-5 | HD14 | U-3 | CACHE# | J-5 | AD14 | A-9 | LOCK# | E-5 | TA3 | V17 |
| HA18 | W-5 | HD15 | V-2 | D/C# | T-7 | AD15 | C-8 | PAR | E-12 | TA2 | U17 |
| HA19 | V-5 | HD16 | U-4 | EADS# | R-5 | AD16 | C-7 | PCLK | E-10 | TA1 | U16 |
| HA20 | U-5 | HD17 | T-3 | HITM# | T-8 | AD17 | B-7 | PGNT# | D-5 | TA0 | P16 |
| HA21 | V-9 | HD18 | U-2 | HCLK | K-16 | AD18 | A-7 | PREQ# | D-4 | | |
| HA22 | U-10 | HD19 | R-3 | HLOCK# | G-5 | AD19 | C-6 | | | | |
| HA23 | U-P | HD20 | T-4 | KEN#/INV | K-5 | AD20 | B-6 | REQ0# | D-7 | | |
| HA24 | W-10 | HD21 | R-2 | M/IO# | H-5 | AD21 | A-6 | REQ1# | D-9 | | |
| HA25 | W-9 | HD22 | T-2 | NA# | N-5 | AD22 | C-5 | REQ2# | D-11 | | |
| HA26 | V-10 | HD23 | P-1 | RESET# | T-15 | AD23 | B-5 | REQ3# | D-13 | | |
| HA27 | Y-9 | HD24 | R-4 | SMIACT# | T-10 | AD24 | D-6 | | | | |
| HA28 | Y-10 | HD25 | P-3 | W/R# | T-9 | AD25 | C-4 | SERR# | E-13 | | |
| HA29 | U-11 | HD26 | P-2 | | | AD26 | B-4 | STOP# | E-11 | | |
| HA30 | W-11 | HD27 | N-1 | | | AD27 | A-4 | TRDY# | E-8 | | |
| HA31 | W-8 | HD28 | P-4 | | | AD28 | B-3 | | | | |
| | | HD29 | N-3 | | | AD29 | A-3 | | | | |
| | | HD30 | N-2 | | | AD30 | B-2 | | | | |
| | | HD31 | N-4 | | | AD31 | A-2 | | | | |
| | | HD32 | M-3 | | | | | | | | |
| | | HD33 | M-2 | | | | | | | | |
| | | HD34 | K-1 | | | | | | | | |
| | | HD35 | M-4 | | | | | | | | |
| | | HD36 | J-2 | | | | | | | | |
| | | HD37 | J-1 | | | | | | | | |
| | | HD38 | J-4 | | | | | | | | |
| | | HD39 | H-1 | | | | | | | | |
| | | HD40 | H-2 | | | | | | | | |
| | | HD41 | G-2 | | | | | | | | |
| | | HD42 | J-3 | | | | | | | | |
| | | HD43 | G-4 | | | | | | | | |
| | | HD44 | H-4 | | | | | | | | |
| | | HD45 | G-1 | | | | | | | | |
| | | HD46 | H-3 | | | | | | | | |
| | | HD47 | F-2 | | | | | | | | |
| | | HD48 | F-1 | | | | | | | | |
| | | HD49 | G-3 | | | | | | | | |
| | | HD50 | F-4 | | | | | | | | |
| | | HD51 | F-3 | | | | | | | | |
| | | HD52 | E-1 | | | | | | | | |
| | | HD53 | E-4 | | | | | | | | |
| | | HD54 | E-2 | | | | | | | | |
| | | HD55 | D-1 | | | | | | | | |
| | | HD56 | E-3 | | | | | | | | |
| | | HD57 | D-3 | | | | | | | | |
| | | HD58 | D-2 | | | | | | | | |
| | | HD59 | C1 | | | | | | | | |
| | | HD60 | C-2 | | | | | | | | |
| | | HD61 | C-3 | | | | | | | | |
| | | HD62 | B-1 | | | | | | | | |
| | | HD63 | A-1 | | | | | | | | |

## Table 11-1. Functional Grouping (continued)

| DRAM Address | | DRAM Data | | DRAM Control | | Power | | Power | | Ground | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Pin Name | Pin No. | Pin Name | Pin No. | Pin Name | Pin No. | Pin Name | Pin No. | Pin Name | Pin No. | Pin Name | Pin No. |
| MA0 | V-12 | MD0 | W-20 | CAS7# | L16 | $V_{DD}$ | E14 | $V_{DD3}$ | F5 | Ground | E15 |
| MA1 | T-12 | MD1 | T-17 | CAS6# | G16 | | G6 | | F6 | | J9 |
| MA2 | Y-13 | MD2 | T-19 | CAS5# | J16 | | | | F14 | | J10 |
| MA3 | W-13 | MD3 | R-20 | CAS4# | H16 | | | | F15 | | J11 |
| MA4 | V-13 | MD4 | P-19 | CAS3# | L17 | | | | P15 | | J12 |
| MA5 | U-13 | MD5 | N-18 | CAS2# | H17 | | | | R6 | | K9 |
| MA6 | T-13 | MD6 | L-20 | CAS1# | K17 | | | | R7 | | K10 |
| MA7 | W-14 | MD7 | K-19 | CAS0# | J17 | | | | R15 | | K11 |
| MA8 | U-14 | MD8 | F-18 | | | | | | R16 | | K12 |
| MA9 | Y-15 | MD9 | D-20 | RAS5# | N17 | | | | | | L9 |
| MA10 | Y-14 | MD10 | C-19 | RAS4# | M17 | | | | | | L10 |
| MA11 | V-14 | MD11 | C-18 | RAS3# | E16 | | | | | | L11 |
| MA12 | F-7 | MD12 | B-18 | RAS2# | F16 | | | | | | L12 |
| MA13 | N-16 | MD13 | B-16 | RAS1# | F17 | | | | | | M9 |
| | | MD14 | B-15 | RAS0# | G17 | | | | | | M10 |
| | | MD15 | C-13 | | | | | | | | M11 |
| | | MD16 | U-18 | SCASA# | W12 | | | | | | M12 |
| | | MD17 | U-20 | SCASB# | U12 | | | | | | T6 |
| | | MD18 | R-18 | SCASC# | P6 | | | | | | T16 |
| | | MD19 | P-17 | SRASA# | Y20 | | | | | | |
| | | MD20 | N-20 | SRASB# | W17 | | | | | | |
| | | MD21 | M-19 | SRASC# | H6 | | | | | | |
| | | MD22 | L-18 | | | | | | | | |
| | | MD23 | J-20 | WEA# | M16 | | | | | | |
| | | MD24 | E-19 | WEB# | G15 | | | | | | |
| | | MD25 | E-17 | WEC# | T11 | | | | | | |
| | | MD26 | D-18 | | | | | | | | |
| | | MD27 | A-19 | MPD7 | G18 | | | | | | |
| | | MD28 | C-17 | MPD6 | H20 | | | | | | |
| | | MD29 | A-17 | MPD5 | G20 | | | | | | |
| | | MD30 | D-16 | MPD4 | H18 | | | | | | |
| | | MD31 | B-14 | MPD3 | F20 | | | | | | |
| | | MD32 | V-19 | MPD2 | J18 | | | | | | |
| | | MD33 | U-19 | MPD1 | G19 | | | | | | |
| | | MD34 | T-20 | MPD0 | H19 | | | | | | |
| | | MD35 | R-17 | | | | | | | | |
| | | MD36 | P-18 | | | | | | | | |
| | | MD37 | M-20 | | | | | | | | |
| | | MD38 | L-19 | | | | | | | | |
| | | MD39 | K-18 | | | | | | | | |
| | | MD40 | F-19 | | | | | | | | |
| | | MD41 | E-18 | | | | | | | | |
| | | MD42 | C-20 | | | | | | | | |
| | | MD43 | A-20 | | | | | | | | |
| | | MD44 | D-17 | | | | | | | | |
| | | MD45 | C-16 | | | | | | | | |
| | | MD46 | D-15 | | | | | | | | |
| | | MD47 | C-14 | | | | | | | | |
| | | MD48 | V-20 | | | | | | | | |
| | | MD49 | T-18 | | | | | | | | |
| | | MD50 | R-19 | | | | | | | | |
| | | MD51 | P-20 | | | | | | | | |
| | | MD52 | N-19 | | | | | | | | |
| | | MD53 | M-18 | | | | | | | | |
| | | MD54 | K-20 | | | | | | | | |
| | | MD55 | J-19 | | | | | | | | |
| | | MD56 | E-20 | | | | | | | | |
| | | MD57 | D-19 | | | | | | | | |
| | | MD58 | B-20 | | | | | | | | |
| | | MD59 | B-19 | | | | | | | | |
| | | MD60 | A-18 | | | | | | | | |
| | | MD61 | B-17 | | | | | | | | |
| | | MD62 | A-16 | | | | | | | | |
| | | MD63 | C-15 | | | | | | | | |

**Table 11-2. AMD-640 System Controller Pin Diagram (Top View)**

| Key | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **A** | HD63 | AD31 | AD29 | AD27 | CBE3 | AD21 | AD18 | CBE2# | AD14 | AD11 | AD8 | AD6 | AD3 | AD1 | AD0 | MD62 | MD29 | MD60 | MD27 | MD43 |
| **B** | HD62 | AD30 | AD28 | AD26 | AD23 | AD20 | AD17 | CBE1# | AD13 | AD10 | CBE0# | AD5 | AD2 | MD31 | MD14 | MD13 | MD61 | MD12 | MD59 | MD58 |
| **C** | HD59 | HD60 | HD61 | AD25 | AD22 | AD19 | AD16 | AD15 | AD12 | AD9 | AD7 | AD4 | MD15 | MD47 | MD63 | MD45 | MD28 | MD11 | MD10 | MD42 |
| **D** | HD55 | HD58 | HD57 | PREQ# | PGNT# | AD24 | REQ0# | GNT0# | REQ1# | GNT1# | REQ2# | GNT2# | REQ3# | GNT3# | MD46 | MD30 | MD44 | MD26 | MD57 | MD9 |
| **E** | HD52 | HD54 | HD56 | HD53 | LOCK# | FRAME# | IRDY# | TRDY# | DEVSEL# | PCLK | STOP# | PAR | SERR# | VDD5 | GND | RAS3# | MD25 | MD41 | MD24 | MD56 |
| **F** | HD48 | HD47 | HD51 | HD50 | VDD3 | VDD3 | MA12 | | | | | | | VDD3 | VDD3 | RAS2# | RAS1# | MD8 | MD40 | MPD3 |
| **G** | HD45 | HD41 | HD49 | HD43 | HLOCK# | VDD5 | | | | | | | | | WEB# | CAS6# | RAS0# | MPD7 | MPD1 | MPD5 |
| **H** | HD39 | HD40 | HD46 | HD44 | M/IO# | SRASC# | | | | | | | | | | CAS4# | CAS2# | MPD4 | MPD0 | MPD6 |
| **J** | HD37 | HD36 | HD42 | HD38 | CACHE# | | | | GND | GND | GND | GND | | | | CAS5# | CAS0# | MPD2 | MD55 | MD23 |
| **K** | HD34 | BE0# | BE1# | BE2# | KEN# | | | | GND | GND | GND | GND | | | | HCLK | CAS1# | MD39 | MD7 | MD54 |
| **L** | BE3# | BE4# | BE5# | BE6# | AHOLD | | | | GND | GND | GND | GND | | | | CAS7# | CAS3# | MD22 | MD38 | MD6 |
| **M** | BE7 | HD33 | HD32 | HD35 | BRDY# | | | | GND | GND | GND | GND | | | | WEA# | RAS4# | MD53 | MD21 | MD37 |
| **N** | HD27 | HD30 | HD29 | HD31 | NA# | | | | | | | | | | | MA13 | RAS5# | MD5 | MD52 | MD20 |
| **P** | HD23 | HD26 | HD25 | HD28 | BOFF# | SCASC# | | | | | | | | VDD3 | VDD3 | TA0 | MD19 | MD36 | MD4 | MD51 |
| **R** | HD7 | HD21 | HD19 | HD24 | EADS | VDD3 | VDD3 | | | | | | | | VDD3 | VDD3 | MD35 | MD18 | MD50 | MD3 |
| **T** | HD12 | HD22 | HD17 | HD20 | ADS# | GND | D/C# | HITM# | W/R# | SMIACT# | SWEC# | MA1 | MA6 | TA4 | RESET# | GND | MD1 | MD49 | MD2 | MD34 |
| **U** | HD8 | HD18 | HD14 | HD16 | HA20 | HA16 | HA12 | HA5 | HA23 | HA22 | HA29 | SCASB# | MA5 | MA8 | COE# | TA1 | TA2 | MD16 | NMD33 | MD17 |
| **V** | HD6 | HD15 | HD10 | HD13 | HA19 | HA14 | HA9 | HA8 | HA21 | HA26 | HA3 | MA0 | MA4 | MA11 | CADV# | TAGWE# | TA3 | TA5 | MD32 | MD48 |
| **W** | HD4 | HD5 | HD9 | HD11 | HA18 | HA15 | HA11 | HA31 | HA25 | HA24 | HA30 | SCASA# | MA3 | MA7 | CADS# | CE1# | SRASB# | TA6 | TA8 | MD0 |
| **Y** | HD0 | HD2 | HD1 | HD3 | HA17 | HA13 | HA10 | HA7 | HA27 | HA28 | HA4 | HA6 | MA2 | MA10 | MA9 | GWE# | BWE# | TA7 | TA9 | SRASA# |

# 12 Package Specifications

The AMD-640 system controller comes in a 328-pin plastic ball grid array (PBGA). The dimensions and thermal specification are shown below.

$\theta_{JA}$ <= 25 °C/W

$\theta_{JC}$ = Not available. Replaced by $\Psi_{J\text{-}T,}$ also not available.

**Table 12-1. 328-Pin BGA Package Preliminary Specification**

| Symbol | Millimeters | | Inches | | Notes |
|---|---|---|---|---|---|
| | Min | Max | Min | Max | |
| A | 26.80 | 27.20 | 1.055 | 1.071 | |
| B | 24.03 | 24.23 | 0.946 | 0.954 | |
| C | 24 | 24 | 0.945 | 0.945 | REF |
| D | 24 | 24 | 0.945 | 0.945 | REF |
| E | 2.20 | 2.46 | 0.087 | 0.097 | |
| F | 1.17 | 1.17 | 0.046 | 0.046 | REF |
| G | 0.56 | 0.56 | 0.022 | 0.022 | REF |
| H | 0.50 | 0.70 | 0.020 | 0.028 | |
| M | 1.27 | 1.27 | 0.050 | 0.050 | NOM |

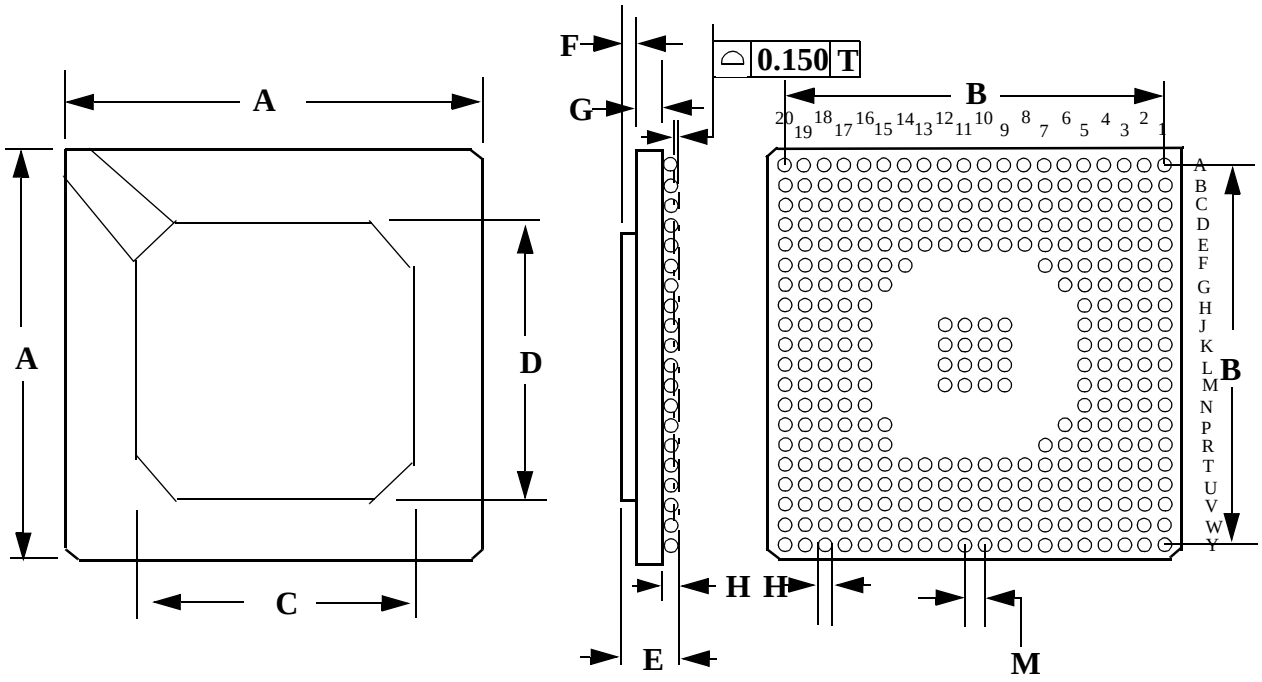



**Figure 12-1. 328-Pin BGA Package Preliminary Specification**