82434LX/82434NX PCI, CACHE AND MEMORY CONTROLLER (PCMC)

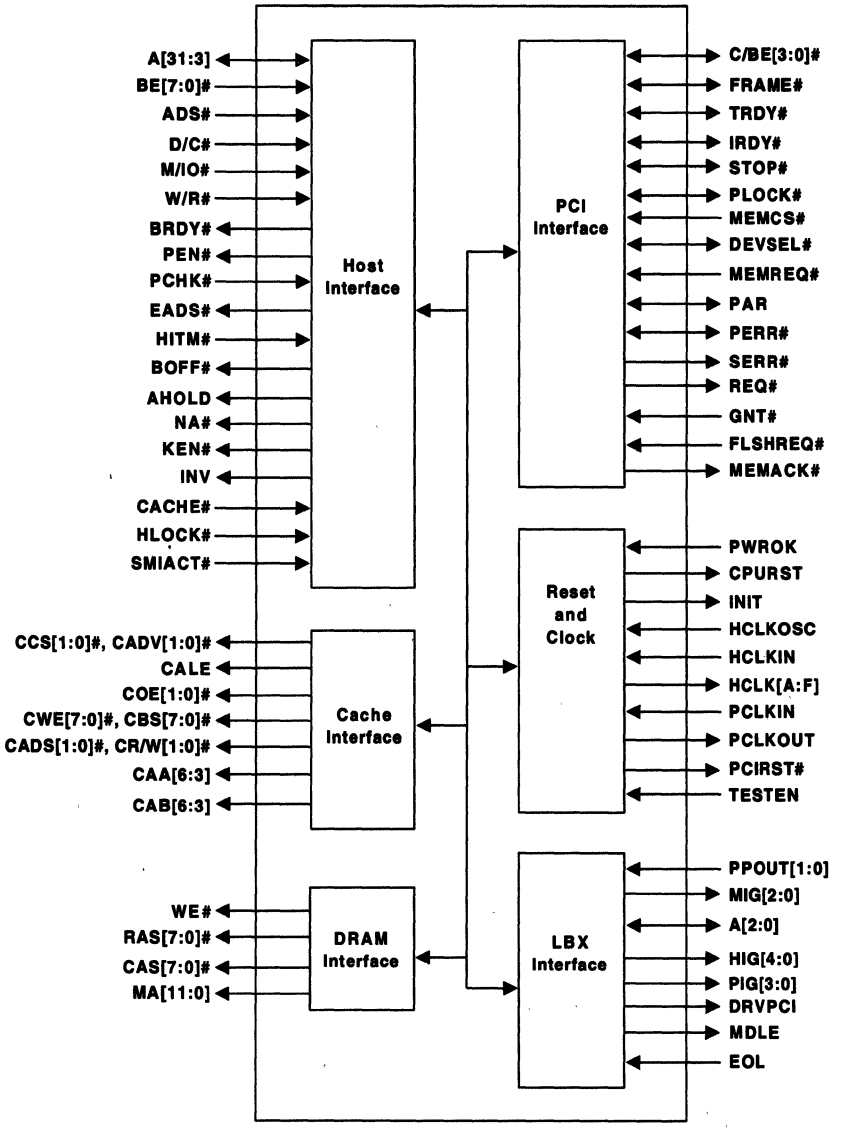
- Supports the Pentium™ Processor at ICOMP™ Index 510\60 MHz and ICOMP Index 567\66 MHz
- Supports the Pentium Processor at ICOMP Index 735\90 MHz, ICOMP Index 815\100 MHz, and ICOMP Index 610\75 MHz
- Supports Pipelined Addressing Capability of the Pentium Processor
- The 82430NX Drives 3.3V Signal Levels on the CPU and Cache Interfaces
- High Performance CPU/PCI/Memory Interfaces via Posted Write and Read Prefetch Buffers
- Fully Synchronous PCI Interface with Full Bus Master Capability
- Supports the Pentium Processor Internal Cache in Either Write-Through or Write-Back Mode
- Programmable Attribute Map of DOS and BIOS Regions for System Flexibility
- Integrated Low Skew Clock Driver for Distributing Host Clock
- Integrated Second Level Cache Controller
  - Integrated Cache Tag RAM
  - Write-Through and Write-Back Cache Modes for the 82434LX
  - Write-Back for the 82434NX
  - 82434NX Supports Low-Power Cache Standby
  - Direct Mapped Organization
  - Supports Standard and Burst SRAMs
  - 256-KByte and 512-KByte Sizes
  - Cache Hit Cycle of 3-1-1-1 on Reads and Writes Using Burst SRAMs
  - Cache Hit Cycle of 3-2-2-2 on Reads and 4-2-2-2 on Writes Using Standard SRAMs
- Integrated DRAM Controller
  - Supports 2 MBytes to 192 MBytes of Cacheable Main Memory for the 82434LX
  - Supports 2 MBytes to 512 MBytes of Cacheable Main Memory for the 82434NX
  - Supports DRAM Access Times of 70 ns and 60 ns
  - CPU Writes Posted to DRAM 4-1-1-1
  - Refresh Cycles Decoupled from ISA Refresh to Reduce the DRAM Access Latency
  - Six RAS# Lines (82434LX)
  - Eight RAS# Lines (82434NX)
  - Refresh by RAS#-Only, or CAS-Before-RAS#, in Single or Burst of Four
- Host/PCI Bridge
  - Translates CPU Cycles into PCI Bus Cycles
  - Translates Back-to-Back Sequential CPU Memory Writes into PCI Burst Cycles
  - Burst Mode Writes to PCI in Zero PCI Wait-States (i.e. Data Transfer Every Cycle)
  - Full Concurrency Between CPU-to-Main Memory and PCI-to-PCI Transactions
  - Full Concurrency Between CPU-to-Second Level Cache and PCI-to-Main Memory Transactions
  - Same Cache and Memory System Logic Design for ISA and EISA Systems
  - Cache Snoop Filter Ensures Data Consistency for PCI-to-Main Memory Transactions
- 208-Pin QFP Package

*Other brands and names are the property of their respective owners.

December 1994
Order Number: 290479-004

This document describes both the 82434LX and 82434NX. Unshaded areas describe the 82434LX. Shaded areas, like this one, describe 82434NX operations that differ from the 82434LX.

The 82434LX/82434NX PCI, Cache, Memory Controllers (PCMC) integrate the cache and main memory DRAM control functions and provide bus control for transfers between the CPU, cache, main memory, and the PCI Local Bus. The cache controller supports write-back (or write-through for 82434LX) cache policy and cache sizes of 256-KBytes and 512-KBytes. The cache memory can be implemented with either standard or burst SRAMs. The PCMC cache controller integrates a high-performance Tag RAM to reduce system cost.



NOTE:
RAS[7:6]# and MA11 are only on the 82434NX. CCS[1:0] functionality is only on the 82434NX.

Simplified Block Diagram of the PCMC

82434LX/82434NX PCI, CACHE AND MEMORY CONTROLLER (PCMC)

CONTENTS

PAGE

1.0 ARCHITECTURAL OVERVIEW 2-222
1.1 System Overview 2-222
1.1.1 BUS HIERARCHY—CONCURRENT OPERATIONS 2-222
1.1.2 BUS BRIDGES 2-225
1.2 PCMC Overview 2-225
1.2.1 CACHE OPERATIONS 2-226
1.2.1.1 Cache Consistency 2-227
1.2.2 ADDRESS/DATA PATHS 2-227
1.2.2.1 Read/Write Buffers 2-227
1.2.3 HOST/PCI BRIDGE OPERATIONS 2-227
1.2.4 DRAM MEMORY OPERATIONS 2-228
1.2.5 3.3V SIGNALS 2-228

2.0 SIGNAL DESCRIPTIONS 2-228
2.1 Host Interface 2-229
2.2 DRAM Interface 2-234
2.3 Cache Interface 2-235
2.4 PCI Interface 2-236
2.5 LBX Interface 2-240
2.6 Reset And Clock 2-240

3.0 REGISTER DESCRIPTION 2-242
3.1 I/O Mapped Registers 2-243
3.1.1 CONFADD—CONFIGURATION ADDRESS REGISTER 2-243
3.1.2 CSE—CONFIGURATION SPACE ENABLE REGISTER 2-244
3.1.3 TRC—TURBO-RESET CONTROL REGISTER 2-245
3.1.4 FORW—FORWARD REGISTER 2-246
3.1.5 PMC—PCI MECHANISM CONTROL REGISTER 2-246
3.1.6 CONFDATA—CONFIGURATION DATA REGISTER 2-246
3.2 PCI Configuration Space Mapped Registers 2-247
3.2.1 CONFIGURATION SPACE ACCESS MECHANISM 2-248
3.2.1.1 Access Mechanism #1: 2-248
3.2.1.2 Access Mechanism #2 2-249
3.2.2 VID—VENDOR IDENTIFICATION REGISTER 2-252
3.2.3 DID—DEVICE IDENTIFICATION REGISTER 2-252

CONTENTS

PAGE

3.2.4 PCICMD—PCI COMMAND REGISTER 2-253
3.2.5 PCISTS—PCI STATUS REGISTER 2-254
3.2.6 RID—REVISION IDENTIFICATION REGISTER 2-255
3.2.7 RLPI—REGISTER-LEVEL PROGRAMMING INTERFACE REGISTER 2-255
3.2.8 SUBC—SUB-CLASS CODE REGISTER 2-255
3.2.9 BASEC—BASE CLASS CODE REGISTER 2-256
3.2.10 MLT—MASTER LATENCY TIMER REGISTER 2-256
3.2.11 BIST—BIST REGISTER 2-256
3.2.12 HCS—HOST CPU SELECTION REGISTER 2-257
3.2.13 DFC—DETURBO FREQUENCY CONTROL REGISTER 2-258
3.2.14 SCC—SECONDARY CACHE CONTROL REGISTER 2-258
3.2.15 HBC—HOST READ/WRITE BUFFER CONTROL 2-260
3.2.16 PBC—PCI READ/WRITE BUFFER CONTROL REGISTER 2-261
3.2.17 DRAMC—DRAM CONTROL REGISTER 2-262
3.2.18 DRAMT—DRAM TIMING REGISTER 2-263
3.2.19 PAM—PROGRAMMABLE ATTRIBUTE MAP REGISTERS (PAM[6:0]) 2-263
3.2.20 DRB—DRAM ROW BOUNDARY REGISTERS 2-266
3.2.20.1 82434LX Description 2-266
3.2.20.2 82434NX Description 2-268
3.2.21 DRBE—DRAM ROW BOUNDARY EXTENSION REGISTER 2-270
3.2.22 ERRCMD—ERROR COMMAND REGISTER 2-270
3.2.23 ERRSTS—ERROR STATUS REGISTER 2-272
3.2.24 SMRS—SMRAM SPACE REGISTER 2-273
3.2.25 MSG—MEMORY SPACE GAP REGISTER 2-273
3.2.26 FBR—FRAME BUFFER RANGE REGISTER 2-274

4.0 PCMC ADDRESS MAP 2-276
4.1 CPU Memory Address Map 2-276
4.2 System Management RAM—SMRAM 2-276
4.3 PC Compatibility Range 2-277
4.4 I/O Address Map 2-278

2

CONTENTS

PAGE

5.0 SECOND LEVEL CACHE INTERFACE 2-279
5.1 82434LX Cache 2-279
5.1.1 CLOCK LATENCIES (82434LX) 2-287
5.1.2 STANDARD SRAM CACHE CYCLES (82434LX) 2-288
5.1.2.1 Burst Read (82434LX) 2-288
5.1.2.2 Burst Write (82434LX) 2-290
5.1.2.3 Cache Line Fill (82434LX) 2-292
5.1.3 BURST SRAM CACHE CYCLES (82434LX) 2-296
5.1.3.1 Burst Read (82434LX) 2-296
5.1.3.2 Burst Write (82434LX) 2-298
5.1.3.3 Cache Line Fill (82434LX) 2-300
5.1.4 SNOOP CYCLES 2-302
5.1.5 FLUSH, FLUSH ACKNOWLEDGE AND WRITE-BACK SPECIAL CYCLES 2-310
5.2 82434NX Cache 2-310
5.2.1 CYCLE LATENCY SUMMARY (82434NX) 2-314
5.2.2 STANDARD SRAM CACHE CYCLES (82434NX) 2-315
5.2.3 SECOND LEVEL CACHE STANDBY 2-315
5.2.4 SNOOP CYCLES 2-315
5.2.5 FLUSH, FLUSH ACKNOWLEDGE, AND WRITE-BACK SPECIAL CYCLES 2-315

6.0 DRAM INTERFACE 2-316
6.1 82434LX DRAM Interface 2-316
6.1.1 DRAM CONFIGURATIONS 2-317
6.1.2 DRAM ADDRESS TRANSLATION 2-317
6.1.3 CYCLE TIMING SUMMARY 2-320
6.1.4 CPU TO DRAM BUS CYCLES 2-320
6.1.4.1 Read Page Hit 2-320
6.1.4.2 Read Page Miss 2-322
6.1.4.3 Read Row Miss 2-323
6.1.4.4 Write Page Hit 2-324
6.1.4.5 Write Page Miss 2-325
6.1.4.6 Write Row Miss 2-326
6.1.4.7 Read Cycle, 0-Active RAS# Mode 2-327
6.1.4.8 Write Cycle, 0-Active RAS# Mode 2-328

CONTENTS

PAGE

6.1.5 REFRESH 2-329
6.1.5.1 RAS#-Only Refresh-Single 2-329
6.1.5.2 CAS#-Before-RAS# Refresh-Single 2-331
6.1.5.3 Hidden Refresh-Single 2-332
6.2 82434NX DRAM Interface 2-333
6.2.1 DRAM ADDRESS TRANSLATION 2-333
6.2.2 CYCLE TIMING SUMMARY 2-334
6.2.3 CPU TO DRAM BUS CYCLES 2-334
6.2.3.1 Burst DRAM Read Page Hit 2-335
6.2.3.2 Burst DRAM Read Page Miss 2-336
6.2.3.3 Burst DRAM Read Row Miss 2-337
6.2.3.4 Burst DRAM Write Page Hit 2-338
6.2.3.5 Burst DRAM Write Page Miss 2-339
6.2.3.6 Burst DRAM Write Row Miss 2-340
6.2.4 REFRESH 2-341
6.2.4.1 RAS#-Only Refresh—Single 2-341
6.2.4.2 CAS#-before-RAS# Refresh—Single 2-342
6.2.4.3 Hidden Refresh-Single 2-343

7.0 PCI INTERFACE 2-344
7.1 PCI Interface Overview 2-344
7.2 CPU-to-PCI Cycles 2-344
7.2.1 CPU WRITE TO PCI 2-344
7.3 Register Access Cycles 2-345
7.3.1 CPU WRITE CYCLE TO PCMC INTERNAL REGISTER 2-346
7.3.2 CPU READ FROM PCMC INTERNAL REGISTER 2-347
7.3.3 CPU WRITE TO PCI DEVICE CONFIGURATION REGISTER 2-348
7.3.4 CPU READ FROM PCI DEVICE CONFIGURATION REGISTER 2-350
7.4 PCI-to-Main Memory Cycles 2-353
7.4.1 PCI MASTER WRITE TO MAIN MEMORY 2-353
7.4.2 PCI MASTER READ FROM MAIN MEMORY 2-355

CONTENTS

PAGE

8.0 SYSTEM CLOCKING AND RESET 2-356
8.1 Clock Domains 2-356
8.2 Clock Generation and Distribution 2-356
8.3 Phase Locked Loop Circuitry 2-357
8.4 System Reset 2-359
8.5 82434NX Reset Sequencing 2-361

9.0 ELECTRICAL CHARACTERISTICS 2-362
9.1 Absolute Maximum Ratings 2-362
9.2 Thermal Characteristics 2-362
9.3 82434LX DC Characteristics 2-362
9.4 82434NX DC Characteristics 2-364
9.5 82434LX AC Characteristics 2-366
9.5.1 HOST CLOCK TIMING, 66 MHz (82434LX) 2-366
9.5.2 CPU INTERFACE TIMING, 66 MHz (82434LX) 2-367
9.5.3 SECOND LEVEL CACHE STANDARD SRAM TIMING, 66 MHz (82434LX) 2-369
9.5.4 SECOND LEVEL CACHE BURST SRAM TIMING, 66 MHz (82434LX) 2-370
9.5.5 DRAM INTERFACE TIMING, 66 MHz (82434LX) 2-370
9.5.6 PCI CLOCK TIMING, 66 MHz (82434LX) 2-370
9.5.7 PCI INTERFACE TIMING, 66 MHz (82434LX) 2-371
9.5.8 LBX INTERFACE TIMING, 66 MHz (82434LX) 2-372
9.5.9 HOST CLOCK TIMING, 60 MHz (82434LX) 2-372
9.5.10 CPU INTERFACE TIMING, 60 MHz (82434LX) 2-373
9.5.11 SECOND LEVEL CACHE STANDARD SRAM TIMING, 60 MHz (82434LX) 2-375
9.5.12 SECOND LEVEL CACHE BURST SRAM TIMING, 60 MHz (82434LX) 2-376
9.5.13 DRAM INTERFACE TIMING, 60 MHz (82434LX) 2-376
9.5.14 PCI CLOCK TIMING, 60 MHz (82434LX) 2-377
9.5.15 PCI INTERFACE TIMING, 60 MHz (82434LX) 2-377
9.5.16 LBX INTERFACE TIMING, 60 MHz (82434LX) 2-378

CONTENTS

PAGE

9.6 82434NX AC Characteristics 2-379
9.6.1 HOST CLOCK TIMING, 66 MHz (82434NX), PRELIMINARY 2-379
9.6.2 CPU INTERFACE TIMING, 66 MHz (82434NX), PRELIMINARY 2-380
9.6.3 SECOND LEVEL CACHE STANDARD SRAM TIMING, 66 MHz (82434NX), PRELIMINARY 2-382
9.6.4 SECOND LEVEL CACHE BURST SRAM TIMING, 66 MHz (82434NX), PRELIMINARY 2-383
9.6.5 DRAM INTERFACE TIMING, 66 MHz (82434NX), PRELIMINARY 2-383
9.6.6 PCI CLOCK TIMING, 66 MHz (82434NX), PRELIMINARY 2-384
9.6.7 PCI INTERFACE TIMING, 66 MHz (82434NX), PRELIMINARY 2-384
9.6.8 LBX INTERFACE TIMING, 66 MHz (82434NX), PRELIMINARY 2-385
9.6.9 HOST CLOCK TIMING, 50 and 60 MHz (82434NX) 2-385
9.6.10 CPU INTERFACE TIMING, 50 AND 60 MHz (82434NX) 2-386
9.6.11 SECOND LEVEL CACHE STANDARD SRAM TIMING, 50 AND 60 MHz (82434NX) 2-388
9.6.12 SECOND LEVEL CACHE BURST SRAM TIMING, 50 AND 60 MHz (82434NX) 2-389
9.6.13 DRAM INTERFACE TIMING, 50 AND 60 MHz (82434NX) 2-389
9.6.14 PCI CLOCK TIMING, 50 AND 60 MHz (82434NX) 2-390
9.6.15 PCI INTERFACE TIMING, 50 AND 60 MHz (82434NX) 2-390
9.6.16 LBX INTERFACE TIMING, 50 AND 60 MHz (82434NX) 2-391
9.6.17 TIMING DIAGRAMS 2-391

10.0 PINOUT AND PACKAGE INFORMATION 2-394
10.1 Pin Assignment 2-394
10.2 Package Characteristics 2-401

11.0 TESTABILITY 2-402

1.0 ARCHITECTURAL OVERVIEW

This section provides an 82430LX/82430NX PCIset system overview that includes a description of the bus hierarchy and bridges between the buses. The 82430LX PCIset consists of the 82434LX PCMC and 82433LX LBX components plus either a PCI/ISA bridge or a PCI/EISA bridge. The 82430NX PCIset consists of the 82434NX PCMC and 82433NX LBX components plus either a PCI/ISA bridge or a PCI/EISA bridge. The PCMC and LBX provide the core cache and main memory architecture and serve as the Host/PCI bridge. An overview of the PCMC follows the system overview section.

1.1 System Overview

The 82430LX/82430NX PCIset provides the Host/PCI bridge, cache and main memory controller, and an I/O subsystem core (either PCI/EISA or PCI/ISA bridge) for the next generation of high-performance personal computers based on the Pentium processor. System designers can take advantage of the power of the PCI (Peripheral Component Interconnect) local bus while maintaining access to the large base of EISA and ISA expansion cards. Extensive buffering and buffer management within the bridges ensures maximum efficiency in all three buses (Host CPU, PCI, and EISA/ISA Buses).

For an ISA-based system, the PCIset includes the System I/O (82378IB SIO) component (Figure 1) as the PCI/ISA bridge. For an EISA-based system (Figure 2), the PCIset includes the PCI-EISA bridge (82375EB PCEB) and the EISA System Component (82374EB ESC). The PCEB and ESC work in tandem to form the complete PCI/EISA bridge.

1.1.1. BUS HIERARCHY—CONCURRENT OPERATIONS

Systems based on the 82430LX/82430NX PCIset contain three levels of buses structured in the following hierarchy:

- Host Bus as the execution bus
- PCI Bus as a primary I/O bus
- ISA or EISA Bus as a secondary I/O bus.

This bus hierarchy allows concurrency for simultaneous operations on all three buses. Data buffering permits concurrency for operations that crossover into another bus. For example, the Pentium processor could post data destined to the PCI in the LBX. This permits the Host transaction to complete in minimum time, freeing up the Host Bus for further transactions. The Pentium processor does not have to wait for the transfer to complete to its final destination. Meanwhile, any ongoing PCI Bus transactions are permitted to complete. The posted data is then transferred to the PCI Bus when the PCI Bus is available. The LBX implements extensive buffering for Host-to-PCI, Host-to-main memory, and PCI-to-main memory transactions. In addition, the PCEB/ESC chip set and the SIO implement extensive buffering for transfers between the PCI Bus and the EISA and ISA Buses, respectively.

Host Bus

Designed to meet the needs of high-performance computing, the Host Bus features:

- 64-bit data path
- 32-bit address bus with address pipelining
- Synchronous frequencies of 60 MHz and 66 MHz
- Synchronous frequency of 50 MHz (82430NX)
- Burst read and write transfers
- Support for first level and second level caches
- Capable of full concurrency with the PCI and memory subsystems
- Byte data parity
- Full support for Pentium processor machine check and DOS compatible parity reporting
- Support for Pentium processor System Management Mode (SMM).

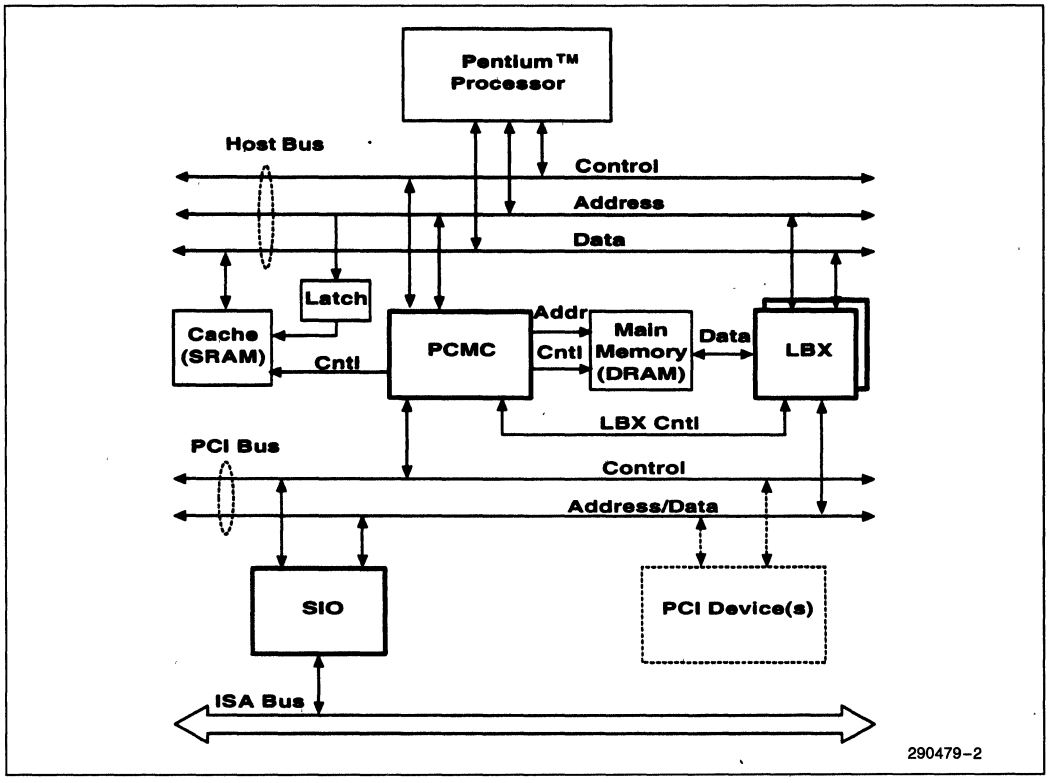


Figure 1. Block Diagram of a 82430LX/82430NX PCIset ISA System

PCI Bus

The PCI Bus is designed to address the growing industry needs for a standardized *local bus* that is not directly dependent on the speed and the size of the processor bus. New generations of personal computer system software such as Windows™ and Win-NT™ with sophisticated graphical interfaces, multi-tasking, and multi-threading bring new requirements that traditional PC I/O architectures cannot satisfy. In addition to the higher bandwidth, reliability and robustness of the I/O subsystem are becoming increasingly important. PCI addresses these needs and provides a future upgrade path. PCI features include:

- Processor independent
- Multiplexed, burst mode operation
- Synchronous at frequencies up to 33 MHz
- 120 MByte/sec usable throughput (132 MByte/sec peak) for a 32-bit data path

- Low latency random access (60 ns write access latency to slave registers from a master parked on the bus)
- Capable of full concurrency with the processor/memory subsystem
- Full multi-master capability allowing any PCI master peer-to-peer access to any PCI slave
- Hidden (overlapped) central arbitration
- Low pin count for cost effective component packaging (multiplexed address/data)
- Address and data parity
- Three physical address spaces: memory, I/O, and configuration
- Comprehensive support for autoconfiguration through a defined set of standard configuration functions.

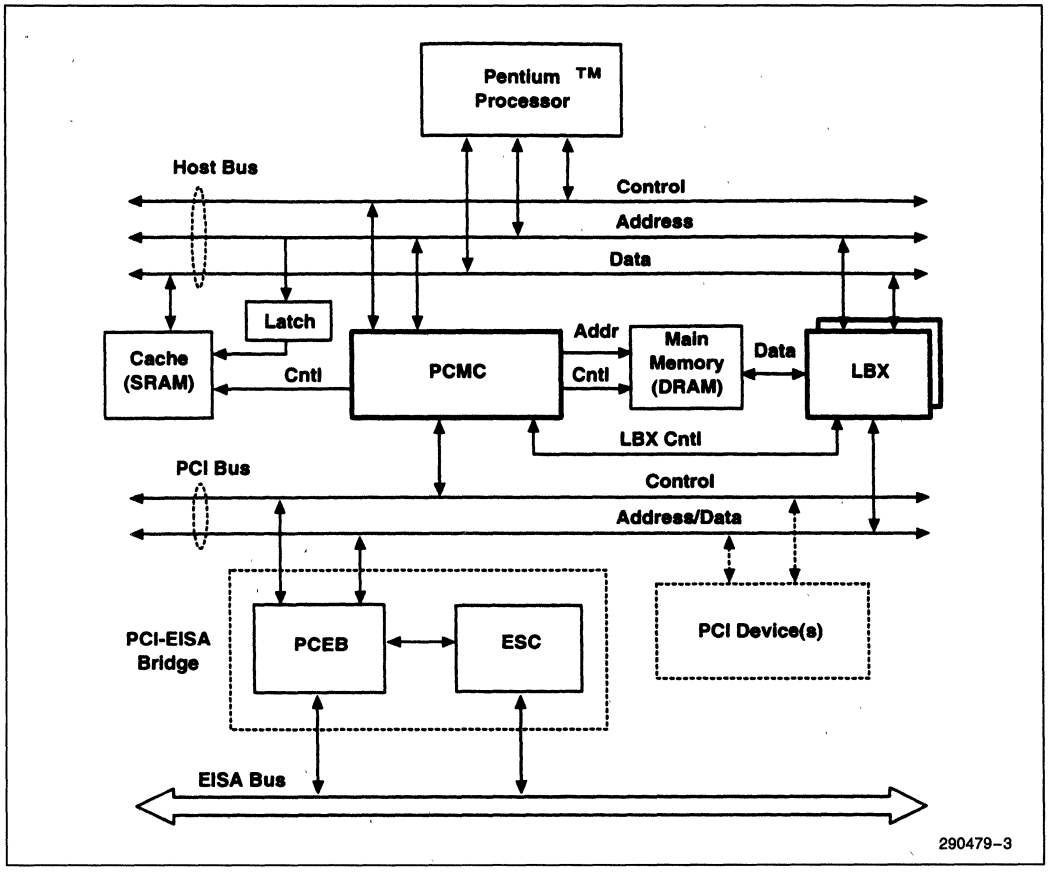


Figure 2. Block Diagram of the 82430LX/82430NX PCIset EISA System

ISA Bus

Figure 1 represents a system using the ISA Bus as the second level I/O bus. It allows personal computer platforms built around the PCI as a primary I/O bus to leverage the large ISA product base. The ISA Bus has 24-bit addressing and a 16-bit data path.

EISA Bus

Figure 2 represents a system using the EISA Bus as the second level I/O bus. It allows personal computer platforms built around the PCI as a primary I/O bus to leverage the large EISA/ISA product base. Combinations of PCI and EISA buses, both of which can be used to provide expansion functions, will satisfy even the most demanding applications.

Along with compatibility for 16-bit and 8-bit ISA hardware and software, the EISA bus provides the following key features:

- 32-bit addressing and 32-bit data path
- 33 MByte/sec bus bandwidth
- Multiple bus master support through efficient arbitration
- Support for autoconfiguration.

1.1.2 BUS BRIDGES

Host/PCI Bridge Chip Set (PCMC and LBX)

The PCMC and LBX enhance the system performance by allowing for concurrency between the Host CPU Bus and PCI Bus, giving each greater bus throughput and decreased bus latency. The LBX contains posted write buffers for Host-to-PCI, Host-to-main memory, and PCI-to-main memory transfers. The LBX also contains read prefetch buffers for Host reads of PCI, and PCI reads of main memory. There are two LBXs per system. The LBXs are controlled by commands from the PCMC. The PCMC/LBX Host/PCI bridge chip set is covered in more detail in Section 1.2, PCMC Overview.

PCI-EISA Bridge Chip Set (PCEB and ESC)

The PCEB provides the master/slave functions on both the PCI Bus and the EISA Bus. Functioning as a bridge between the PCI and EISA buses, the PCEB provides the address and data paths, bus controls, and bus protocol translation for PCI-to-EISA and EISA-to-PCI transfers. Extensive data buffering in both directions increase system performance by maximizing PCI and EISA Bus efficiency and allowing concurrency on the two buses. The PCEB's buffer management mechanism ensures data coherency. The PCEB integrates central bus control functions including a programmable bus arbiter for the PCI Bus and EISA data swap buffers for the EISA Bus. Integrated system functions include PCI parity generation, system error reporting, and programmable PCI and EISA memory and I/O address space mapping and decoding. The PCEB also contains a BIOS Timer that can be used to implement timing loops. The PCEB is intended to be used with the ESC to provide an EISA I/O subsystem interface.

The ESC integrates the common I/O functions found in today's EISA-based PCs. The ESC incorporates the logic for EISA Bus controller, enhanced seven channel DMA controller with scatter-gather support, EISA arbitration, 14 level interrupt controller, Advanced Programmable Interrupt Controller (APIC), five programmable timer/counters, non-maskable-interrupt (NMI) control, and power management. The ESC also integrates support logic to decode peripheral devices (e.g., the flash BIOS, real time clock, keyboard/mouse controller, floppy controller, two serial ports, one parallel port, and IDE hard disk drive).

PCI/ISA Bridge (SIO):

The SIO component provides the bridge between the PCI Bus and the ISA Bus. The SIO also integrates many of the common I/O functions found in today's ISA-based PCs. The SIO incorporates the logic for a PCI interface (master and slave), ISA interface (master and slave), enhanced seven channel DMA controller that supports fast DMA transfers and scatter-gather, data buffers to isolate the PCI Bus from the ISA Bus and to enhance performance, PCI and ISA arbitration, 14 level interrupt controller, a 16-bit BIOS timer, three programmable timer/counters, and non-maskable-interrupt (NMI) control logic. The SIO also provides decode for peripheral devices (e.g., the flash BIOS, real time clock, keyboard/mouse controller, floppy controller, two serial ports, one parallel port, and IDE hard disk drive).

1.2 PCMC Overview

The PCMC (along with the LBX) provides three basic functions: a cache controller, a main memory DRAM controller, and a Host/PCI bridge. This section provides an overview of these functions. Note that, in this document, operational descriptions assume that the PCMC and LBX components are used together.

1.2.1 CACHE OPERATIONS

The PCMC provides the control for a second level cache memory array implemented with either standard asynchronous SRAMs or synchronous burst SRAMs. The data memory array is external to the PCMC and located on the Host address/data bus. Since the Pentium processor contains an internal cache, there can be two separate caches in a Host subsystem. The cache inside the Pentium processor is referred to as the first level cache (also called primary cache). A detailed description of the first level cache is beyond the scope of this document. The PCMC cache control circuitry and associated external memory array is referred to as the second level cache (also called secondary cache). The second level cache is unified, meaning that both CPU data and instructions are stored in the cache. The 82434LX PCMC supports both write-through and write-back caching policies and the 82434NX supports write-back.

The optional second level cache memory array can be either 256-KBytes or 512-KBytes in size. The cache is direct-mapped and is organized as either 8K or 16K cache lines of 32 bytes per line.

In addition to the cache data RAM, the second level cache contains a 4K set of cache tags that are internal to the PCMC. Each tag contains an address that is associated with the corresponding data sector (2 lines for a 256 KByte cache and 4 lines for a 512 KByte cache) and two status bits for each line in the sector.

During a main memory read or write operation, the PCMC first searches the cache. If the addressed code or data is in the cache, the cycle is serviced by the cache. If the addressed code or data is not in the cache, the cycle is forwarded to main memory.

For the write-through (82434LX only) and write-back (both 82434LX and 82434NX) policies, the cache operation is determined by the CPU read or write cycle as follows:

Write Cycle

If the caching policy is write-through and the write cycle hits in the cache, both the cache and main memory are updated. Upon a cache miss, only main memory is updated. The cache is not updated (no write-allocate).

If the caching policy is write-back and the write cycle hits in the cache, only the cache is updated; main memory is not affected. Upon a cache miss, only main memory is updated. The cache is not updated (no write-allocate).

Read Cycle

Upon a cache hit, the cache operation is the same for both write-through and write-back. In this case, data is transferred from the cache to the CPU. Main memory is not accessed.

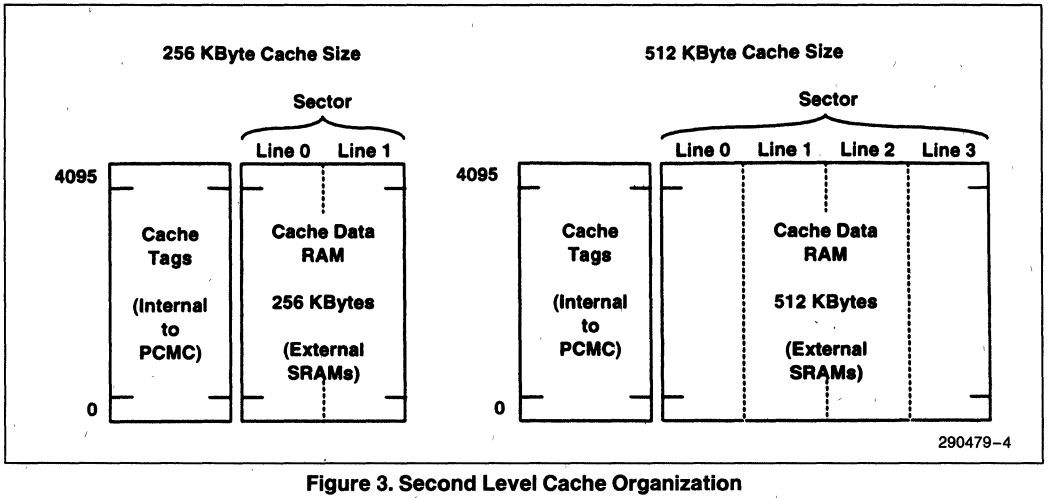


Figure 3. Second Level Cache Organization

If the read cycle causes a cache miss, the line containing the requested data is transferred from main memory to the cache and to the CPU. In the case of a write-back cache, if the cache line fill is to a sector containing one or more modified lines, the modified lines are written back to main memory and the new line is brought into the cache. For a modified line write-back operation, the PCMC transfers the modified cache lines to main memory via a write buffer in the LBX. Before writing the last modified line from the write buffer to main memory, the PCMC updates the first and second level caches with the new line, allowing the CPU access to the requested data with minimum latency.

1.2.1.1 Cache Consistency

The Snoop mechanism in the PCMC ensures data consistency between cache (both first level and second level) and main memory. The PCMC monitors PCI master accesses to main memory and when needed, initiates an inquire (snoop) cycle to the first and second level caches. The snoop mechanism guarantees that consistent data is always delivered to both the host CPU and PCI masters.

1.2.2 ADDRESS/DATA PATHS

Address paths between the CPU/cache and PCI and data paths between the CPU/cache, PCI, and main memory are supplied by two LBX components. The LBX is a companion component to the PCMC. Together, they form a Host/PCI bridge. The PCMC (via the PCMC/LBX interface signals), controls the address and data flow through the LBXs. Refer to the LBX data sheet for more details on the address and data paths.

Data is transferred to and from the PCMC internal registers via the PCMC address lines. When the Host CPU performs a write operation, the data is sent to the LBXs. When the PCMC decodes the cycle as an access to one of its internal registers, it asserts AHOLD to the CPU and instructs the LBXs to copy the data onto the Host address lines. When the PCMC decodes a Host read as an access to a PCMC internal register, it asserts AHOLD to the CPU. The PCMC then places the register data on its address lines and instructs the LBX to copy the data on the Host address bus to the Host data bus. When the register data is on the Host data bus, the PCMC negates AHOLD and completes the cycle.

1.2.2.1 Read/Write Buffers

The LBX provides an interface for the CPU address and data buses, PCI Address/Data bus, and the main memory DRAM data bus. There are three posted write buffers and one read-prefetch buffers implemented in the LBXs to increase performance and to maximize concurrency. The buffers are:

- CPU-to-Main Memory Posted Write Buffer (4 Qwords)
- CPU-to-PCI Posted Write Buffer (4 Dwords)
- PCI-to-Main Memory Posted Write Buffer (2 x 4 Dwords)
- PCI-to-Main Memory Read Prefetch Buffer (line buffer, 4 Qwords).

Refer to the LBX data sheet for details on the operation of these buffers.

1.2.3 HOST/PCI BRIDGE OPERATIONS

The PCMC permits the Host CPU to access devices on the PCI Bus. These accesses can be to PCI I/O space, PCI memory space, or PCI configuration space.

As a PCI device, the PCMC can be either a master initiating a PCI Bus operation or a target responding to a PCI Bus operation. The PCMC is a PCI Bus master for Host-to-PCI cycles and a target for PCI-to-main memory transfers. Note that the PCMC does not permit peripherals to be located on the Host Bus. CPU I/O cycles, other than to PCMC internal registers, are forwarded to the PCI Bus and PCI Bus accesses to the Host Bus are not supported.

When the CPU initiates a bus cycle to a PCI device, the PCMC becomes a PCI Bus master and translates the CPU cycle into the appropriate PCI Bus cycle. The Host/PCI Posted write buffer in the LBXs permits the CPU to complete CPU-to-PCI Dword memory writes in three CPU clocks (1 wait-state), even if the PCI Bus is currently busy. The posted data is written to the PCI device when the PCI Bus is available.

When a PCI Bus master initiates a main memory access, the PCMC (and LBXs) become the target of the PCI Bus cycle and responds to the read/write access. During PCI-to-main memory accesses, the PCMC automatically performs cache snoop operations on the Host Bus, when needed, to maintain data consistency.

As a PCI device, the PCMC contains all of the required PCI configuration registers. The Host CPU reads and writes these registers as described in Section 3.0, Register Description.

1.2.4 DRAM MEMORY OPERATIONS

The PCMC contains a DRAM controller that supports CPU and PCI master accesses to main memory. The PCMC DRAM interface supplies the control signals and address lines and the LBXs supply the data path. DRAM parity is generated for main memory writes and checked for memory reads.

For the 82434LX, the memory array is 64-bits wide and ranges in size from 2 MBytes–192 MBytes. The array can be implemented with either single-sided or double-sided SIMMs. DRAM SIMM sizes of 256K x 36, 1M x 36, and 4M x 36 are supported.

For the 82434NX, the memory array is 64-bits wide and ranges in size from 2 MBytes–512 MBytes. The array can be implemented with either single-sided or double-sided SIMMs. DRAM SIMM sizes of 256K x 36, 1M x 36, 4M x 36, and 16M x 36 are supported.

To provide optimum support for the various cache configurations, and the resultant mix of bus cycles, the system designer can select between 0-active RAS# and 1-active RAS# modes. These modes affect the behavior of the RAS# signal following either CPU-to-main memory cycles or PCI-to-main memory cycles.

The PCMC also provides programmable memory and cacheability attributes on 14 memory segments of various sizes in the ISA compatibility range (512 KByte–1 MByte address range). Access rights to these memory segments from the PCI Bus are controlled by the expansion bus bridge.

The PCMC permits a gap to be created in main memory within the 1 MByte–16 MBytes address range, accommodating ISA devices which are mapped into this range (e.g., ISA LAN card or an ISA frame buffer).

1.2.5 3.3V SIGNALS

The 82434NX PCMC drives 3.3V signal levels on the CPU and second level cache interfaces. Thus, no extra logic (i.e. 5V/3.3V translation) is required when interfacing to 3.3V processors and SRAMs. Six of the power pins on the 82434NX are VDD3 pins. These pins are connected to a 3.3V power supply. The VDD3 pins power the output buffers on the CPU and second level cache interfaces. The VDD3 pins also power the output buffers for the HCLK[A-F] outputs.

2.0 SIGNAL DESCRIPTIONS

This section provides a detailed description of each signal. The signals are arranged in functional groups according to their associated interface. The states of all of the signals during hard reset are provided in Section 8.0, System Clocking and Reset.

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

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

The following notations are used to describe the signal type.

in Input is a standard input-only signal

out Totem pole output is a standard active driver

o/d Open drain

t/s Tri-State is a bi-directional, tri-state input/output pin

s/t/s Sustained tri-state is an active low tri-state signal owned and driven by one and only one agent at a time. The agent that drives a s/t/s pin low must drive it high for at least one clock before letting it float. A new agent can not start driving a s/t/s signal any sooner than one clock after the previous owner tri-states it. An external pull-up is required to sustain the inactive state until another agent drives it and must be provided by the central resource.

2.1 Host Interface

Signal	Type	Description
A[31:0]	t/s	**ADDRESS BUS:** A[31:0] are the address lines of the Host Bus. A[31:3] are connected to the CPU A[31:3] lines and to the LBXs. A[2:0] are only connected to the LBXs. Along with the byte enable signals, the A[31:3] lines define the physical area of memory or I/O being accessed. During CPU cycles, the A[31:3] lines are inputs to the PCMC. They are used for address decoding and second level cache tag lookup sequences. Also during CPU cycles, A[2:0] are outputs and are generated from BE[7:0]#. A[27:24] provide hardware strapping options for test features. For more details on theses options, refer to Section 11.0 Testability. During inquire cycles, A[31:5] are inputs from the LBXs to the CPU and the PCMC to snoop the first and the second level cache tags, respectively. In response to a Flush or Flush Acknowledge Special Cycle, the PCMC asserts AHOLD and drives the addresses of the second level cache lines to be written back to main memory on A[18:7]. During CPU to PCI configuration cycles, the PCMC drives A[31:0] with the PCI configuration space address that is internally derived from the CPU physical I/O address. All PCMC internal configuration registers are accessed via A[31:0]. During CPU reads from PCMC internal configuration registers, the PCMC asserts AHOLD and drives the contents of the addressed register on A[31:0]. The PCMC then signals the LBXs to copy this value from the address lines onto the host data lines. During writes to PCMC internal configuration registers, the PCMC asserts AHOLD and signals the LBXs to copy the write data onto the A[31:0] lines. Finally, when in deturbo mode, the PCMC periodically asserts AHOLD and then drives A[31:0] to valid logic levels to keep these lines from floating for an extended period of time. A[31:28] provide hardware strapping options at powerup. For more details on strapping options, refer to Section 8.0, System Clocking and Reset. A[27:24] provide hardware strapping options for test features. For more details on these options, refer to Section 11.0 Testability.

Signal	Type	Description
BE[7:0]#	in	**BYTE ENABLES:** The byte enables indicate which byte lanes on the CPU data bus carry valid data during the current bus cycle. In the case of cacheable reads, all 8 bytes of data are driven to the Pentium processor, regardless of the state of the byte enables. The byte enable signals indicate the type of special cycle when M/IO# = D/C# = 0 and W/R# = 1. During special cycles, only one byte enable is asserted by the CPU. The following table depicts the special cycle types and their byte enable encodings: **Special Cycle Type** — **Asserted Byte Enable** Shutdown — BE0# Flush — BE1# Halt/Stop Grant — BE2# Write Back — BE3# Flush Acknowledge — BE4# Branch Trace Message — BE5# When the PCMC decodes a Shutdown Special Cycle, it asserts AHOLD, drives 000...000 (the PCI Shutdown Special Cycle Encoding) on the A[31:0] lines and signals the LBXs to latch the host address bus. The PCMC then drives a Special Cycle on PCI, signaling the LBXs to drive the latched address (00...00) on the AD[31:0] lines during the data phase. The PCMC then asserts INIT for 16 HCLKs. In response to Flush and Flush Acknowledge Special Cycles, the PCMC internally inspects the Valid and Modified bits for each of the Second Level Cache Sectors. If a line is both valid and modified, the PCMC drives the cache address of the line on the A[18:7] and CAA/CAB[6:3] lines and writes the line back to main memory. The valid and modified bits are both reset to 0. All valid and unmodified lines are simply marked invalid. In response to a write back special cycle, the PCMC simply returns BRDY# to the CPU. The second level cache will be written back to main memory in response to the following flush special cycle. If BE2# is asserted during a special cycle, the 82434NX uses A4 to determine if the cycle is a Halt or Stop Grant Special Cycle. If A4 = 0, the cycle is a Halt Special Cycle and if A4 = 1, the cycle is a Stop Grant Special cycle. In response to a halt special cycle, the PCMC asserts AHOLD, drives 000...001 (the PCI halt special cycle encoding) on the A[31:0] lines, and signals the LBXs to latch the host address bus. The PCMC then drives a special cycle on PCI, signaling the LBXs to drive the latched address (00...01) on the AD[31:0] lines during the data phase. When the 82434NX PCMC detects a CPU Stop Grant Special Cycle (M/IO# = 0, D/C# = 0, W/R# = 1, A4 = 1, BE[7:0]# = FBh), it generates a PCI Stop Grant Special cycle, with 0002h in the message field (AD[15:0]) and 0012h in the message dependent data field (AD[31:16]) during the first data phase (IRDY# asserted).
ADS#	in	**ADDRESS STROBE:** The Pentium processor asserts ADS# to indicate that a new bus cycle is beginning. ADS# is driven active in the same clock as the address, byte enable, and cycle definition signals. The PCMC ignores a floating low ADS# that may occur when BOFF# is asserted as the CPU is asserting ADS#.

Signal	Type	Description
BRDY#	out	**BURST READY:** BRDY# indicates that the system has responded in one of three ways: 1. valid data has been placed on the Pentium processor data pins in response to a read, 2. CPU write data has been accepted by the system, or 3. the system has responded to a special cycle.
NA#	out	**NEXT ADDRESS:** The PCMC asserts NA# for one clock when the memory system is ready to accept a new address from the CPU, even if all data transfers for the current cycle have not completed. The CPU may drive out a pending cycle two clocks after NA# is asserted and has the ability to support up to two outstanding bus cycles.
AHOLD	out	**ADDRESS HOLD:** The PCMC asserts AHOLD to force the Pentium processor to stop driving the address bus so that either the PCMC or LBXs can drive the bus. During PCI master cycles, AHOLD is asserted to allow the LBXs to drive a snoop address onto the address bus. If the PCI master locks main memory, AHOLD remains asserted until the PCI master locked sequence is complete and the PCI master negates PLOCK#. AHOLD is asserted during all accesses to PCMC internal configuration registers to allow configuration register accesses to occur over the A[31:0] lines. When in deturbo mode, the PCMC periodically asserts AHOLD to prevent the processor from initiating bus cycles in order to emulate a slower system. The duration of AHOLD assertion in deturbo mode is controlled by the Deturbo Frequency Control Register (offset 51h). When PWROK is negated, the PCMC asserts AHOLD to allow the strapping options on A[31:28] to be read. For more details on strapping options, see the System Clocking and Reset section.
EADS#	out	**EXTERNAL ADDRESS STROBE:** The PCMC asserts EADS# to indicate to the Pentium processor that a valid snoop address has been driven onto the CPU address lines to perform an inquire cycle. During PCI master cycles, the PCMC signals the LBXs to drive a snoop address onto the host address lines and then asserts EADS# to cause the CPU to sample the snoop address.
INV	out	**INVALIDATE:** The INV signal specifies the final state (invalid or shared) that a first level cache line transitions to in the event of a cache line hit during a snoop cycle. When snooping the caches during a PCI master write, the PCMC asserts INV with EADS#. When INV is asserted with EADS#, an inquire hit results in the line being invalidated. When snooping the caches during a PCI master read, the PCMC does not assert INV with EADS#. In this case, an inquire cycle hit results in a line transitioning to the shared state.
BOFF#	out	**BACKOFF:** The PCMC asserts BOFF# to force the Pentium processor to abort all outstanding bus cycles that have not been completed and float its bus in the next clock. The PCMC uses this signal to force the CPU to re-order a write-back due to a snoop cycle around a currently outstanding bus cycle. The PCMC also asserts BOFF# to obtain the CPU data bus for write-back cycles from the secondary cache due to a snoop hit. The CPU remains in bus hold until BOFF# is negated.
HITM#	in	**HIT MODIFIED:** The Pentium processor asserts HITM# to inform the PCMC that the current inquire cycle hit a modified line. HITM# is asserted by the Pentium processor two clocks after the assertion of EADS# if the inquire cycle hits a modified line in the primary cache.

Signal	Type	Description
M/IO# D/C# W/R#	in	**BUS CYCLE DEFINITION (MEMORY/INPUT-OUTPUT, DATA/CONTROL, WRITE/READ):** M/IO, D/C# and W/R# define Host Bus cycles as shown in the table below. **M/IO# D/C# W/R# Bus Cycle Type** Low Low Low Interrupt Acknowledge Low Low High Special Cycle Low High Low I/O Read Low High High I/O Write High Low Low Code Read High Low High Reserved High High Low Memory Read High High High Memory Write Interrupt acknowledge cycles are forwarded to the PCI Bus as PCI interrupt acknowledge cycles (i.e. C/BE[3:0]# = 0000 during the address phase). All I/O cycles and any memory cycles that are not directed to memory controlled by the PCMC DRAM controller are forwarded to PCI. The Pentium processor generates six different types of special cycles. The special cycle type is encoded on the BE[7:0]# lines.
HLOCK#	in	**HOST BUS LOCK:** The Pentium processor asserts HLOCK# to indicate the current bus cycle is locked. HLOCK# is asserted in the first clock of the first locked bus cycle and is negated after the BRDY# is returned for the last locked bus cycle. The Pentium processor guarantees HLOCK# to be negated for at least one clock between back-to-back locked operations. When a CPU locked cycle is directed to main memory, the PCMC guarantees that once the locked operation begins in main memory, the CPU has exclusive access to main memory (i.e., PCI master accesses to main memory will not be initiated until the CPU locked operation completes). When a CPU locked cycle is directed to PCI, the PCMC arbitrates for PLOCK# (PCI LOCK#) before initiating the cycle on PCI, except when the cycle is to the memory range defined by the Frame Buffer Range Register and the No Lock Requests bit in that register is set to 1.
CACHE#	in	**CACHEABILITY:** The Pentium processor asserts CACHE# to indicate the internal cacheability of a read cycle or that a write cycle is a burst write-back cycle. If the CPU drives CACHE# inactive during a read cycle, the returned data is not cached, regardless of the state of KEN#. The CPU asserts CACHE# for cacheable data reads, cacheable code fetches, and cache line write-backs. CACHE# is driven along with the cycle definition pins.
KEN#	out	**CACHE ENABLE:** The PCMC asserts KEN# to indicate to the CPU that the current cycle is cacheable. KEN# is asserted for all accesses to memory ranges 0–512-KBytes and 1024-KBytes to the top of main memory controlled by the PCMC when the Primary Cache Enable bit is set to 1, except in the following case: KEN# is not asserted for accesses to the top 64-KByte of main memory controlled by the PCMC when the SMRAM Enable bit in the DRAM Control Register (Offset 57h) is set to 1 and the area is not write protected. If the area is write protected and cacheable, KEN# is asserted for code read cycles, but is not asserted during data read cycle. KEN# is asserted for any CPU access within the range of 512-KBytes–1024-KBytes if the corresponding Cache Enable bit in the PAM[6:0] Registers (offsets 59h–5Fh) is set to 1. When the Pentium processor indicates that the current read cycle can be cached by asserting CACHE# and the PCMC responds with KEN#, the cycle is converted into a burst cache line fill. The CPU samples KEN# with the first of either BRDY# or NA#.

Signal	Type	Description
SMIACT#	in	**SYSTEM MANAGEMENT INTERRUPT ACTIVE:** The Pentium processor asserts SMIACT# to indicate that the processor is operating in System Management Mode (SMM). When the SMRAM Enable bit in the DRAM Control Register (offset 57h) is set to 1, the PCMC allows CPU accesses SMRAM as permitted by the SMRAM Space Register at configuration space offset 72h.
PEN#	out	**PARITY ENABLE:** The PEN# signal, along with the MCE bit in CR4 of the Pentium processor, determines whether a machine check exception will be taken by the CPU as a result of a parity error on a read cycle. The PCMC asserts PEN# during DRAM read cycles if the MCHK on DRAM/L2 Cache Data Parity Error Enable bit in the Error Command Register (offset 70h) is set to 1. The PCMC asserts PEN# during CPU second level cache read cycles if the MCHK on DRAM/L2 Cache Data Parity Error Enable and the L2 Cache Parity Enable bits in the Error Command Register (offset 70h) are both set to 1.
PCHK#	in	**DATA PARITY CHECK:** PCHK# is sampled by the PCMC to detect parity errors on CPU read cycles from main memory if the Parity Error Mask Enable bit in the DRAM Control Register (offset 57h) is reset to 0. PCHK# is sampled by the PCMC to detect parity errors on CPU read cycles from the second level cache if the L2 Cache Parity Enable bit in the Error Command Register (offset 70h) is set to 1. If incorrect parity was detected on a data read, the PCHK# signal is asserted by the Pentium processor two clocks after BRDY# is returned. PCHK# is asserted for one clock for each clock in which a parity error was detected.

2.2 DRAM Interface

Signal	Type	Description
RAS[5:0]#	out	**ROW ADDRESS STROBES:** The RAS[5:0]# signals are used to latch the row address on the MA[10:0] lines into the DRAMs. Each RAS[5:0]# signal corresponds to one DRAM row. The 82434LX PCMC supports up to 6 rows in the DRAM array. Each row is eight bytes wide. These signals drive the RAS# lines of the DRAM array directly, without external buffers.
RAS[7:6]#	out	**ROW ADDRESS STROBES:** The 82434NX supports up to eight rows of DRAM. RAS[7:6]# are used with RAS[5:0] to latch the row address on the MA[11:0] lines into the DRAMs. Each row is eight bytes wide. These signals drive the RAS# lines of the DRAM array directly, without external buffers.
CAS[7:0]#	out	**COLUMN ADDRESS STROBES:** The CAS[7:0]# signals are used to latch the column address on the MA[10:0] lines into the DRAMs. Each CAS[7:0]# signal corresponds to one byte of the eight byte-wide array. These signals drive the CAS# lines of the DRAM array directly, without external buffers. In a minimum configuration, each CAS[7:0]# line only has one SIMM load, while the maximum configuration has 6 SIMM loads.
WE#	out	**DRAM WRITE ENABLE:** WE# is asserted during both CPU and PCI master writes to main memory. During burst writes to main memory, WE# is asserted before the first assertion of CAS[7:0]# and is negated with the last CAS[7:0]#. The WE# signal is externally buffered to drive the WE# inputs on the DRAMs.
MA[10:0]	out	**DRAM MULTIPLEXED ADDRESS:** MA[10:0] provide the row and column address to the DRAM array. The 82434LX uses MA[10:0] for the complete DRAM address bus. The MA[10:0] lines are externally buffered to drive the multiplexed address lines of the DRAM array.
MA11	out	**DRAM MULTIPLEXED ADDRESS:** MA11 provides the extra addressability for the 16M x 36 SIMMs that are supported by the 82434NX. MA[11:0] provide the row and column address to the DRAM array. Like MA[10:0], MA11 is externally buffered to drive the multiplexed address lines of the DRAM array.

2.3 Cache Interface

Signal	Type	Description
CALE	out	**CACHE ADDRESS LATCH ENABLE:** CALE controls the external latch between the host address lines and the cache address lines. CALE is asserted to open the external latch, allowing the host address lines to propagate to the cache address lines. CALE is negated to latch the cache address lines.
CADS[1:0]#, CR/W[1:0]#	out	This signal pin has two functions, depending on the type of SRAMs used for the second level cache. **CACHE ADDRESS STROBE:** CADS[1:0]# are used with burst SRAMs. When asserted, CADS[1:0]# cause the burst SRAMs to latch the cache address on the rising edge of HCLK. CADS[1:0]# are glitch-free synchronous signals. CADS[1:0]# functionality is selected by the SRAM type bit in the Secondary Cache Control Register. Two copies of this signal are provided for timing reasons only. **CACHE READ/WRITE:** CR/W# provide read/write control to the second level cache when using asynchronous dual-byte select SRAMs. This functionality is selected by the SRAM Type and Cache Byte Control Bits in the Secondary Cache Control Register. The two copies of this signal are always driven to the same logic level.
CADV[1:0]#, CCS[1:0]#	out	This signal pin has two functions. The Cache Chip Select function is only enabled when the SRAM connectivity bit (bit 2) in the SCC Register is set to 1. **CACHE ADVANCE:** CADV[1:0]# are used with burst SRAMs to advance the internal two bit address counter inside the SRAMs to the next address of the burst sequence. Two copies of this signal are provided for timing reasons only. The two copies are always driven to the same logic level. **CACHE CHIP SELECT:** CCS[1:0]# are used with asynchronous SRAMs to de-select the SRAMs, placing them in a low power standby mode. When the CPU runs a halt or stop grant special cycle, the 82434NX negates CCS[1:0]#, placing the second level cache in a power saving mode. The PCMC then asserts CCS[1:0]# (activating the SRAMs) when the CPU asserts ADS#. When using burst SRAMs, only CCS1# implements the CCS# function. CADV0# retains the address advance function. CCS1# serve two purposes with burst SRAMs: 1) It is used (along with CADS[1:0]#) to place the SRAMs in a low power standby mode. When the CPU runs a halt or stop grant special cycle, the 82434NX negates CCS1# and asserts CADS[1:0]# for one clock, placing the SRAMs in a power saving mode. The PCMC then asserts CCS1# so that the next ADS# from the CPU places the SRAMs in an active mode. 2) CCS1# is used to block pipelined cycles from the SRAMs when the SRAMs are servicing a cycle. After NA# is asserted, the PCMC negates CCS1# preventing the SRAMs from sampling a new address. CCS1# is asserted again when the SRAMs have completed the current cycle.
CAA[6:3] CAB[6:3]	out	**CACHE ADDRESS [6:3]:** CAA[6:3] and CAB[6:3] are connected to address lines A[3:0] on the second level cache SRAMs. CAA[4:3] and CAB[4:3] are used with standard SRAMs to advance through the burst sequence. CAA[6:5] and CAB[6:5] are used during second level cache write-back cycles to address the modified lines within the addressed sector. Two copies of these signals are provided for timing reasons only. The two copies are always driven to the same logic level.

Signal	Type	Description
COE[1:0]#	out	**CACHE OUTPUT ENABLE:** COE[1:0]# are asserted when data is to be read from the second level cache and are negated at all other times. Two copies of this signal are provided for timing reasons only. The two copies are always driven to the same logic level.
CWE[7:0]#, CBS[7:0]#	out	This signal pin has two functions, depending on the type of SRAMs used for the second level cache. **CACHE WRITE ENABLES:** CWE[7:0]# are asserted to write data to the second level cache SRAMs on a byte-by-byte basis. CWE7# controls the most significant byte while CWE0# controls the least significant byte. These signals are cache write enables when using burst SRAMs (SRAM Type bit in SCC Register is 1) or when using asynchronous SRAMs (SRAM Type bit in SCC Register is 0) and the Cache Byte Control Bit is 1. **CACHE BYTE SELECTS:** The CBS[7:0]# lines provide byte control to the secondary cache when using dual-byte select asynchronous SRAMs. These signals are Cache Byte select lines when the SRAM Type and Cache Byte Control Bits in the SCC Register are both 0.

2.4 PCI Interface

Signal	Type	Description
C/BE[3:0]#	t/s	**PCI BUS COMMAND AND BYTE ENABLES:** C/BE[3:0]# are driven by the current bus master during the address phase of a PCI cycle to define the PCI command, and during the data phase as the PCI byte enables. The PCI commands indicate the current cycle type, and the PCI byte enables indicate which byte lanes carry meaningful data. C/BE[3:0]# are outputs of the PCMC during CPU cycles that are directed to PCI. C/BE[3:0]# are inputs when the PCMC acts as a slave. The command encodings and types are listed below. **C/BE[3:0]# Command** 0000 Interrupt Acknowledge 0001 Special Cycle 0010 I/O Read 0011 I/O Write 0100 Reserved 0101 Reserved 0110 Memory Read 0111 Memory Write 1000 Reserved 1001 Reserved 1010 Configuration Read 1011 Configuration Write 1100 Memory Read Multiple 1101 Reserved 1110 Memory Read Line 1111 Memory Write and Invalidate

Signal	Type	Description
FRAME#	s/t/s	**CYCLE FRAME:** FRAME# is driven by the current bus master to indicate the beginning and duration of an access. FRAME# is asserted to indicate that a bus transaction is beginning. While FRAME# is asserted, data transfers continue. When FRAME# is negated, the transaction is in the final data phase. FRAME# is an output of the PCMC during CPU cycles which are directed to PCI. FRAME# is an input to the PCMC when the PCMC acts as a slave.
IRDY#	s/t/s	**INITIATOR READY:** The assertion of IRDY# indicates the current bus master's ability to complete the current data phase. IRDY# works in conjunction with TRDY# to indicate when data has been transferred. On PCI, data is transferred on each clock that both IRDY# and TRDY# are asserted. During read cycles, IRDY# is used to indicate that the master is prepared to accept data. During write cycles, IRDY# is used to indicate that the master has driven valid data on the AD[31:0] lines. Wait states are inserted until both IRDY# and TRDY# are asserted together. IRDY# is an output of the PCMC when the PCMC is the PCI master. IRDY# is an input to the PCMC when the PCMC acts as a slave.
TRDY#	s/t/s	**TARGET READY:** TRDY# indicates the target device's ability to complete the current data phase of the transaction. It is used in conjunction with IRDY#. A data phase is completed on each clock that TRDY# and IRDY# are both sampled asserted. During read cycles, TRDY# indicates that valid data is present on AD[31:0] lines. During write cycles, TRDY# indicates the target is prepared to accept data. Wait states are inserted on the bus until both IRDY# and TRDY# are asserted together. TRDY# is an output of the PCMC when the PCMC is the PCI slave. TRDY# is an input to the PCMC when the PCMC is a master.
DEVSEL#	s/t/s	**DEVICE SELECT:** When asserted, DEVSEL# indicates that the driving device has decoded its address as the target of the current access. DEVSEL# is an output of the PCMC when PCMC is a PCI slave and is derived from the MEMCS# input. MEMCS# is generated by the expansion bus bridge as a decode to the main memory address space. During CPU-to-PCI cycles, DEVSEL# is an input. It is used to determine if any device has responded to the current bus cycle, and to detect a target abort cycle. Master-Abort termination results if no subtractive decode agent exists in the system, and no one asserts DEVSEL# within a programmed number of clocks.
STOP#	s/t/s	**STOP:** STOP# indicates that the current target is requesting the master to stop the current transaction. This signal is used in conjunction with DEVSEL# to indicate disconnect, target-abort, and retry cycles. When PCMC is acting as a master on PCI, if STOP# is sampled active on a rising edge of PCLKIN, FRAME# is negated within a maximum of 3 clock cycles. STOP# may be asserted by the PCMC in three cases. If a PCI master attempts to access main memory when another PCI master has locked main memory, the PCMC asserts STOP# to signal retry. The PCMC detects this condition when sampling FRAME# and LOCK# both active during an address phase. When a PCI master is reading from main memory, the PCMC asserts STOP# when the burst cycle is about to cross a cache line boundary. When a PCI master is writing to main memory, the PCMC asserts STOP# upon filling either of the two PCI-to-main memory posted write buffers. Once asserted, STOP# remains asserted until FRAME# is negated.

Signal	Type	Description
PLOCK#	s/t/s	**PCI LOCK:** PLOCK# is used to indicate an atomic operation that may require multiple transactions to complete. PCI provides a mechanism referred to as "resource lock" in which only the target of the PCI transaction is locked. The assertion of GNT# on PCI does not guarantee control of the PLOCK# signal. Control of PLOCK# is obtained under its own protocol. When the PCMC is the PCI slave, PLOCK# is sampled as an input on the rising edge of PCLKIN when FRAME# is sampled active. If PLOCK# is sampled asserted, the PCMC enters into a locked state and remains in the locked state until PLOCK# is sampled negated on a following rising edge of PCLKIN, when FRAME# is sampled asserted.
REQ#	out	**REQUEST:** The PCMC asserts REQ# to indicate to the PCI bus arbiter that the PCMC is requesting use of the PCI Bus in response to a CPU cycle directed to PCI.
GNT#	in	**GRANT:** When asserted, GNT# indicates that access to the PCI Bus has been granted to the PCMC by the PCI Bus arbiter.
MEMCS#	in	**MAIN MEMORY CHIP SELECT:** When asserted, MEMCS# indicates to the PCMC that a PCI master cycle is targeting main memory. MEMCS# is generated by the expansion bus bridge. MEMCS# is sampled by the PCMC on the rising edge of PCLKIN on the first and second cycle after FRAME# has been asserted.
FLSHREQ#	in	**FLUSH REQUEST:** When asserted, FLSHREQ# instructs the PCMC to flush the CPU-to-PCI posted write buffer in the LBXs and to disable further posting to this buffer as long as FLSHREQ# remains active. The PCMC acknowledges completion of the CPU-to-PCI write buffer flush operation by asserting MEMACK#. MEMACK# remains asserted until FLSHREQ# is negated. FLSHREQ# is driven by the expansion bus bridge and is used to avoid deadlock conditions on the PCI Bus.
MEMREQ#	in	**MEMORY REQUEST:** When asserted, MEMREQ# instructs the PCMC to flush the CPU-to-PCI and CPU-to-main memory posted write buffers and to disable posting in these buffers as long as MEMREQ# is active. The PCMC acknowledges completion of the flush operations by asserting MEMACK#. MEMACK# remains asserted until MEMREQ# is negated. MEMREQ# is driven by the expansion bus bridge.
MEMACK#	out	**MEMORY ACKNOWLEDGE:** When asserted, MEMACK# indicates the completion of the operations requested by an active FLSHREQ# and/or MEMREQ#.
PAR	t/s	**PARITY:** PAR is an even parity bit across the AD[31:0] and C/BE[3:0]# lines. Parity is generated on all PCI transactions. As a master, the PCMC generates even parity on CPU writes to PCI, based on the PPOUT[1:0] inputs from the LBXs. During CPU read cycles from PCI, the PCMC checks parity by checking the value sampled on the PAR input with the PPOUT[1:0] inputs from the LBXs. As a slave, the PCMC generates even parity on PAR, based on the PPOUT[1:0] inputs during PCI master reads from main memory. During PCI master writes to main memory, the PCMC checks parity by checking the value sampled on PAR with the PPOUT[1:0] inputs.

Signal	Type	Description
PERR#	s/t/s	**PARITY ERROR:** PERR# may be pulsed by any agent that detects a parity error during an address phase, or by the master or the selected target during any data phase in which the AD lines are inputs. The PERR# signal is enabled when the PERR# on Receiving Data Parity Error bit in the Error Command Register (offset 70h) and the Parity Error Enable bit in the PCI Command Register (offset 04h) are both set to 1. When enabled, CPU-to-PCI write data is checked for parity errors by sampling the PERR# signal two PCI clocks after data is driven. Also, when enabled, PERR# is asserted by the PCMC when it detects a data parity error on CPU read data from PCI and PCI master write data to main memory. PERR# is neither sampled nor driven by the PCMC when either the PERR# on Receiving Data Parity Error bit in the Error Command Register or the Parity Error Enable bit in the PCI Command Register is reset to 0.
SERR#	o/d	**SYSTEM ERROR:** SERR# may be pulsed by any agent for reporting errors other than parity. SERR# is asserted by the PCMC whenever a serious system error (not necessarily a PCI error) occurs. The intent is to have the PCI central agent (for example, the expansion bus bridge) assert NMI to the processor. Control over the SERR# signal is provided via the Error Command Register (offset 70h) when the Parity Error Enable bit in the PCI Command Register (offset 04h) is set to 1. When the SERR# DRAM/L2 Cache Data Parity Error bit is set to 1, SERR# is asserted upon detecting a parity error on CPU read cycles from DRAM. If the L2 Cache Parity bit is also set to 1, SERR# will be asserted upon detecting a parity error on CPU read cycles from the second level cache. The Pentium processor indicates these parity errors to the PCMC via the PCHK# signal. When the SERR# on PCI Address Parity Error bit is set to 1, the PCMC asserts SERR# if a parity error is detected during the address phase of a PCI master cycle. When the SERR# on Received PCI Data Parity bit is set to 1, the PCMC asserts SERR# if a parity error is detected on PCI during a CPU read from PCI. During CPU to PCI write cycles, when the SERR# on Transmitted PCI Data Parity Error bit is set to 1, the PCMC asserts SERR# in response to sampling PERR# active. When the SERR# on Received Target Abort bit is set to 1, the PCMC asserts SERR# when the PCMC receives a target abort on a PCMC initiated PCI cycle. If the Parity Error Enable bit in the PCI Command Register is reset to 0, SERR# is disabled and is never asserted by the PCMC.

2.5 LBX Interface

Signal	Type	Description
HIG[4:0]	out	**HOST INTERFACE GROUP:** HIG[4:0] are outputs of the PCMC used to control the LBX HA (Host Address) and HD (Host Data) buses. Commands driven on HIG[4:0] cause the host data and/or address lines to be either driven or latched by the LBXs. See the 82433LX (LBX) Local Bus Accelerator Data Sheet for a listing of the HIG[4:0] commands.
MIG[2:0]	out	**MEMORY INTERFACE GROUP:** MIG[2:0] are outputs of the PCMC and control the LBX MD (Memory Data) bus. Commands driven on the MIG[2:0] lines cause the memory data lines to be either driven or latched by the LBXs. See the 82433LX (LBX) Local Bus Accelerator Data Sheet for a listing of the MIG[2:0] commands.
MDLE	out	**MEMORY DATA LATCH ENABLE:** During CPU reads from main memory, MDLE is used to control the latching of memory read data on the CPU data bus. MDLE is negated as CAS[7:0]# are negated to close the latch between the memory data bus and the host data bus. During CPU reads from main memory, the PCMC closes the memory data to host data latch in the LBXs as BRDY# is asserted and opens the latch after the CPU has sampled the data.
PIG[3:0]	out	**PCI INTERFACE GROUP:** PIG[3:0] are outputs of the PCMC used to control the LBX AD (PCI Address/Data) bus. Commands driven on the PIG[3:0] lines cause the AD lines to be either driven or latched. See the 82433LX (LBX) Local Bus Accelerator Data Sheet for a listing of the PIG[3:0] commands.
DRVPCI	out	**DRIVE PCI:** DRVPCI acts as an output enable for the LBX AD lines. When sampled asserted, the LBXs begin driving the PCI AD lines. When negated, the AD lines on the LBXs are tri-stated. The LBX AD lines are tri-stated asynchronously from the falling edge of DRVPCI.
EOL	in	**END OF LINE:** EOL is asserted by the low order LBX when a PCI master read or write transaction is about to overrun a cache line boundary. EOL has an internal pull-up resistor inside the PCMC. The low order LBX EOL signal connects to this PCMC input. The high order LBX EOL signal is connected to ground through an external pull-down resistor.
PPOUT[1:0]	in	**PCI PARITY OUT:** These signals reflect the parity of the 32 AD lines driven from or latched in the LBXs, depending on the command driven on PIG[3:0]. The PPOUT0 pin has a weak internal pull-down resistor. The PPOUT1 pin has a weak internal pull-up resistor.

2.6 Reset And Clock

Signal	Type	Description
HCLKOSC	in	**HOST CLOCK OSCILLATOR:** The HCLKOSC input is driven externally by a crystal oscillator. The PCMC generates six copies of HCLK from HCLKOSC (HCLKA–HCLKF). During power-up, HCLKOSC must stabilize for 1 ms before PWROK is asserted. If an external clock driver is used to clock the CPU, PCMC, LBXs and second level cache SRAMs instead of the HCLKA–HCLKF outputs, HCLKOSC must be tied either high or low.
HCLKA–HCLKF	out	**HOST CLOCK OUTPUTS:** HCLKA–HCLKF are six low skew copies of the host clock. These outputs eliminate the need for an external low skew clock driver.

Signal	Type	Description
HCLKIN	in	**HOST CLOCK INPUT:** All timing on the host, DRAM and second level cache interfaces is based on HCLKIN. If an external clock driver is used to clock the CPU, PCMC, LBXs and second level cache SRAMs, the externally generated clock must be connected to HCLKIN. During power-up HCLKIN must stabilize for 1 ms before PWROK is asserted.
CPURST	out	**CPU HARD RESET:** The CPURST pin is asserted in response to one of two conditions. **Powerup** 82434LX: During powerup the 82434LX asserts CPURST when PWROK is negated. When PWROK is asserted, the 82434LX first ensures that it has been initialized before negating CPURST. 82434NX: During powerup, the 82434NX PCMC negates CPURST while PWROK is negated. When PWROK is asserted, the 82434NX asserts CPURST for 2 ms. **Software** CPURST is also asserted when the System Hard Reset Enable bit in the Turbo-Reset Control Register (I/O address 0CF9h) is set to 1 and the Reset CPU bit toggles from 0 to 1 (82434LX and 82434NX). CPURST is driven synchronously to the rising edge of HCLKIN.
INIT	out	**INITIALIZATION:** INIT is asserted in response to any one of two conditions. When the System Hard Reset Enable bit in the Turbo-Reset Control Register is reset to 0 and the Reset CPU bit toggles from 0 to 1, the PCMC initiates a soft reset by asserting INIT. The PCMC also initiates a soft reset by asserting INIT in response to a shutdown special cycle. In both cases, INIT is asserted for a minimum of 2 Host clocks.
PWROK	in	**POWER OK:** When asserted, PWROK is an indication to the PCMC that power and HCLKIN have stabilized for at least 1 ms. PWROK can be driven asynchronously. 82434LX: When PWROK is negated, the 82434LX asserts both CPURST and PCIRST#. When PWROK is driven high, the 82434LX ensures that it is initialized before negating CPURST and PCIRST#. 82434NX: When PWROK is negated, the 82434NX negates CPURST and asserts PCIRST#. When PWROK is asserted, the 82434NX asserts CPURST for 2 ms. PCIRST# is negated 1 ms after PWROK is asserted.
PCLKOUT	out	**PCI CLOCK OUTPUT:** PCLKOUT is internally generated by a Phase Locked Loop (PLL) that divides the frequency of HCLKIN by 2. This output must be buffered externally to generate multiple copies of the PCI Clock. One of the copies must be connected to the PCLKIN pin.

Signal	Type	Description
PCLKIN	in	**PCI CLOCK INPUT:** An internal PLL locks PCLKIN in phase with HCLKIN. All timing on the PCMC PCI interface is referenced to the PCLKIN input. All output signals on the PCI interface are driven from PCLKIN rising edges and all input signals on the PCI interface are sampled on PCLKIN rising edges.
PCIRST#	out	**PCI RESET:** PCIRST# is asserted to initiate hard reset on PCI. PCIRST# is asserted in response to one of two conditions. **Power-up** During power-up the PCMC asserts PCIRST# when PWROK is negated. 82434LX: When PWROK is asserted the PCMC will first ensure that it has been initialized before negating PCIRST#. 82434NX: When PWROK is negated, the 82434NX asserts PCIRST#. The 82434NX then negates PCIRST# 1 ms after PWROK is asserted. **Software** PCIRST# is also asserted when the System Hard Reset Enable bit in the Turbo/Reset Control Register is set to 1 and the Reset CPU bit toggles from 0 to 1 (82434LX and 82434NX). PCIRST# is driven asynchronously.
TESTEN	in	**TEST ENABLE:** TESTEN must be tied low for normal system operation.

3.0 REGISTER DESCRIPTION

The 82434LX/82434NX PCMC contains two sets of software accessible registers. These registers are accessed via the Host CPU I/O address space. The PCMC also contains a set of configuration registers that reside in PCI configuration space and are used to specify PCI configuration, DRAM configuration, cache configuration, operating parameters and optional system features (see Section 3.2, PCI Configuration Space Mapped Registers). The PCMC internal registers (both I/O Mapped and Configuration registers) are only accessible by the Host CPU and cannot be accessed by PCI masters. The registers can be accessed as Byte, Word (16-bit), or Dword (32-bit) quantities. All multi-byte numeric fields use "little-endian" ordering (i.e., lower addresses contain the least significant parts of the field).

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

In addition to reserved bits within a register, the PCMC contains address locations in the PCI configuration space that are marked "Reserved" (Table 1). The PCMC responds to accesses to these address locations by completing the Host cycle. When a reserved register location is read, 0000h is returned. Writes to reserved registers have no affect on the PCMC.

Upon receiving a hard reset via the PWROK signal, the PCMC sets its internal configuration registers to predetermined **default** states. The default state represents the minimum functionality feature set required to successfully bring up the system. Hence, it does not represent the optimal system configuration. It is the responsibility of the system initialization software (usually BIOS) to properly determine the DRAM configurations, cache configuration, operating parameters and optional system features that are applicable, and to program the PCMC registers accordingly.

The following nomenclature is used for access attributes.

RO **Read Only.** If a register is read only, writes to this register have no effect.

R/W **Read/Write.** A register with this attribute can be read and written.

R/WC **Read/Write Clear.** A register bit with this attribute can be read and written. However, a write of a 1 clears (sets to 0) the corresponding bit and a write of a 0 has no effect.

3.1 I/O Mapped Registers

The 82434LX PCMC contains three registers that reside in the CPU I/O address space—the Configuration Space Enable (CSE) Register, the Turbo-Reset Control (TRC) Register and the Forward (FORW) Register. These registers can not reside in PCI configuration space because of the special functions they perform. The CSE Register enables/disables the configuration space and, hence, can not reside in that space. The TRC Register enables/disables deturbo mode which effectively slows the processor to accommodate software programs that rely on the slow speed of PC/XT systems to time certain events. The FORW Register determines which of the possible hierarchical PCI Buses a cycle is directed. The 82434LX uses mechanism #2 for accessing PCI configuration space.

The 82434NX PCMC contains five registers that reside in the CPU I/O address spacethe Configuration Address (CONFADD) Register, the Configuration Space Enable (CSE) Register, the Turbo-Reset Control (TRC) Register, the Forward (FORW) Register, and the PCI Mechanism Control (PMC) Register. The CSE, TRC, and FORW Registers are the same for both the 82434LX and 82434NX PCMCs. The 82434NX can use either Configuration Access Mechanism #1 or #2 for accessing PCI configuration space. When Configuration Access Mechanism #1 is used (See Section 3.2, PCI Configuration Space Mapped Registers), The CONFADD Register enables/disables the configuration space and determines what portion of configuration space is visible through the Configuration Data (CONFDATA) window. The CSE and FORW Registers are used for Configuration Access Mechanism #2. The PCI Mechanism Control (PMC) Register selects whether Configuration Access Mechanism 1 or 2 is used (see the Rev 2.0 PCI Local Bus Specification).

3.1.1 CONFADD—CONFIGURATION ADDRESS REGISTER

I/O Address: 0CF8h Accessed as a Dword
Default Value: 00000000h
Access: Read/Write
Size: 32 bits

CONFADD is a 32-bit register used in Configuration Access Mechanism #1. It is accessed only when referenced as a Dword and PCAMS in the PMC Register is set to 1. Byte or Word references "pass through" the CONFADD Register to the I/O locations "behind" it. For example a byte access to 0CF8h will access the CSE Register, while a word access to CF8h will access both the CSE and TRC Registers. The CONFADD Register contains the Bus Number, Device Number, Function Number, and Register Number where the CONFDATA window is located.

Bit	Description
31	**CONFIGURATION ENABLE (CONE)—R/W:** When CONE = 1, accesses to PCI configuration space are enabled, if the PCAMS bit of the PMC register is also 1. When CONE = 0, accesses to PCI configuration space are disabled, if the PCAMS bit is 1. If the PCAMS bit is 0, this bit has no effect.
30:24	**RESERVED**
23:16	**BUS NUMBER (BUSNUM)—R/W:** When the BUSNUM is programmed to 00h, the target of the Configuration Cycle is either the PCMC or the PCI Local Bus that is directly connected to the PCMC. PCI Access Mechanism #1 can generate either type 0 or type 1 configuration cycles on PCI. A type 0 Configuration Cycle is generated on PCI if the Bus Number is programmed to 00h and the PCMC is not the target. If the Bus Number is non-zero a type 1 configuration cycle is generated on PCI with the Bus Number mapped to AD[23:16] during the address phase.
15:11	**DEVICE NUMBER (DEVNUM)—R/W:** This field selects one agent on the PCI Bus selected by the Bus Number. During a Type 1 Configuration cycle this field is mapped to AD[15:11]. During a Type 0 Configuration Cycle this field is decoded and one of AD[31:17] is driven to a 1. The PCMC is always Device Number 0.
10:8	**FUNCTION NUMBER (FUNCNUM)—R/W:** This field is mapped to AD[10:8] during PCI configuration cycles. This allows the configuration registers of a particular function in a multi-function device to be accessed.
7:2	**REGISTER NUMBER (REGNUM)—R/W:** This field selects one register within a particular Bus, Device, and Function as specified by the other fields in the Configuration Address Register. REGNUM is mapped to AD[7:2] during PCI configuration cycles.
1:0	**RESERVED**

3.1.2 CSE—CONFIGURATION SPACE ENABLE REGISTER

I/O Address: 0CF8h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

The CSE Register enables/disables configuration space access and provides access to specific functions within a PCI agent. The register is located in the CPU I/O address space. The PCMC, as a Host/PCI Bridge, supports multi-function devices on the PCI Bus. The function number permits individual configuration spaces for up to eight functions within an agent. The register is located in the CPU I/O address space.

Bit	Description
7:4	**KEY FIELD (KEY)—R/W:** This field is used only when the PCI Mechanism Control Register (PMC) indicates Configuration Access Mechanism 2 is to be used. When the key field is programmed to 0h, the PCI configuration space is disabled. When the key field is programmed to a non-zero value, all CPU accesses to CnXXh (where n is a non zero value) are forwarded to PCI as configuration space accesses. Additionally, when the key field is programmed to a non-zero value, all CPU accesses to C0XXh are intercepted by the PCMC and directed to a PCMC internal register.
3:1	**FUNCTION NUMBER (FN)—R/W:** For multi-function devices, this field selects a particular function within a PCI device. During a configuration cycle, bits[3:1] become part of the PCI Bus address and correspond to AD[10:8].
0	**RESERVED**

3.1.3 TRC—TURBO-RESET CONTROL REGISTER

I/O Address: 0CF9h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

The TRC Register is an 8-bit read/write register that selects turbo/deturbo mode of the CPU, initiates PCI Bus and CPU reset cycles, and initiates the CPU Built In Self Test (BIST). TRC is located in CPU I/O address space.

Bit	Description
7:3	**RESERVED**
2	**RESET CPU (RCPU)—R/W:** RCPU is used to initiate a hard reset or soft reset to the CPU. During a hard reset, the PCMC asserts CPURST and PCIRST#. The PCMC initiates a hard reset when this register is programmed for a hard reset or when the PWROK signal is asserted. During a soft reset, the PCMC asserts INIT. The PCMC initiates a soft reset when this register is programmed for a soft reset and in response to a shutdown special cycle. Note that a hard reset initializes the entire system and invalidates the CPU cache. A soft reset initializes only the CPU. The contents of the CPU cache are unaffected. This bit is used in conjunction with bit 1 of this register. Bit 1 must be set up prior to writing a 1 to this register. Thus, two write operations are required to initiate a reset using this bit. The first write operation programs bit 1 to the appropriate state while setting this bit to 0. The second write operation keeps bit 1 at the programmed state (1 or 0) while setting this bit to a 1. When RCPU transitions from a 0 to a 1, a hard reset is initiated if bit 1 = 1 and a soft reset is initiated if bit 1 = 0.
1	**SYSTEM HARD RESET ENABLE (SHRE)—R/W:** This bit is used in conjunction with bit 2 of this register to initiate either a hard or soft reset. When SHRE = 1, the PCMC initiates a hard reset to the CPU when bit 2 transitions from 0 to 1. When SHRE = 0, the PCMC initiates a soft reset when bit 2 transitions from 0 to 1.
0	**DETURBO MODE (DM)—R/W:** This bit enables and disables deturbo mode. When DM = 1, the PCMC is in the deturbo mode. In this mode, the PCMC periodically asserts the AHOLD signal to slow down the effective speed of the CPU. The AHOLD duty cycle is programmable through the Deturbo Frequency Control (DFC) Register. When DM = 0, the deturbo mode is disabled. Deturbo mode can be used to maintain backward compatibility with older software packages that rely on the operating speed of older processors. For accurate speed emulation, caching should be disabled. If caching is disabled during runtime, the following steps should be performed to make sure that modified lines have been flushed from the cache to main memory before entering deturbo mode. Disable the primary cache via the PCE bit in the HCS Register. This prevents the KEN# signal from being asserted, which prevents any further first and second level cache line fills. At this point, software executes the WBINVD instruction to flush the caches, and then sets DM to 1. When exiting the deturbo mode, the system software must first set DM to 0, then enable first and second level caching by writing to the HCS Register.

3.1.4 FORW—FORWARD REGISTER

I/O Address: 0CFAh
Default Value: 00h
Attribute: Read/Write
Size: 8 Bits

This 8-bit register specifies which PCI Bus configuration space is enabled in a multiple PCI Bus configuration. The default value for the FORW Register enables the configuration space of the PCI Bus connected to the PCMC.

Bit	Description
7:0	**FORWARD BUS NUMBER—R/W:** When this register value is 00h, the configuration space of the PCI Bus connected to the PCMC is enabled and the PCMC initiates a type 0 configuration cycle. If the value of this register is not 00h, the PCMC initiates a type 1 configuration cycle to forward the cycle (via one or more PCI/PCI Bridges) to the PCI Bus specified by the contents of this register. For non-zero values, bits[7:0] are mapped to AD[23:16], respectively.

3.1.5 PMC—PCI MECHANISM CONTROL REGISTER

I/O Address: 0CFBh
Default Value: 00h
Access: Read/Write
Size: 8 bits

The PMC Register selects whether PCI Configuration Access Mechanism 1 or 2 is to be used. The register is located in the CPU I/O address space.

Bit	Description
7:1	**RESERVED**
0	**PCI CONFIGURATION ACCESS MECHANISM SELECT (PCAMS)—R/W:** When PCAMS = 0, the PCMC uses to PCI Configuration Access Mechanism #2. When PCAMS = 1, the PCMC uses to PCI Configuration Access Mechanism #1. The CONFADD and CONFDATA Registers are only accessible when PCAMS = 1.

3.1.6 CONFDATA—CONFIGURATION DATA REGISTER

I/O Address: 0CFCh
Default Value: 00h
Access: Read/Write
Size: 32 bits

CONFDATA is a 32 bit read/write window into configuration space. The portion of configuration space that is referenced by CONFDATA is determined by the contents of CONFADD.

Bit	Description
31:0	**CONFIGURATION DATA WINDOW (CDW)—R/W:** When using Configuration Access Mechanism #1 if bit 31 of CONFADD is 1 any I/O reference that falls in the CONFDATA I/O space will be mapped to configuration space using the contents of CONFADD.

3.2 PCI Configuration Space Mapped Registers

The PCI Bus defines a slot based "configuration space" that allows each device to contain up to 256 8-bit configuration registers. The PCI specification defines two bus cycles to access the PCI configuration space—**Configuration Read** and **Configuration Write**. While memory and I/O spaces are supported by the Pentium processor, configuration space is not supported. For PCI configuration space access, the PCMC translates the Pentium processor I/O cycles into PCI configuration cycles. Table 1 shows the PCMC configuration space.

Table 1. PCMC Configuration Space

Address Offset	Register Symbol	Register Name	Access
00–01h	VID	Vendor Identification	RO
02–03h	DID	Device Identification	RO
04–05h	PCICMD	Command Register	R/W
06–07h	PCISTS	Status Register	RO, R/WC
08h	RID	Revision Identification	RO
09h	RLPI	Register-Level Programming Interface	RO
0Ah	SCCD	Sub-Class Code	RO
0Bh	BCCD	Base Class Code	RO
0Ch	—	Reserved	—
0Dh	MLT	Master Latency Timer	R/W
0Eh	—	Reserved	—
0Fh	BIST	BIST Register	RO
10–4Fh	—	Reserved	—
50h	HCS	Host CPU Selection	R/W
51h	DFC	Deturbo Frequency Control	R/W
52h	SCC	Secondary Cache Control	R/W
53h	HBC	Host Read/Write Buffer Control	R/W
54h	PBC	PCI Read/Write Buffer Control	R/W
55h	—	Reserved	—
56h	—	Reserved	—
57h	DRAMC	DRAM Control	R/W
58h	DRAMT	DRAM Timing	R/W
59–5Fh	PAM[6:0]	Programmable Attribute Map (7 Registers)	R/W
60–65h	DRB[5:0]	DRAM Row Boundary (6 Registers)	R/W
66–67h	DRB[7:6]	DRAM Row Boundary (2 Registers)	R/W
68–6Bh	DRBE	DRAM Row Boundary Extension	R/W
6C–6Fh	—	Reserved	—
70h	ERRCMD	Error Command	R/W

Table 1. PCMC Configuration Space (Continued)

Address Offset	Register Symbol	Register Name	Access
71h	ERRSTS	Error Status	R/WC
72h	SMRS	SMRAM Space Control	R/W
73–77h	—	Reserved	—
78–79h	MSG	Memory Space Gap	R/W
7A–7B	—	Reserved	—
7C–7Fh	FBR	Frame Buffer Range	R/W
80–FFh	—	Reserved	—

NOTE:
Shaded rows indicate register differences between the 82434LX and 82434NX devices. For non-shaded rows, the registers are the same for the two devices.

3.2.1 CONFIGURATION SPACE ACCESS MECHANISM

The 82434LX supports Configuration Space Access Mechanism #2 and the 82434NX supports both configuration space access mechanisms #1 and #2. The mechanism is selected via the PCAMS bit in the PMC Register. The bus cycles used to access PCMC internal configuration registers are described in Section 7.0, PCI Interface.

3.2.1.1 Access Mechanism #1:

For configuration access mechanism #1, the 82434NX PCMC uses the CONFADD and CONFDATA Registers. Note that while the CONFADD and PMC Register address spaces overlap, the CONFADD Register is referenced only by a Dword read or write to CF8h. This allows the PMC Register to be accessed by a byte write to CFBh, even when using configuration access mechanism #1.

To reference a configuration register with access mechanism #1, a Dword I/O write loads the CONFADD Register with a 32-bit value that specifies the PCI Bus, the device on that bus, the function within the device, and a specific configuration register of the device function being accessed (Figure 4). Bit 31 of the CONFADD Register must be 1 to enable a configuration cycle. CONFDATA then becomes a four byte window of configuration space specified by the contents of the CONFADD Register. A read or write to CONFDATA results in the PCMC translating CONFADD into a PCI configuration cycle.

Type 0 Access

If the BUSNUM field is 0, a Type 0 configuration cycle is performed on the PCI. Bus CONFADD[10:2] are mapped directly to AD[10:2]. The DEVNUM field is decoded onto AD[31:17] and AD[15:11] (for accesses to device 1, AD17 is asserted; for accesses to device #2, AD18 is asserted; etc.). The PCMC is Device #0 and does not pass its configuration cycles to the PCI Bus. Thus, AD16 is never asserted. For accesses to device 15, AD31 is asserted, etc. This mapping allows the same Device Number to activate the same AD line in either configuration access mechanism. All other AD lines are 0.

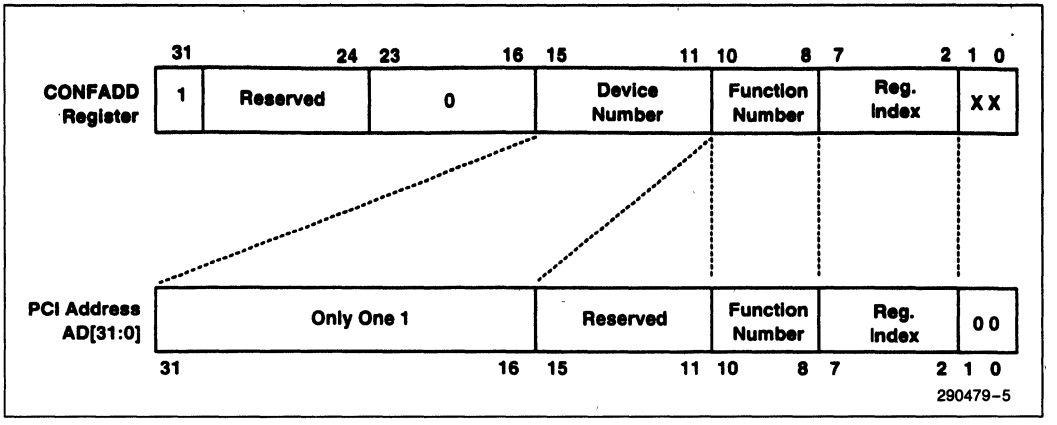


Figure 4. Mechanism #1 Type 0 Configuration Address to PCI Address Mapping

2

Type 1 Access

If the BUSNUM field of the CONFADD Register is non-zero, a Type 1 configuration cycle is performed on the PCI Bus. CONFADD[23:2] are mapped directly to AD[23:2] (Figure 5). AD[1:0] are driven to 01 to indicate a Type 1 Configuration cycle. All other lines are driven to 0.

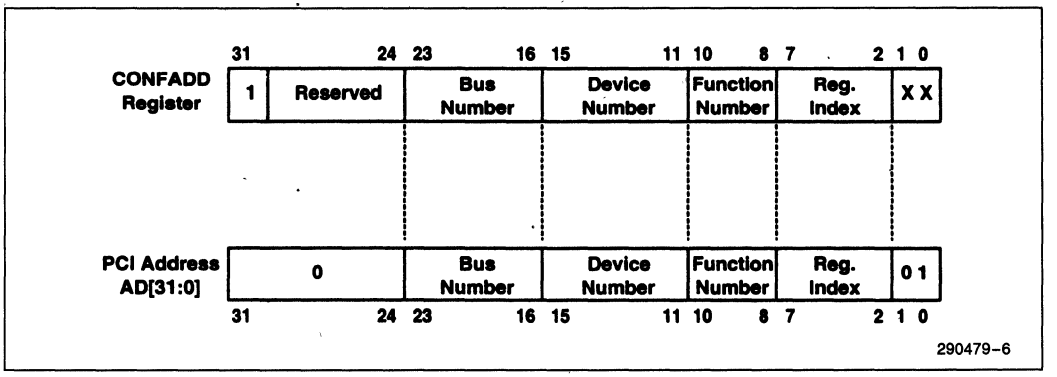


Figure 5. Mechanism #1 Type 1 Configuration Address to PCI Address Mapping

3.2.1.2 Access Mechanism #2

The 82434LX/82434NX PCMC uses the CSE and Forward Registers for configuration access mechanism #2. When PCI configuration space is enabled via the CSE Register, the PCMC maps PCI configuration space into 4-KBytes of CPU I/O space. Each PCI device has its own 256-Byte configuration space. When configuration space is enabled, CPU accesses to I/O locations CXXXh are translated into configuration space accesses. In this mode, the PCMC translates all I/O cycles in the C100h–CFFFh range into configuration cycles on the PCI Bus. I/O accesses within the C000h–C0FFh range are intercepted by the PCMC and are directed to the PCMC internal configuration registers. These cycles are not forwarded to the PCI Bus.

When configuration space access is disabled, CPU accesses to I/O locations CXXXh are forwarded to the PCI Bus I/O space. CPU cycles to I/O locations other than CXXXh are unaffected by whether the configuration mode is enabled or disabled. These cycles are always treated as ordinary I/O cycles by the PCMC.

Type 0 Access

If the Forward Register contains 00h a Type 0 configuration access is generated on the PCI Bus (Figure 6). For type 0 configuration cycles, AD[1:0] = 00. Host CPU address bits A[7:2] are not translated and become AD[7:2] on the PCI Bus. AD[7:2] select one of the 256 8-bit I/O locations in the PCI configuration space. The FUNCTION NUMBER field from the CSE Register (CSE[3:1]) is driven on AD[10:8]. Host CPU address bits A[11:8] are mapped to an IDSEL input for each of the 16 possible PCI devices. The IDSEL input for each PCI device must be hard-wired to one of the AD[31:16] signals on the PCI Bus. AD16 is reserved for the PCMC. When CPU address A[11:8] = Fh, PCI address bits A31 = 1 and A[30:16] = 00h. Other devices on the PCI Bus should not use AD16. Note that when A[11:8] = 0h, an access to the PCMC internal registers occurs and the cycle is not forwarded to the PCI Bus.

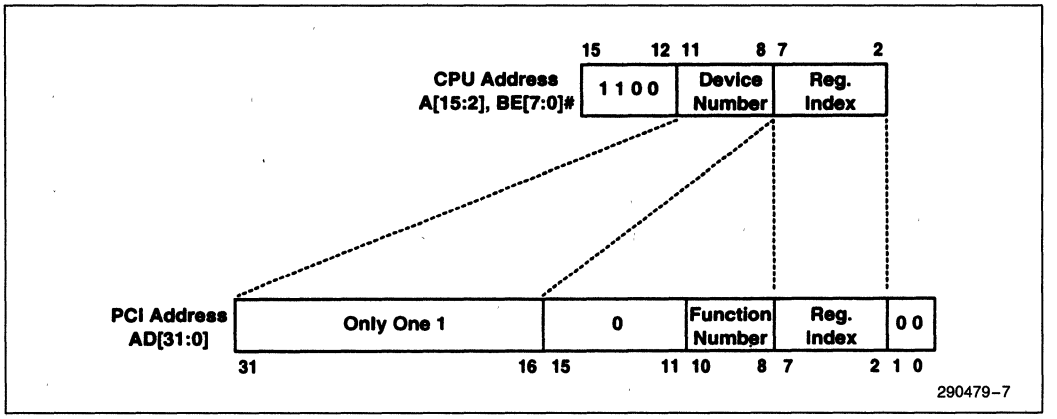


Figure 6. Mechanism #2 Type 0 Host-to-PCI Address Mapping

Type 1 Access

If the Forward Register is non-zero a Type 1 configuration access is generated on PCI. For type 1 configuration cycles, AD[1:0] = 01. AD[10:2] are generated the same as for the type 0 configuration cycle. Host CPU address bits A[11:8] contain the specific device number and are mapped to AD[14:11]. AD[23:16] contain the Bus Number of the PCI Bus that is to be accessed and corresponds to the Forward Address Register bits [7:0].

During a Type 1 configuration access AD[1:0] = 01 (Figure 7). The Register Index and Function Number are mapped to the AD lines the same way in Type 1 configuration access as in a Type 0 configuration access. CPU address bits A[11:8] are mapped directly to PCI lines AD[14:11] as the Device Number. The contents of the Forward Register are mapped to AD[23:16] to form the Bus Number.

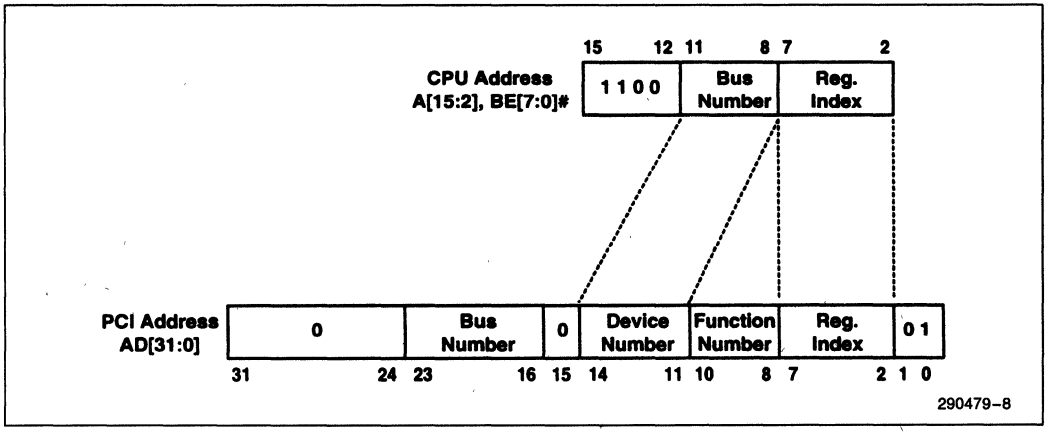


Figure 7. Mechanism #2 Type 1 Host-to-PCI Address Mapping

3.2.2 VID—VENDOR IDENTIFICATION REGISTER

Address Offset: 00–01h
Default Value: 8086h
Attribute: Read Only
Size: 16 bits

The VID Register contains the vendor identification number. This 16-bit register combined with the Device Identification Register uniquely identify any PCI device. Writes to this register have no effect.

Bits	Description
15:0	**VENDOR IDENTIFICATION NUMBER:** This is a 16-bit value assigned to Intel.

3.2.3 DID—DEVICE IDENTIFICATION REGISTER

Address Offset: 02–03h
Default Value: 04A3h
Attribute: Read Only
Size: 16 bits

This 16-bit register combined with the Vendor Identification Register uniquely identifies any PCI device. Writes to this register have no effect.

Bits	Description
15:0	**DEVICE IDENTIFICATION NUMBER:** This is a 16 bit value assigned to the PCMC.

3.2.4 PCICMD—PCI COMMAND REGISTER

Address Offset: 04–05h
Default: 06h
Attribute: Read/Write
Size: 16 bits

This 16-bit register provides basic control over the PCMC's ability to respond to PCI cycles. The PCICMD Register enables and disables the SERR# signal, the parity error signal (PERR#), PCMC response to PCI special cycles, and enables and disables PCI master accesses to main memory.

Bits	Description
15:9	**RESERVED**
8	**SERR# ENABLE (SERRE):** SERRE enables/disables the SERR# signal. When SERRE = 1 and PERRE = 1, SERR# is asserted if the PCMC detects a PCI Bus address/data parity error, or main memory (DRAM) or cache parity error, and the corresponding errors are enabled in the Error-Command Register. When SERRE = 1 and bit 7 in the Error Command Register is set to 1, the PCMC asserts SERR# when it detects a target abort on a PCMC-initiated PCI cycle. When SERRE = 0, SERR# is never asserted.
7	**RESERVED**
6	**PARITY ERROR ENABLE (PERRE):** PERRE controls the PCMC's response to PCI parity errors. This bit is a master enable for bit 3 of the ERRCMD Register. PERRE works in conjunction with the SERRE bit to enable SERR# assertion when the PCMC detects a PCI bus parity error, or a main memory or cache parity error.
5:3	**RESERVED**
2	**BUS MASTER ENABLE (BME):** The PCMC does not support disabling of its bus master capability on the PCI Bus. This bit is always set to 1, permitting the PCMC to function as a PCI Bus master. Writes to this bit position have no affect.
1	**MEMORY ACCESS ENABLE (MAE):** This bit enables/disables PCI master access to main memory (DRAM). When MAE = 1, the PCMC permits PCI masters to access main memory if the MEMCS# signal is asserted. When MAE = 0, the PCMC does not respond to PCI master main memory accesses (MEMCS# asserted).
0	**I/O ACCESS ENABLE (IOAE):** The PCMC does not respond to PCI I/O cycles, hence this command is not supported. PCI master access to I/O space on the Host Bus is always disabled.

3.2.5 PCISTS—PCI STATUS REGISTER

Address Offset: 06–07h
Default Value: 40h
Attribute: Read Only, Read/Write Clear
Size: 16 bits

PCISTS is a 16-bit status register that reports the occurrence of a PCI master abort, PCI target abort, and DRAM or cache parity error. PCISTS also indicates the DEVSEL# timing that has been set by the PCMC hardware. Bits[15:12] are read/write clear and bits[10:9] are read only.

Bits	Attribute	Description
15		**RESERVED**
14	R/WC	**SIGNALED SYSTEM ERROR (SSE):** When the PCMC asserts the SERR# signal, this bit is also set to 1. Software sets SSE to 0 by writing a 1 to this bit.
13	R/WC	**RECEIVED MASTER ABORT STATUS (RMAS):** When the PCMC terminates a Host-to-PCI transaction (PCMC is a PCI master), which is not a special cycle, with a master abort, this bit is set to 1. Software resets this bit to 0 by writing a 1 to it.
12	R/WC	**RECEIVED TARGET ABORT STATUS (RTAS):** When a PCMC-initiated PCI transaction is terminated with a target abort, RTAS is set to 1. The PCMC also asserts SERR# if the SERR# Target Abort bit in the ERRCMD Register is 1. Software resets RTAS to 0 by writing a 1 to it.
11		**RESERVED**
10:9	RO	**DEVSEL# TIMING (DEVT):** This 2-bit field indicates the timing of the DEVSEL# signal when the PCMC responds as a target. The PCI specification defines three allowable timings for assertion of DEVSEL#: 00 = fast, 01 = medium, and 10 = slow (DEVT = 11 is reserved). DEVT indicates the slowest time that a device asserts DEVSEL# for any bus command, except configuration read and write cycles. Note that these two bits determine the slowest time that the PCMC asserts DEVSEL#. However, the PCMC can also assert DEVSEL# in medium time. The PCMC asserts DEVSEL# in response to sampling MEMCS# asserted. The PCMC samples MEMCS# one and two clocks after FRAME# is asserted. If MEMCS# is asserted one PCI clock after FRAME# is asserted, then the PCMC responds with DEVSEL# in slow time.
8	R/WC	**DATA PARITY DETECTED (DPD):** This bit is set to 1 when all of the following conditions are met: 1). The PCMC asserted PERR# or sampled PERR# asserted. 2). The PCMC was the bus master for the operation in which the error occurred. 3). The PERRE bit in the Command Register is set to 1. Software resets DPD to 0 by writing a 1 to it.
7:0		**RESERVED**

3.2.6 RID—REVISION IDENTIFICATION REGISTER

Address Offset: 08h
Default Value: 03h for A-3 Stepping (82434LX)
01h for A-1 Stepping (82434LX)
10h for A-0 Stepping (82434NX)
11h for A-1 Stepping (82434NX)
Attribute: Read Only
Size: 8 bits

This register contains the revision number of the PCMC. These bits are read only and writes to this register have no effect. For the A-2 Stepping of the 82434LX, this value is 03h.

For the A-1 Stepping of the 82434NX, this value is 11h.

Bits	Description
7:0	**REVISION IDENTIFICATION NUMBER:** This is an 8-bit value that indicates the revision identification number for the PCMC.

2

3.2.7 RLPI—REGISTER-LEVEL PROGRAMMING INTERFACE REGISTER

Address Offset: 09h
Default Value: 00h
Attribute: Read Only
Size: 8 bits

This register defines the PCMC as having no defined register-level programming interface.

Bits	Description
7:0	**REGISTER-LEVEL PROGRAMMING INTERFACE (RLPI):** The value of 00h defines the PCMC as having no defined register-level programming interface.

3.2.8 SUBC—SUB-CLASS CODE REGISTER

Address Offset: 0Ah
Default Value: 00h
Attribute: Read Only
Size: 8 bits

This register defines the PCMC as a host bridge.

Bits	Description
7:0	**SUB-CLASS CODE (SCCD):** The value of this register is 00h defining the PCMC as host bridge.

3.2.9 BASEC—BASE CLASS CODE REGISTER

Address Offset: 0Bh
Default Value: 06h
Attribute: Read Only
Size: 8 bits

This register defines the PCMC as a bridge device.

Bits	Description
7:0	**BASE CLASS CODE (BCCD):** The value in this register is 06h defining the PCMC as bridge device.

3.2.10 MLT—MASTER LATENCY TIMER REGISTER

Address Offset: 0Dh
Default Value: 20h
Attribute: Read/Write
Size: 8 bits

MLT is an 8-bit register that controls the amount of time the PCMC, as a bus master, can burst data on the PCI Bus. MLT is used when the PCMC becomes the PCI Bus master and is cleared and suspended when the PCMC is not asserting FRAME#. When the PCMC asserts FRAME#, the counter is enabled and begins counting. If the PCMC finishes its transaction before the count expires, the MLT count is ignored. If the count expires before the transaction completes, the PCMC initiates a transaction termination as soon as its GNT# is removed. The number of clocks programmed in the MLT represents the guaranteed time slice (measured in PCI clocks) allotted to the PCMC, after which it must surrender the bus as soon as its GNT# is taken away. The number of clocks in the Master Latency Timer is the count value field multiplied by 16.

Bits	Description
7:4	**MASTER LATENCY TIMER COUNT VALUE:** If GNT# is negated after the burst cycle is initiated, the PCMC limits the duration of the burst cycle to the number of PCI Bus clocks specified by this field multiplied by 16.
3:0	**RESERVED**

3.2.11 BIST—BIST REGISTER

Address Offset: 0Fh
Default Value: 0h
Attribute: Read Only
Size: 8 bits

The BIST function is not supported by the PCMC. Writes to this register have no affect.

Bits	Attribute	Description
7	RO	**BIST SUPPORTED:** This read only bit is always set to 0, disabling the BIST function. Writes to this bit position have no affect.
6	RW	**START BIST:** This function is not supported and writes have no affect.
5:4		**RESERVED**
3:0	RO	**COMPLETION CODE:** This read only field always returns 0 when read and writes have no affect.

3.2.12 HCS—HOST CPU SELECTION REGISTER

Address Offset: 50h
Default Value: 82h (82434LX)
A2h (82434NX)
Access: Read/Write, Read Only
Size: 8 bits

The HCS Register is used to specify the Host CPU type and speed. This 8-bit register is also used to enable and disable the first level cache.

Bits	Access	Description
7:5	RO	**HOST CPU TYPE (HCT):** This field defines the Host CPU type. **82434LX** These bits are hardwired to 100 which selects the Pentium processor. All other combinations are reserved. **82434NX** In the 82434NX, these bits are reserved. Reads and writes to these bits have no effect.
4:3		**RESERVED**
2	R/W	**FIRST LEVEL CACHE ENABLE (FLCE):** FLCE enables and disables the first level cache. When FLCE = 1, the PCMC responds to CPU cycles with KEN# asserted for cacheable memory cycles. When FLCE = 0, KEN# is always negated. This prevents new cache line fills to either the first level or second level caches.
1:0	R/W	**HOST OPERATING FREQUENCY (HOF):** The DRAM refresh rate is adjusted according to the frequency selected by this field. For the 82434LX, only bit 0 is used and bit 1 is reserved. **82434LX** Bit 1 is reserved. If bit 0 is 1, the 82434LX supports a 66 MHz CPU. If bit 0 is 0, the 82434LX supports a 60 MHz CPU. **82434NX** These bits select the Host CPU frequency supported as follows: **Bits[1:0] Host CPU Frequency** 00 Reserved 01 50 MHz 10 60 MHz 11 66 MHz

2

3.2.13 DFC—DETURBO FREQUENCY CONTROL REGISTER

Address Offset: 51h
Default Value: 80h
Attribute: Read/Write
Size: 8 bits

Some software packages rely on the operating speed of the processor to time certain system events. To maintain backward compatibility with these software packages, the PCMC provides a mechanism to emulate a slower operating speed. This emulation is achieved with the PCMC's deturbo mode. The deturbo mode is enabled and disabled via the DM bit in the Turbo-Reset Control Register. When the deturbo mode is enabled, the PCMC periodically asserts AHOLD to slow down the effective speed of the CPU. The duty cycle of the AHOLD active period is controlled by the DFC Register.

Bits	Description
7:6	**DETURBO MODE FREQUENCY ADJUSTMENT VALUE:** This 8-bit value effectively defines the duty cycle of the AHOLD signal. DFC[7:6] are programmable and DFC[5:0] are 0. The value programmed into this register is compared against a free running 8-bit counter running at 1/8 the CPU clock. When the counter is greater than the value specified in this register, AHOLD is asserted. AHOLD is negated when the counter value is equal to or smaller than the contents of this register. AHOLD is negated when the counter rolls over to 00h. The deturbo emulation speed is directly proportional to the value in this register. Smaller values in this register yield slower deturbo emulation speed. The value of 00h is reserved.
5:0	**RESERVED**

3.2.14 SCC—SECONDARY CACHE CONTROL REGISTER

Address Offset: 52h
Default Value: SSS01R10 (82434LX)
SSS01010 (82434NX)
(S = Strapping option)
Attribute: Read/Write
Size: 8 bits

This 8-bit register defines the secondary cache operations. The SCC Register enables and disables the second level cache, adjusts cache size, selects the cache write policy, and defines the cache SRAM type. After hard reset, SCC[7:5] contain the opposite of the signal levels sampled on the Host address lines A[31:29].

Bits	Description
7:6	**SECONDARY CACHE SIZE (SCS):** This field defines the size of the second level cache. The values sampled on the A[31:30] lines at the rising edge of the PWROK signal are inverted and stored in this field. **Bits[7:6] — Secondary Cache Size** 00 — Cache not populated 01 — Reserved 10 — 256-KBytes 11 — 512-KBytes

Bits	Description
5	**SRAM TYPE (SRAMT):** This bit selects between standard SRAMs or burst SRAMS to implement the second level cache. When SRAMT = 0, standard SRAMs are selected. When SRAMT = 1, burst SRAMs are selected. This bit reflects the signal level on the A29 pin at the rising edge of the PWROK signal. This value can be overwritten with subsequent writes to the SCC Register.
4	**82434LX: SECONDARY CACHE ALLOCATION (SCA):** SCA controls when the PCMC performs line fills in the second level cache. When SCA is set to 0, only CPU reads of cacheable main memory with CACHE# asserted are cached in the second level cache. When SCA is set to 1, all CPU reads of cacheable main memory are cached in the second level cache.
3	**CACHE BYTE CONTROL (CBC):** When programmed for asynchronous SRAMs, this bit defines whether the cache uses individual write enables per byte or has a single write enable and byte select lines per byte. When CBC is set to 1, write enable control is used. When CBC is set to 0, byte select control is used.
2	**82434LX: RESERVED** **82434NX: SRAM CONNECTIVITY (SRAMC):** This bit enables different connectivities for the second level cache. When SRAMC is set to 0, the second level cache is in 82434LX compatible mode and all connections between the PCMC and second level cache SRAMs are the same as the 82434LX. When asynchronous SRAMs are used, setting this bit to 1 enables the CCS[1:0]# functionality. CCS[1:0]# are used with asynchronous SRAMs to de-select the SRAMs, placing them in a low power standby mode. When the CPU runs a halt or stop grant special cycle, the 82434NX negates CCS[1:0]#, placing the second level cache in a power saving mode. The PCMC then asserts CCS[1:0]# (activating the SRAMs) when the CPU asserts ADS#. When using burst SRAMs, setting this bit to 1 enables the CCS1# functionality and indicates to the PCMC that no external address latch is present.
1	**82434LX: SECONDARY CACHE WRITE POLICY (SCWP):** SCWP selects between write-back and write-through cache policies for the second level cache. When SCWP = 0 and the second level cache is enabled (bit 0 = 1), the second level cache is configured for write-through mode. When SCWP = 1 and the second level cache is enabled (bit 0 = 1), the second level cache is configured for write-back mode. **82434NX: RESERVED:** Secondary cache write-through mode is not supported. The secondary cache is always in write-back mode and this bit has no affect. SCWP can be set to 0, however, the 82434NX will still operate the secondary cache in write-back mode.
0	**SECONDARY CACHE ENABLE (SCE):** SCE enables and disables the secondary cache. When SCE = 1, the secondary cache is enabled. When SCE = 0, the secondary cache is disabled. When the secondary cache is disabled, the PCMC forwards all main memory cycles to the DRAM interface. Note that setting this bit to 0 does not affect existing valid cache lines. If a cache line contains modified data, the data is not written back to memory. Valid lines in the cache remain valid. When the secondary cache is disabled, the CWE[7:0]# lines remain negated. COE[1:0]# may still toggle. When system software disables secondary caching through this register during run-time, the software should first flush the second level cache. This process is accomplished by first disabling first level caching via the PCE bit in the HCS Register. This prevents the KEN# signal from being asserted, which disables any further line fills. At this point, software executes the WBINVD instruction to flush the caches. When the instruction completes, bit 0 of this register can be reset to 0, disabling the secondary cache. The first level cache can then be enabled by writing the PCE bit in the HCS Register.

3.2.15 HBC—HOST READ/WRITE BUFFER CONTROL

Address Offset: 53h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

The HBC Register enables and disables Host-to-main memory and Host-to-PCI posting of write cycles. When posting is enabled, the write buffers in the LBX devices post the data that is destined for either main memory or PCI. This register also permits a CPU-to-main memory read cycle to be performed before any pending posted write data is written to memory.

Bits	Description
7:4	**RESERVED**
3	**READ-AROUND-WRITE ENABLE (RAWCM):** If enabled, the PCMC, during a CPU read cycle to memory where posted write cycles are pending, internally snoops the write buffers. If the address of the read differs from the posted write addresses, the PCMC initiates the memory read cycle ahead of the pending posted memory write. When RAWCM = 0, the pending posted write is written to memory before the memory read is performed. When RAWCM = 1, the PCMC initiates the memory read ahead of the pending posted memory writes.
2	**RESERVED**
1	**HOST-TO-PCI POSTING ENABLE (HPPE):** This bit enables/disables the posting of Host-to-PCI write data in the LBX posting buffers. When HPPE = 1, up to 4 Dwords of data can be posted to PCI. HPPE = 0 is reserved. Buffering is disabled and each CPU write does not complete until the PCI transaction completes (TRDY# is asserted).
0	**82434LX: HOST-TO-MEMORY POSTING ENABLE (HMPE):** This bit enables/disables the posting of Host-to-main memory write data in the LBX buffers. When HMPE = 1, the CPU can post a single write or a burst write (4 Qwords). The CPU burst write completes at 4-1-1-1 when the second level cache is in write-back mode and at 3-1-1-1 when the second level cache is either disabled or in write-through mode. When HMPE = 0, Host-to-main memory posting is disabled and the CPU write cycles do not complete until the data is written to memory. **82434NX: RESERVED:** For the 82434NX, posting is always enabled and this bit has no affect. The CPU can post a single write or burst write (4 Qwords). HMPE can be set to 0, however, the 82434NX will still allow posting of CPU-to-main memory writes.

3.2.16 PBC—PCI READ/WRITE BUFFER CONTROL REGISTER

Address Offset: 54h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

The PBC Register enables and disables PCI-to-main memory write posting and permits single CPU-to-PCI writes to be assembled into PCI burst cycles.

Bits	Description
7:3	**RESERVED**
2	**LBXs CONNECTED TO TRDY#:** The TRDY# pin on the LBXs can be connected either to the PCI TRDY# signal or to ground. The cycle time for CPU-to-PCI writes is improved if TRDY# is connected to the LBXs. Since there are two LBXs used in a system, connecting this signal to the LBXs increases the electrical loading of TRDY# by two loads. When the LBXs are externally hard-wired to TRDY#, this bit should be set to 1. Note that this should be done prior to the first Host-to-PCI write or data corruption will occur. Setting this bit to 1 enables the capability of CPU-to-PCI writes at 2-1-1-1 . . . (PCI clocks). When this bit is 0, the LBXs are not connected to TRDY# and CPU-to-PCI writes are completed at 2-2-2-2 . . . timing.
1	**PCI BURST WRITE ENABLE (PBWE):** This bit enables and disables PCI Burst memory write cycles for back-to-back sequential CPU memory write cycles to PCI. When PBWE is set to 1, PCI burst writes are enabled. When PBWE is reset to 0, PCI burst writes are disabled and each single CPU write to PCI invokes a single PCI write cycle (each cycle has an associated FRAME# sequence).
0	**PCI-TO-MEMORY POSTING ENABLE (PMPE):** This bit enables and disables posting of PCI-to-memory write cycles. The posting occurs in a pair of four Dword-deep buffers in the LBXs. When PMPE is set to 1, these buffers are used to post PCI-to-main memory write data. When PMPE is reset to 0, PCI write transactions to main memory are limited to single transfers. The PCMC asserts STOP# with the first TRDY# to disconnect the PCI Master.

3.2.17 DRAMC—DRAM CONTROL REGISTER

Address Offset: 57h
Default Value: 31h
Attribute: Read/Write
Size: 8 bits

This 8-bit register controls main memory DRAM operating modes and features.

Bits	Description
7:6	**82434LX: RESERVED** **82434NX: DRAM BURST TIMING (DBT):** The DRAM interface can be configured for 3 different burst timings. The CAS# pulse width for X-3-3-3 timing is one clock shorter than the CAS# pulse width for X-4-4-4 timing. **Bits[7:6] Burst Timing** 0 0 X-4-4-4 Read/Write timing (default) 0 1 X-4-4-4 Read, X-3-3-3 Write timing 1 0 Reserved 1 1 X-3-3-3 Read/Write timing
5	**PARITY ERROR MASK (PERRM):** When PERRM = 1, parity errors generated during DRAM read cycles initiated by either the CPU request or a PCI Master are masked. This bit affects bits 0 and 1 of the Error Command Register and the ability of the PCMC to respond to PCHK# and assert SERR# when a DRAM parity error occurs. When PERRM is reset to 0, parity errors are not masked.
4	**0-ACTIVE RAS# MODE:** This bit determines if the DRAM page for a particular row remains open (i.e. RAS# remains asserted after a DRAM cycle) enabling the possibility that the next DRAM access may be either a page hit, a page miss, or a row miss. The DRAM interface is then in 1-active RAS# mode. If this bit is reset to 0, RAS# remains asserted after a DRAM cycle. If this bit is set to 1, RAS# is negated after every DRAM cycle, resulting in a row miss for every DRAM cycle. The DRAM interface is then in 0-active RAS# mode.
3	**SMRAM ENABLE (SMRE):** When SMRE = 1, CPU accesses to SMM space are qualified with the SMIACT# pin of the CPU. The location of this space is determined by the SBS field of the SMRAM Register. Read and write cycles to SMM space function normally if SMIACT# is asserted. If SMIACT# is negated when accessing this space, the cycle is forwarded to PCI. When SMRE = 0, accesses to SMM space are treated normally and SMIACT# has no effect. SMRE must be set to 1 to enable the use of the SMRAM Register at configuration space offset 72h.
2	**BURST OF FOUR REFRESH (BFR):** When BFR is set to 1, refreshes are performed in sets of four, at a frequency ¼ of the normal refresh rate. The PCMC defers refreshes to idle times, if possible. When BFR is reset to 0, single refreshes occur at 15.6 μs refresh rate.
1	**82434LX: REFRESH TYPE (RT):** When RT = 1, the PCMC uses CAS#-before-RAS# timing to refresh the DRAM array. For this refresh type, the PCMC does not supply refresh addresses. When RT = 0, RAS# Only refresh is used and the PCMC drives refresh addresses on the MA[10:0] lines. RAS# only refresh can be used with any type of second level cache configuration (i.e., no second level cache is present, or either a burst SRAM or standard SRAM second level cache is implemented). CAS#-before-RAS# refresh should not be used when a standard SRAM second level cache is implemented. **82434NX: REFRESH TYPE (RT):** In addition to above, when RT = 0, RAS# only refresh is used and the PCMC drives refresh addresses on the MA[11:0] lines. Also, CAS#-before-RAS# refresh can be used with a standrad SRAM second level cache.
0	**REFRESH ENABLE (RE):** When RE is set to 1, the main memory array is refreshed as configured via bits 1 and 2 of this register. When RE is reset to 0, DRAM refresh is disabled. Note that disabling refresh results in the loss of DRAM data.

3.2.18 DRAMT—DRAM TIMING REGISTER

Address Offset: 58h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

For the 82434LX, this register controls the leadoff latency for CPU DRAM accesses.

For the 82434NX, this register provides additional control over DRAM timings. One additional wait-state can be independently added before the assertion of RAS#, the assertion of the first CAS#, or both. This is to allow more flexibility in the layout of the motherboard and in the selection of DRAM speed grades.

Bits	Description
7:2	**RESERVED**
1	**82434LX: RESERVED** **82434NX: RAS# WAIT-STATE (RWS):** When RWS = 1, one additional wait state will be inserted before RAS# is asserted for row misses or page misses in 1-Active RAS mode and all cycles in 0-Active RAS mode. This provides additional MA[11:0] setup time to RAS# assertion.
0	**CAS# WAIT-STATE (CWS):** When CWS = 1, one additional wait state will be inserted before the first assertion of CAS# within a burst cycle. There is no additional delay between CAS# assertions. This provides additional MA[11:0] setup time to CAS# assertion. The CWS bit is typically reset to 0 for 60 MHz operation and set to 1 for 66 MHz operation.

2

3.2.19 PAM—PROGRAMMABLE ATTRIBUTE MAP REGISTERS (PAM[6:0])

Address Offset: 59–5Fh
Default Value: PAM0 = 0Fh, PAM[1:6] = 00h
Attribute: Read/Write

The PCMC allows programmable memory and cacheability attributes on 14 memory segments of various sizes in the 512 KByte–1 MByte address range. Seven Programmable Attribute Map (PAM) Registers are used to support these features. Three bits are used to specify cacheability and memory attributes for each memory segment. These attributes are:

RE: **Read Enable.** When RE = 1, the CPU read accesses to the corresponding memory segment are directed to main memory. Conversely, when RE = 0, the CPU read accesses are directed to PCI.

WE: **Write Enable.** When WE = 1, the CPU write accesses to the corresponding memory segment are directed to main memory. Conversely, when WE = 0, the CPU write accesses are directed to PCI.

CE: **Cache Enable.** When CE = 1, the corresponding memory segment is cacheable. CE must not be set to 1 when RE is reset to 0 for any particular memory segment. When CE = 1 and WE = 0, the corresponding memory segment is cached in the first and second level caches only on CPU coded read cycles.

The RE and WE attributes permit a memory segment to be Read Only, Write Only, Read/Write, or disabled. For example, if a memory segment has RE = 1 and WE = 0, the segment is Read Only. The characteristics for memory segments with these read/write attributes are described in Table 2.

Table 2. Attribute Definition

Read/Write Attribute	Definition
Read Only	Read cycles: CPU cycles are serviced by the DRAM in a normal manner. Write cycles: CPU initiated write cycles are ignored by the DRAM interface as well as the cache. Instead, the cycles are passed to PCI for termination. Areas marked as Read Only are cacheable for Code accesses only. These regions may be cached in the second level cache, however as noted above, writes are forwarded to PCI, effectively write protecting the data.
Write Only	Read cycles: All read cycles are ignored by the DRAM interface as well as the second level cache. CPU-initiated read cycles are passed onto PCI for termination. The write only state can be used while copying the contents of a ROM, accessible on PCI, to main memory for shadowing, as in the case of BIOS shadowing. Write cycles: CPU write cycles are serviced by the DRAM and cache in a normal manner.
Read/Write	This is the normal operating mode of main memory. Both read and write cycles from the CPU and PCI are serviced by the DRAM and cache interface.
Disabled	All read and write cycles to this area are ignored by the DRAM and cache interface. These cycles are forwarded to PCI for termination.

Each PAM Register controls two regions, typically 16-KByte in size. Each of these regions have a 4-bit field. The four bits that control each region have the same encoding and are defined in Table 3.

Table 3. Attribute Bit Assignment

Bits[7,3] Reserved	Bits[6,2] Cache Enable	Bits[5,1] Write Enable	Bits[4,0] Read Enable	Description
x	x	0	0	DRAM Disabled, Accesses Directed to PCI
x	0	0	1	Read Only, DRAM Write Protected, Non-Cacheable
x	1	0	1	Read Only, DRAM Write Protected, Cacheable for Code Accesses Only
x	0	1	0	Write Only
x	0	1	1	Read/Write, Non-Cacheable
x	1	1	1	Read/Write, Cacheable

NOTE:
To enable PCI master access to the DRAM address space from C0000h to FFFFFh the MEMCS# configuration registers of the ISA or EISA bridge must be properly configured. These registers must correspond to the PAM Registers in the PCMC.

As an example, consider a BIOS that is implemented on the expansion bus. During the initialization process the BIOS can be shadowed in main memory to increase the system performance. When a BIOS is shadowed in main memory, it should be copied to the same address location. To shadow the BIOS, the attributes for that address range should be set to write only. The BIOS is shadowed by first doing a read of that address. This read is forwarded to the expansion bus. The CPU then does a write of the same address, which is directed to main memory. After the BIOS is shadowed, the attributes for that memory area are set to read only so that all writes are forwarded to the expansion bus.

Table 4. PAM Registers and Associated Memory Segments

PAM Reg	Attribute Bits		Memory Segment		Comments		Offset
PAM0[3:0]	R	CE	WE	RE	080000h–09FFFFh	512K–640K	59h
PAM0[7:4]	R	CE	WE	RE	0F0000h–0FFFFFh	BIOS Area	59h
PAM1[3:0]	R	CE	WE	RE	0C0000h–0C3FFFh	ISA Add-on BIOS	5Ah
PAM1[7:4]	R	CE	WE	RE	0C4000h–0C7FFFh	ISA Add-on BIOS	5Ah
PAM2[3:0]	R	CE	WE	RE	0C8000h–0CBFFFh	ISA Add-on BIOS	5Bh
PAM2[7:4]	R	CE	WE	RE	0CC000h–0CFFFFh	ISA Add-on BIOS	5Bh
PAM3[3:0]	R	CE	WE	RE	0D0000h–0D3FFFh	ISA Add-on BIOS	5Ch
PAM3[7:4]	R	CE	WE	RE	0D4000h–0D7FFFh	ISA Add-on BIOS	5Ch
PAM4[3:0]	R	CE	WE	RE	0D8000h–0DBFFFh	ISA Add-on BIOS	5Dh
PAM4[7:4]	R	CE	WE	RE	0DC000h–0DFFFFh	ISA Add-on BIOS	5Dh
PAM5[3:0]	R	CE	WE	RE	0E0000h–0E3FFFh	BIOS Extension	5Eh
PAM5[7:4]	R	CE	WE	RE	0E4000h–0E7FFFh	BIOS Extension	5Eh
PAM6[3:0]	R	CE	WE	RE	0E8000h–0EBFFFh	BIOS Extension	5Fh
PAM6[7:4]	R	CE	WE	RE	0EC000h–0EFFFFh	BIOS Extension	5Fh

DOS Application Area (00000h-9FFFh)

The 640-KByte DOS application area is split into two regions. The first region is 0–512-KByte and the second region is 512–640 KByte. Read, write, and cacheability attributes are always enabled and are not programmable for the 0–512 KByte region.

Video Buffer Area (A0000h-BFFFFh)

This 128-KByte area is not controlled by attribute bits. CPU-initiated cycles in this region are always forwarded to PCI for termination. This area is not cacheable.

Expansion Area (C0000h-DFFFFh)

This 128-KByte area is divided into eight 16-KByte segments. Each segment can be assigned one of four Read/Write states: read-only, write-only, read/write, or disabled Memory that is disabled is not remapped. Cacheability status can also be specified for each segment.

Extended System BIOS Area (E0000h-EFFFFh)

This 64-KByte area is divided into four 16-KByte segments. Each segment can be assigned independent cacheability, read, and write attributes. Memory segments that are disabled are not remapped elsewhere.

System BIOS Area (F0000h-FFFFFh)

This area is a single 64-KByte segment. This segment can be assigned cacheability, read, and write attributes. When disabled, this segment is not remapped.

Extended Memory Area (100000h-FFFFFFFFh)

The extended memory area can be split into several parts:

- Flash BIOS area from 4 GByte to 4 GByte–512-KByte (aliased on ISA at 16 MBytes–15.5 MBytes)
- DRAM Memory from 1 MByte to a maximum of 192 MBytes
- PCI Memory space from the top of DRAM to 4 GByte – 512-KByte
- Memory Space Gap between the range of 1 MByte up to 15.5 MBytes
- Frame Buffer Range mapped into PCI Memory Space or the Memory Space Gap.

On power-up or reset the CPU vectors to the Flash BIOS area, mapped in the range of 4 GByte to 4 GByte – 512-KByte. This area is physically mapped on the expansion bus. Since these addresses are in the upper 4 GByte range, the request is directed to PCI.

The DRAM memory space can occupy extended memory from a minimum of 2 MBytes up to 192 MBytes. This memory is cacheable.

The address space on PCI between the Flash BIOS (4 GByte to 4 GByte – 512 KByte) and the top of DRAM (including any remapped memory) may be occupied by PCI memory. This memory space is not cacheable.

3.2.20 DRB—DRAM ROW BOUNDARY REGISTERS

Address Offset:	60–65h (82434LX)
	60–67h (82434NX)
Default Value:	02h
Attribute:	Read/Write
Size:	8 bits

Note the address offset for each DRB Register is DRB0=60h, DRB1=61h, DRB2=62h, DRB3=63h, DRB4=64h, DRB5=65h, DRB6=66h, and DRB7=67h.

3.2.20.1 82434LX Description

The PCMC supports 6 rows of DRAM. Each row is 64 bits wide. The DRAM Row Boundary Registers define upper and lower addresses for each DRAM row. Contents of these 8-bit registers represent the boundary addresses in MBytes.

DRB0 = Total amount of memory in row 0 (in MBytes)
DRB1 = Total amount of memory in row 0 + row 1 (in MBytes)
DRB2 = Total amount of memory in row 0 + row 1 + row 2 (in MBytes)
DRB3 = Total amount of memory in row 0 + row 1 + row 2 + row 3 (in MBytes)
DRB4 = Total amount of memory in row 0 + row 1 + row 2 + row 3 + row 4 (in MBytes)
DRB5 = Total amount of memory in row 0 + row 1 + row 2 + row 3 + row 4 + row 5 (in MBytes)

The DRAM array can be configured with 256K x 36, 1M x 36 and 4M x 36 SIMMs. Each register defines an address range that will cause a particular RAS# line to be asserted (e.g. if the first DRAM row is 2 MBytes in size then accesses within the 0 MByte–2 MBytes range will cause RAS0# to be asserted). The DRAM Row

Boundary (DRB) Registers are programmed with an 8-bit upper address limit value. This upper address limit is compared to A[27:20] of the Host address bus, for each row, to determine if DRAM is being targeted. Since this value is 8 bits and the resolution is 1 MByte, the total bits compared span a 256 MByte space. However, only 192 MBytes of main memory is supported.

Bits	Description
7:0	**ROW BOUNDARY ADDRESS IN MBYTES:** This 8-bit value is compared against address lines A[27:20] to determine the upper address limit of a particular row, i.e. DRB − previous DRB = row size.

Row Boundary Address in MBytes

These 8-bit values represent the upper address limits of the six rows (i.e., this row - previous row = row size). Unpopulated rows have a value equal to the previous row (row size = 0). The value programmed into DRB5 reflects the maximum amount of DRAM in the system. Memory remapped at the top of DRAM, as a result of setting the Memory Space Gap Register, is not reflected in the DRB Registers. The top of memory is always determined by the value written into DRB5 added to the memory space gap size (if enabled).

As an example of a general purpose configuration where 3 physical rows are configured for either single-sided or double-sided SIMMs, the memory array would be configured like the one shown in Figure 8. In this configuration, the PCMC drives two RAS# signals directly to the SIMM rows. If single-sided SIMMs are populated, the even RAS# signal is used and the odd RAS# is not connected. If double-sided SIMMs are used, both RAS# signals are used.

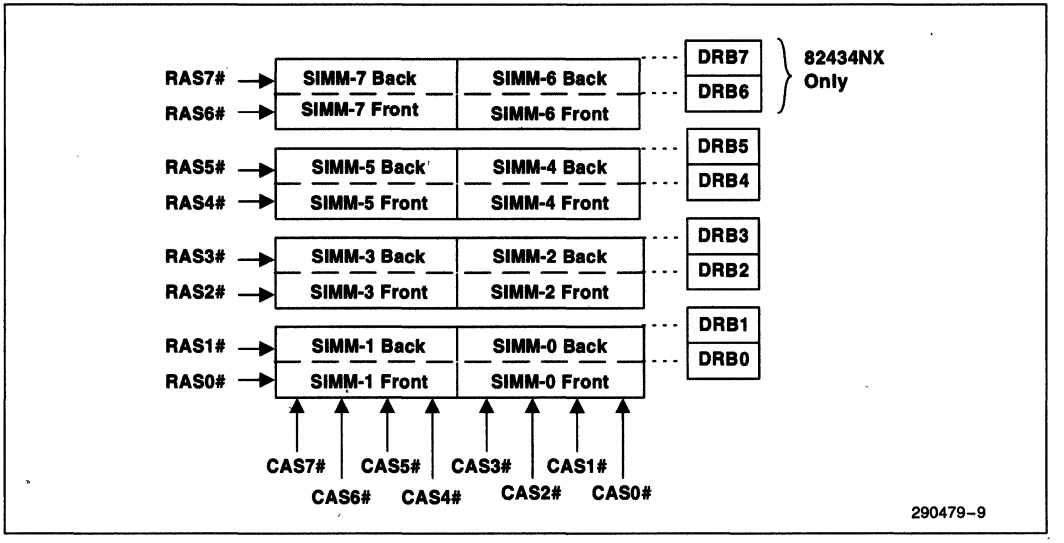


Figure 8. SIMMs and Corresponding DRB Registers

The following 2 examples describe how the DRB Registers are programmed for cases of single-sided and double-sided SIMMs on a motherboard having a total of 6 SIMM sockets.

Example #1

The memory array is populated with six single-sided 256-KByte x 36 SIMMs. Two SIMMs are required for each populated row making each populated row 2 MBytes in size. Filling the array yields 6 MBytes total DRAM. The DRB Registers are programmed as follows:

DRB0 = 02h	populated
DRB1 = 02h	empty row, not double-sided SIMMs
DRB2 = 04h	populated
DRB3 = 04h	empty row, not double-sided SIMMs
DRB4 = 06h	populated
DRB5 = 06h	empty row, not double-sided SIMMs, maximum memory = 6 MBytes.

Example #2

As an another example, if the first four SIMM sockets are populated with 2 MBytes x 36 double-sided SIMMs and the last two SIMM sockets are populated with 4 MBytes x 36 single-sided SIMMs then filling the array yields 64 MBytes total DRAM. The DRB Registers are programmed as follows:

DRB0 = 08h	populated with 8 MBytes, ½ of the double-sided SIMMs
DRB1 = 10h	the other 8 MBytes of the double-sided SIMMs
DRB2 = 18h	populated with 8 MBytes, ½ of the double-sided SIMMs
DRB3 = 20h	the other 8 MBytes of the double-sided SIMMs
DRB4 = 40h	populated with 32 MBytes
DRB5 = 40h	empty row, not double-sided SIMMs, maximum memory = 64 MBytes.

3.2.20.2 82434NX Description

The PCMC supports 8 rows of DRAM. Each row is 64 bits wide. The DRAM Row Boundary Registers define upper and lower addresses for each DRAM row. Contents of these 8-bit registers are concatenated with the associated nibble of the DRBE Register to form 12 bit quantities that represent the row boundary addresses in MBytes.

DRBE[3:0] || DRB0 =	Total amount of memory in row 0 (in MBytes)
DRBE[7:4] || DRB1 =	Total amount of memory in row 0 + row 1 (in MBytes)
DRBE[11:8] || DRB2 =	Total amount of memory in row 0 + row 1 + row 2 (in MBytes)
DRBE[15:12] || DRB3 =	Total amount of memory in row 0 + row 1 + row 2 + row 3 (in MBytes)
DRBE[19:16] || DRB4 =	Total amount of memory in row 0 + row 1 + row 2 + row 3 + row 4 (in MBytes)
DRBE[23:20] || DRB5 =	Total amount of memory in row 0 + row 1 + row 2 + row 3 + row 4 + row 5 (in Bytes)
DRBE[27:24] || DRB6 =	Total amount of memory in row 0 + row 1 + row 2 + row 3 + row 4 + row 5 + row 6 (in MBytes)
DRBE[31:28] || DRB7 =	Total amount of memory in row 0 + row 1 + row 2 + row 3 + row 4 + row 5 + row 6 + row 7 (in MBytes)

The DRAM array can be configured with 256K x 36, 1M x 36, 4M x 36, and 16M x 36 SIMMs. Each register defines an address range that will cause a particular RAS# line to be asserted (e.g. if the first DRAM row is 2 MBytes in size then accesses within the 0 to 2 MBytes range will cause RAS0# to be asserted). The DRAM Row Boundary (DRB) Registers are programmed with an 8-bit upper address limit value. The DRBE Register extends the programming model of this mechanism to 12 bits, however only 10 bits are implemented at this time. This upper address limit is compared to A[29:20] of the Host address bus, for each row, to determine if DRAM is being targeted. Since this value is 10 bits and the resolution is 1 MByte, the total bits compared span a 1 GByte space. However, other resource limits in the PCMC cap the total usable DRAM space at 512 MBytes.

Bits	Description
7:0	**ROW BOUNDARY ADDRESS IN MBYTES:** This 8-bit value is concatenated with a nibble from the DRBE Register and then compared against address lines A[29:20] to determine the upper address limit of a particular row (i.e. DRB − previous DRB = row size).

Row Boundary Address in MBytes

These 10-bit values represent the upper address limits of the 8 rows (i.e., this row - previous row = row size). Unpopulated rows have a value equal to the previous row (row size = 0). The value programmed into DRBE[31:28] || DRB7 reflects the maximum amount of DRAM in the system. Memory remapped at the top of DRAM, as a result of setting the Memory Space Gap Register, is not reflected in the DRB Registers. The top of memory is determined by the value written into DRBE[31:28] || DRB7 added to the memory space gap size (if enabled). If DRBE[31:28] || DRB7 plus the memory space gap is greater than 512 MBytes then 512 MBytes of DRAM are available.

The following 2 examples describe how the DRB Registers are programmed for cases of single-sided and double-sided SIMMs on a motherboard having a total of 8 SIMM sockets.

Example #1

The memory array is populated with eight single-sided 256-KByte x 36 SIMMs. Two SIMMs are required for each populated row making each populated row 2 MBytes in size. Filling the array yields 8 MBytes total DRAM. The DRB Registers are programmed as follows:

DRBE[3:0] = 0h	DRB0 = 02h	populated
DRBE[7:4] = 0h	DRB1 = 02h	empty row, not double-sided SIMMs
DRBE[11:8] = 0h	DRB2 = 04h	populated
DRBE[15:12] = 0h	DRB3 = 04h	empty row, not double-sided SIMMs
DRBE[19:16] = 0h	DRB4 = 06h	populated
DRBE[23:20] = 0h	DRB5 = 06h	empty row, not double-sided SIMMs
DRBE[27:24] = 0h	DRB6 = 08h	populated
DRBE[31:28] = 0h	DRB7 = 08h	empty row, not double-sided SIMMs, max memory = 8 MBytes.

Example #2

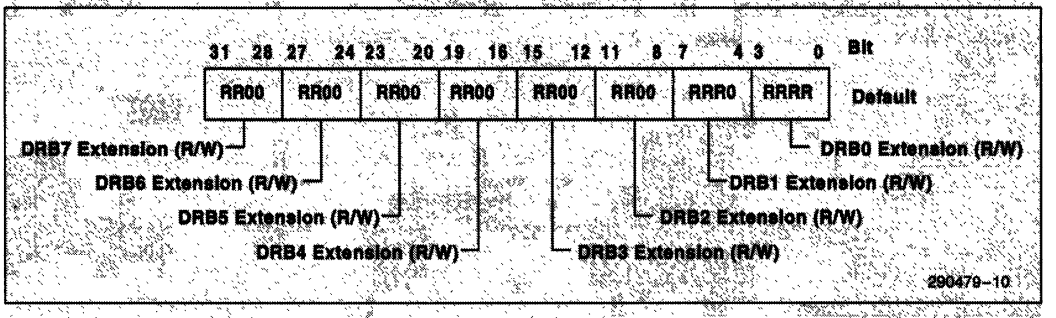
As an another example, if the first four SIMM sockets are populated with 2 MByte x 36 double-sided SIMMs and the last four SIMM sockets are populated with 16 MByte x 36 single-sided SIMMs then filling the array yields 288 MBytes total DRAM. The DRB Registers are programmed as follows:

DRBE[3:0] = 0h	DRB0 = 08h	populated with 8 MBytes, ½ of double-sided SIMMs
DRBE[7:4] = 0h	DRB1 = 10h	the other 8 MBytes of the double-sided SIMMs
DRBE[11:8] = 0h	DRB2 = 18h	populated with 8 MBytes, ½ of double-sided SIMMs
DRBE[15:12] = 0h	DRB3 = 20h	the other 8 MBytes of the double-sided SIMMs
DRBE[19:16] = 0h	DRB4 = A0h	populated with 128 MBytes
DRBE[23:20] = 0h	DRB5 = A0h	empty row, not double-sided SIMMs
DRBE[27:24] = 1h	DRB6 = 20h	populated with 128 MBytes
DRBE[31:28] = 1h	DRB7 = 20h	empty row, not double-sided SIMMs, max memory = 288 MBytes.

3.2.21 DRBE—DRAM ROW BOUNDARY EXTENSION REGISTER

Address Offset: 68-6Bh
Default Value: 0000h
Attribute: Read/Write
Size: 32 bits

The DRBE Register is not implemented in the 82434LX. This register contains an extension for each of the DRAM Row Boundary (DRB) Registers. Each nibble of the DRBE Register is concatenated with a DRB Register (see DRB Register section for details on the use of the DRB and DRBE Registers).



Bits	Description
31:0	**EXTENSIONS FOR DRB0 THROUGH DRB7:** Each nibble corresponds to a DRB. The nibble of the DRBE and its corresponding DRB are concatenated and used to indicate the boundaries between rows of DRAM.

3.2.22 ERRCMD—ERROR COMMAND REGISTER

Address Offset: 70h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

The Error Command Register controls the PCMC responses to various system errors. Bit 6 of the PCICMD Register is the master enable for bit 3 of this register. Bit 6 of the PCICMD Register must be set to 1 to enable the error reporting function defined by bit 3 of this register. Bits 6 and 8 of the PCICMD Register are the master enables for bits 7, 6, 5, 4, and 1 of this register. Both bits 6 and 8 of the PCICMD Register must be set to 1 to enable the error reporting functions defined by bits 7, 6, 5, 4, and 1 of this register.

Bits	Description
7	**SERR# ON RECEIVED TARGET ABORT:** When this bit is set to 1 (and bit 8 of the PCICMD Register is 1), the PCMC asserts SERR# upon receiving a target abort. When this bit is set to 0, the PCMC is disabled from asserting SERR# upon receiving a target abort.
6	**SERR# ON TRANSMITTED PCI DATA PARITY ERROR:** When this bit is set to 1 (and bits 6 and 8 of the PCICMD Register are both 1), the PCMC asserts SERR# when it detects a data parity error as a result of a CPU-to-PCI write (PERR# detected asserted). When this bit is set to 0, the PCMC is disabled from asserting SERR# when data parity errors are detected via PERR#.
5	**82434LX: RESERVED** **82434NX: SERR# ON RECEIVED PCI DATA PARITY ERROR:** When this bit is set to 1 (and bits 6 and 8 of the PCICMD Register are both 1), the PCMC asserts SERR# when it detects a data parity error as a result of a CPU-to-PCI read (PAR incorrect with received data). In this case, the SERR# signal is asserted when parity errors are detected on PCI return data. When this bit is set to 0, the PCMC is disabled from asserting SERR# when data parity errors are detected during a CPU-to-PCI read.
4	**82434LX: RESERVED** **82434NX: SERR# ON PCI ADDRESS PARITY ERROR:** When this bit is set to 1 (and bits 6 and 8 of the PCICMD Register are both 1), the PCMC asserts SERR# when it detects an address parity error on PCI transactions. When this bit is set to 0, the PCMC is disabled from asserting SERR# when address parity errors are detected on PCI transactions.
3	**82434LX: RESERVED** **82434NX: PERR# ON RECEIVING A DATA PARITY ERROR:** This bit indicates whether the PERR# signal is implemented in the system. When this bit is set to 1 (and bit 6 of the PCICMD Register is 1), the PCMC asserts PERR# when it detects a data parity error (PAR incorrect with received data), either from a CPU-to-PCI read or a PCI master write to memory. When this bit is set to 0 (or bit 6 of the PCICMD Register is set to 0), the PERR# signal is not asserted by the PCMC.
2	**L2 CACHE PARITY ENABLE:** This bit indicates that the second level cache implements parity. When this bit is set to 1, bits 0 and 1 of this register control the checking of parity errors during CPU reads from the second level cache. If this bit is 0, parity is not checked when the CPU reads from the second level cache (PCHK# ignored) and neither bit 1 nor bit 0 apply.
1	**SERR# ON DRAM/L2 CACHE DATA PARITY ERROR ENABLE:** This bit enables/disables the SERR# signal for parity errors on reads from main memory or the second level cache. When this bit is set to 1 and bit 0 of this register is set to 1 (and bits 6 and 8 of the PCICMD Register are set to 1), SERR# is enabled upon a PCHK# assertion from the CPU when reading from main memory or the second level cache. The processor indicates that a parity error was received by asserting PCHK#. The PCMC then latches status information in the Error Status Register and asserts SERR#. When this bit is 0, SERR# is not asserted upon detecting a parity error. Bits[1:0] = 10 is a reserved combination. 0 = Disable assertion of SERR# upon detecting a DRAM/second level cache read parity error. 1 = Enable assertion of SERR# upon detecting a DRAM/second level cache read parity error.
0	**MCHK ON DRAM/L2 CACHE DATA PARITY ERROR ENABLE:** When this bit is set to 1, PEN# is asserted for data returned from main memory or the second level cache. The processor indicates that a parity error was received by asserting the PCHK# signal. In addition, the processor invokes a machine check exception, if enabled via the MCE bit in CR4 in the Pentium processor. The PCMC then latches status information in the Error Status register. When this bit is 0, PEN# is not asserted. Bits[1:0] = 10 is a reserved combination.

3.2.23 ERRSTS—ERROR STATUS REGISTER

Address Offset: 71h
Default Value: 00h
Attribute: Read/Write Clear
Size: 8 bits

The Error Status Register is an 8-bit register that reports the occurrence of PCI, second level cache, and DRAM parity errors. This register also reports the occurrence of a CPU shutdown cycle.

Bits	Description
7	**RESERVED**
6	**PCI TRANSMITTED DATA PARITY ERROR:** The PCMC sets this bit to a 1 when it detects a data parity error (PERR# asserted) as a result of a CPU-to-PCI write. Software resets this bit to 0 by writing a 1 to it.
5	**82434LX: RESERVED** **82434NX: PCI RECEIVED DATA PARITY ERROR:** The PCMC sets this bit to a 1 when it detects a data parity error (PAR incorrect with received data) as a result of a CPU-to-PCI read. Software resets this bit to 0 by writing a 1 to it.
4	**82434LX: RESERVED** **82434NX: PCI ADDRESS PARITY ERROR:** The PCMC sets this bit to a 1 when it detects an address parity error (PAR incorrect with received address and C/BE# lines) on a PCI master transaction. Software resets this bit to 0 by writing a 1 to it.
3	**MAIN MEMORY DATA PARITY ERROR:** The PCMC sets this bit to a 1 when it detects a parity error from the CPU PCHK# signal resulting from a CPU-to-main memory read. Software resets this bit to 0 by writing a 1 to it.
2	**L2 CACHE DATA PARITY ERROR:** The PCMC sets this bit to a 1 when it detects a parity error from the CPU PCHK# signal resulting from a CPU read access that hit in the second level cache. Software resets this bit to 0 by writing a 1 to it.
1	**RESERVED**
0	**SHUTDOWN CYCLE DETECTED:** The PCMC sets this bit to a 1 when it detects a shutdown special cycle on the Host Bus. Under this condition the PCMC drives a shutdown special cycle on PCI and asserts INIT. Software resets this bit to 0 by writing a 1 to it.

3.2.24 SMRS—SMRAM SPACE REGISTER

Address Offset: 72h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

The PCMC supports a 64-KByte SMRAM space that can be selected to reside at the top of main memory, segment A0000–AFFFFh or segment B0000–BFFFFh. The SMM space defined by this register is not cacheable. This register defines a mechanism that allows the CPU to execute code out of the SMM space at either A0000h or B0000h while accessing the frame buffer on PCI. The SMRAM Enable bit in the DRAM Control Register must be 1 to enable the features defined by this register. Register bits[5:3] apply only when segment A0000-AFFFFh or B0000-BFFFFh are selected.

Bits	Description
7:6	**RESERVED**
5	**OPEN SMRAM SPACE (OSS):** When OSS = 1, the CPU can access SMM space without being in SMM mode. That is, accesses to SMM space are permitted even with SMIACT# negated. This bit is intended to be used during POST to allow the CPU to initialize SMRAM space before the first SMI# interrupt is issued.
4	**CLOSE SMRAM SPACE (CSS):** When CSS = 1 and SMRAM is enabled, CPU code accesses to the SMM memory range are directed to SMM space in main memory and data accesses are forwarded to PCI. This bit allows the CPU to read and write the frame buffer on PCI while executing SMM code. When CSS = 0 and SMRAM is enabled, all accesses to the SMRAM memory range, both code and data, are directed to SMRAM (main memory).
3	**LOCK SMRAM SPACE (LSS):** When LSS = 1, this bit prevents the SMM space from being manually opened, effectively disabling bit 5 of this register. Only a power-on reset can set this bit to 0.
2:0	**SMM BASE SEGMENT (SBS):** This field defines the 64 KByte base segment where SMM space is located. The memory that is defined by this field is non-cacheable.

Bits[2:0]	SMRAM Location	Bits[2:0]	SMRAM Location
000	Top of main memory	100	Reserved
001	Reserved	101	Reserved
010	A0000–AFFFFh	110	Reserved
011	B0000–BFFFFh	111	Reserved

3.2.25 MSG—MEMORY SPACE GAP REGISTER

Address Offset: 78-79h
Default Value: 00h
Attribute: Read/Write
Size: 16 bits

The Memory Space Gap Register defines the starting address and size of a gap in main memory. This register accommodates ISA devices that have their memory mapped into the 1 MByte–15.5 MByte range (e.g., an ISA LAN card or an ISA frame buffer). The Memory Space Gap Register defines a hole in main memory that transfers the cycles in this address space to the PCI Bus instead of main memory. This area is not cacheable.

The memory space gap starting address must be a multiple of the memory space gap size. For example, a 2 MByte gap must start at 2, 4, 6, 8, 10, 12, or 14 MBytes.

NOTE:
Memory that is disabled by the gap created by this register is remapped to the top of memory. This remapped memory is accessible, except in the case where this would cause the top of main memory to exceed 192 MBytes (or 512 MBytes for the 82434NX).

Bits	Description
15	**MEMORY SPACE GAP ENABLE (MSGE):** MSGE enables and disables the memory space gap. When MSGE is set to 1, the CPU accesses to the address range defined by this register are forwarded to PCI bus. The size of the gap created in main memory causes a corresponding amount of DRAM to be remapped at the top of main memory (top specified by DRB Registers). If the Frame Buffer Range is programmed below 16 MBytes and within main memory space, the MSG register must include the Frame Buffer Range. When MSGE is reset to 0, the memory space gap is disabled.
14:12	**MEMORY SPACE GAP SIZE (MSGS):** This 3 bit field defines the size of the memory space gap. If the Frame Buffer Range is programmed below 16 MBytes and within main memory space, this register must include the frame buffer range. The amount of main memory specified by these bits is remapped to the top of main memory. **Bit[14:12] Memory Gap Size** 000 1 MByte 001 2 MBytes 011 4 MBytes 111 8 MBytes **NOTE:** All other combinations are reserved.
11:8	**RESERVED**
7:4	**MEMORY SPACE GAP STARTING ADDRESS (MSGSA):** These 4 bits define the starting address of the memory space gap in the space from 1 MByte–16 MBytes. These bits are compared against A[23:20]. The memory space gap starting address must be a multiple of the memory space gap size. For example, a 2 MBytes gap must start at 2, 4, 6, 8, 10, 12, or 14 MBytes.
3:0	**RESERVED**

3.2.26 FBR—FRAME BUFFER RANGE REGISTER

Address Offset: 7C-7Fh
Default Value: 0000h
Attribute: Read/Write
Size: 32 bits

This 32-bit register enables and disables a frame buffer area and provides attribute settings for the frame buffer area. The attributes defined in this register are intended to increase the performance of the frame buffer. The FBR Register can be used to accommodate PCI devices that have their memory mapped onto PCI from the top of main memory to 4 GByte–512-KByte range (e.g., a linear frame buffer). If the Frame Buffer Range is located within the 1 MByte–16 MBytes main memory region where DRAM is populated, the Memory Space Gap Register must be programmed to include the Frame Buffer Range.

Bits	Description
31:20	**BUFFER OFFSET (BO):** BO defines the starting address of the frame buffer address space in increments of 1 MByte. This 12-bit field is compared directly against A[31:20]. The frame buffer range can either be located at the top of memory, including remapped memory or within the memory space gap (i.e., frame buffer range programmed below 16 MBytes and within main memory space. When bits [31:20] = 0000h and bit 12 = 0, all features defined by this register are disabled.
19:14	**RESERVED**
13	**BYTE MERGING (BM):** Byte merging permits CPU-to-PCI byte writes to the LBX posted write buffer to be combined into a single transfer on the PCI Bus, when appropriate. When BM is set to 1, byte merging on CPU-to-PCI posted write cycles is enabled. When BM is reset to 0, byte merging is disabled.
12	**128K VGA RANGE ATTRIBUTE ENABLE (VRAE):** When VRAE = 1, the attributes defined in this register (bits [13, 10:7]) also apply to the VGA memory range of A0000h–BFFFFh regardless of the value programmed in the Buffer Offset field. When VRAE = 0, the attributes do not apply to the VGA memory range. Note that this bit only affects the mentioned attributes of the VGA memory range and does not enable or disable accesses to the VGA memory range.
11:10	**RESERVED**
9	**NO LOCK REQUESTS (NLR):** When NLR is set to 1, the PCMC never requests exclusive access to a PCI resource via the PCI LOCK# signal in the range defined by this register. When NLR is reset to 0, exclusive access via the PCI LOCK# signal in the range defined by this register is enabled.
8	**RESERVED**
7	**TRANSPARENT BUFFER WRITES (TBW):** When set to a 1, this bit indicates that writes to the Frame Buffer Range need not be flushed for deadlock or coherence reasons on synchronization events (i.e., PCI master reads, and the FLSHBUF#/MEMREQ# protocol). When reset to 0, this bit indicates that upon synchronization events, flushing is required for Frame Buffer writes posted in the CPU-to-PCI Write Buffer in the LBX
6:4	**RESERVED**
3:0	**BUFFER RANGE (BR):** These bits define the size of the frame buffer address space, allowing up to 16 MBytes of frame buffer. If the Frame Buffer Range is within the memory space gap, the buffer range is limited to 8 MBytes and must be included within the memory space gap. The bits listed below in the Reserved Buffer Offset (BO) Bits column are ignored by the PCMC for the corresponding buffer sizes. **Bits[3:0]** — **Buffer Size** — **Reserved Buffer Offset (BO) Bits** 0000 — 1 MByte — None 0001 — 2 MBytes — [20] 0011 — 4 MBytes — [21:20] 0111 — 8 MBytes — [22:20] 1111 — 16 MBytes — [23:20] **NOTE:** (all other combinations are reserved)

2

4.0 PCMC ADDRESS MAP

The Pentium processor has two distinct physical address spaces: Memory and I/O. The memory address space is 4 GBytes and the I/O address space is 64 KBytes. The PCMC maps accesses to these address spaces as described in this section.

4.1 CPU Memory Address Map

Figure 9 shows the address map for the 4 GByte Host CPU memory address space. Depending on the address range and whether a memory gap is enabled via the MSG Register, the PCMC forwards CPU memory accesses to either main memory or PCI memory. Accesses forwarded to main memory invoke operations on the DRAM interface and accesses forwarded to PCI memory invoke operations on PCI. Mapping to the PCI Bus permits PCI or EISA/ISA Bus-based memory.

The main memory size ranges from 2 MBytes–192 MBytes for the 82434LX and 2 MBytes–512 MBytes for the 82434NX. Memory accesses above 192 MBytes (512 MBytes for the 82434NX) are always forwarded to PCI. In addition, a memory gap can be created in the 1 MByte–16 MBytes region that provides a window to PCI-based memory. The location and size of the gap is programmable. Accesses to addresses in the gap are ignored by the DRAM controller and forwarded to PCI. Note that CPU memory accesses that are forwarded to PCI (including the Memory Space Gap) are not cacheable. Only main memory controlled by the PCMC DRAM interface is cacheable.

4.2 System Management RAM—SMRAM

The PCMC supports the use of main memory as System Management RAM (SMRAM) enabling the use of System Management Mode. This function is enabled and disabled via the DRAM Control Register. When this function is disabled, the PCMC memory map is defined by the DRB and PAM Registers. When SMRAM is enabled, the PCMC reserves the top 64-KBytes of main memory for use as SMRAM.

SMRAM can also be placed at A0000–AFFFFh or B0000–BFFFFh via the SMRAM Space Register. Enhanced SMRAM features can also be enabled via this register. PCI masters can not access SMRAM when it is programmed to the A or B segments.

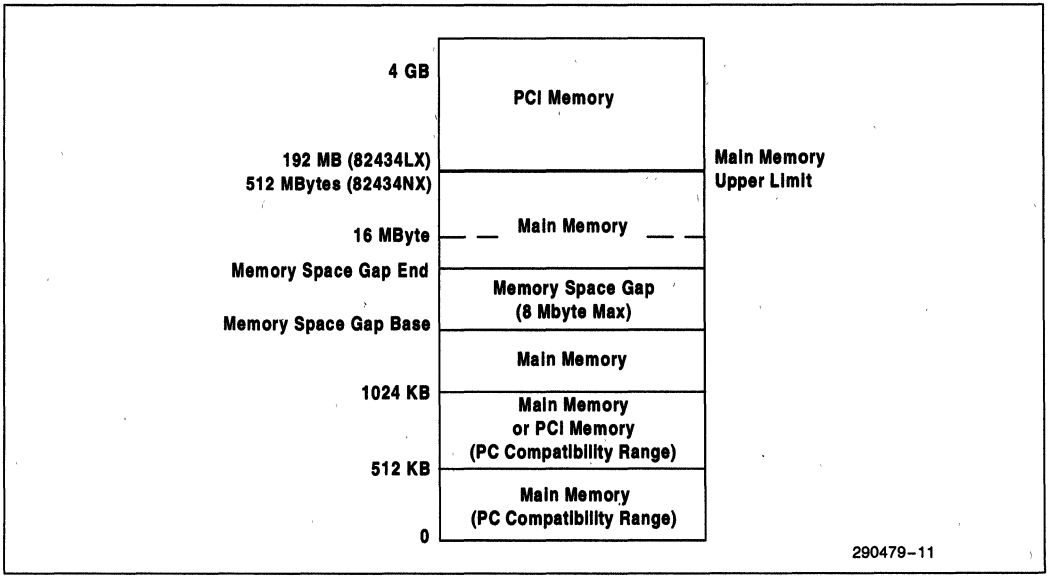


Figure 9. CPU Memory Address Map—Full Range

However, PCI masters can access SMRAM when the top of memory is selected.

When the 82434NX PCMC detects a CPU stop grant special cycle (M/IO# = 0, D/C# = 0, W/R# = 1, A4 = 1, BE[7:0]# = FBh), it generates a PCI Stop Grant Special cycle, with 0002h in the message field (AD[15:0]) and 0012h in the message dependent data field (AD[31:16]) during the first data phase (IRDY# asserted).

4.3 PC Compatibility Range

The PC Compatibility Range is the first MByte of the Memory Map. The 512 KByte–1 MByte range is subdivided into several regions as shown in Figure 10. Each region is provided with programmable attributes in the PAM Registers. The attributes are Read Enable (RE), Write Enable (WE) and Cache Enable (CE). The attributes determine readability, writeability and cacheability of the corresponding memory region. When the associated bit in the PAM Register is set to a 1, the attribute is enabled and when set to a 0 the attribute is disabled. The following rules apply for cacheability in the first level and second level caches:

1. If RE = 1, WE = 1, and CE = 1, the region is cacheable in the first level and second level caches.
2. If RE = 1, WE = 0, and CE = 1, the region is cacheable only on code reads (i.e., D/C# = 0). Data reads do not result in a line fill. Writes to the region are not serviced by the secondary cache, but are forwarded to PCI.

2

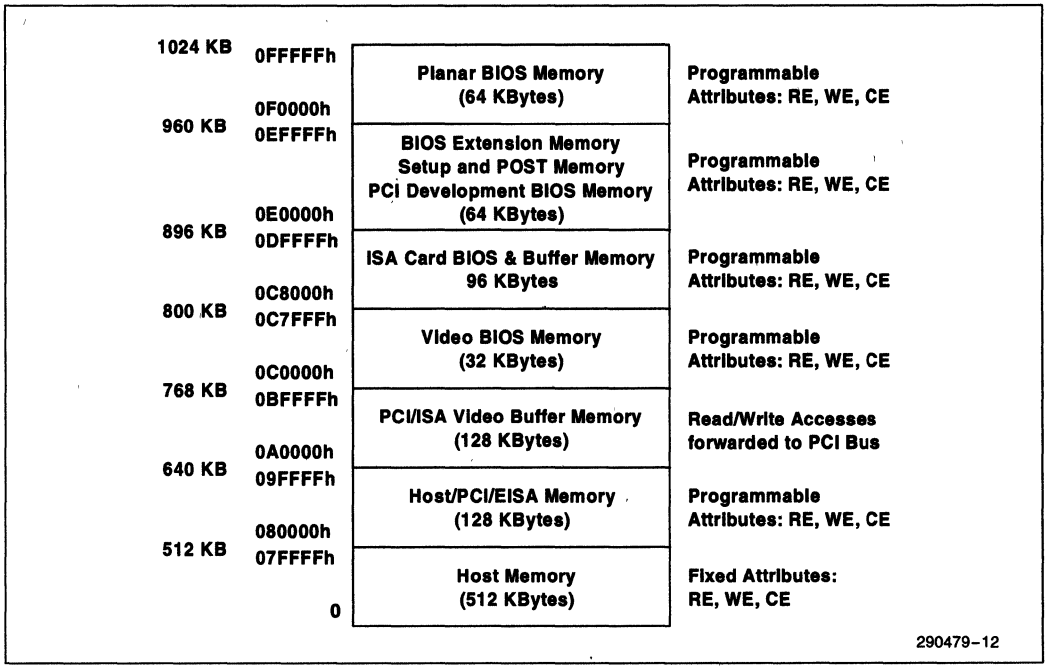
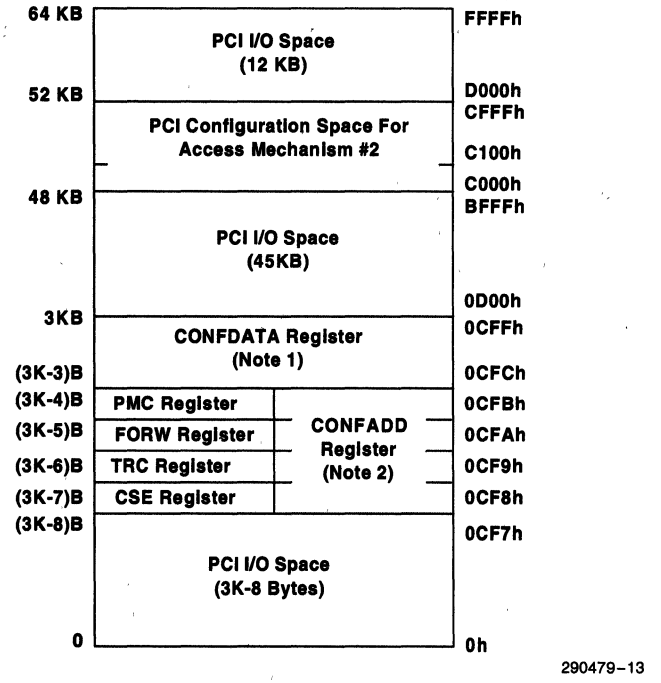


Figure 10. CPU Memory Address Map—PC Compatibility Range

The RE and WE bits for each region are used to shadow BIOS ROM in main memory for improved system performance. To shadow a BIOS area, RE is reset to 0 and WE is set to 1. RE is set to 1 and WE is reset to 0. Any writes to the BIOS area are forwarded to PCI.

4.4 I/O Address Map

I/O devices (other than the PCMC) are not supported on the Host Bus. The PCMC generates PCI Bus cycles for all CPU I/O accesses, except to the PCMC internal registers. Figure 11 shows the mapping for the CPU I/O address space. For the 82434LX, three PCMC registers are located in the CPU I/O address space—the Configuration Space Enable (CSE) Register, the Turbo-Reset Control (TRC) Register, and the Forward (FORW) Register.



NOTES:
1. This 82434NX register is only visible when configuration access mechanism #1 is enabled (via bit 31 of the CONFADD Register). Otherwise, this I/O range is in PCI I/O space.
2. This 82434NX register is accessed during Dword read/writes to 0CF8h. Byte or word cycles access the corresponding 8-bit registers, even if configuration access mechanism #1 is enabled.

Figure 11. CPU I/O Address Map

For the 82434NX, six PCMC registers are located in the CPU I/O address space—the Configuration Space Enable (CSE) Register, the Configuration Address Register (CONFADD), the Turbo-Reset Control (TRC) Register, the Forward (FORW) Register, the PCI Mechanism Control (PMC) Register, and the Configuration Data (CONFDATA) Register.

Except for the I/O locations of the above mentioned registers, all other CPU I/O accesses are mapped to either PCI I/O space or PCI configuration space. If the access is to PCI I/O space, the PCI address is the same as the CPU address. If the access is to PCI configuration space, the CPU address is mapped to a configuration space address as described in Section 3.0, Register Description.

If configuration space is enabled via the CSE Register (access mechanism #2), the PCMC maps accesses in the address range of C100h to CFFFh to PCI configuration space. Accesses to the PCMC configuration register range (C000h to C0FFh) are intercepted by the PCMC and not forwarded to PCI. If the configuration space is disabled in the CSE Register, CPU accesses to the configuration address range (C000h to CFFFh) are forwarded to PCI I/O space.

5.0 SECOND LEVEL CACHE INTERFACE

This section describes the second level cache interface for the 82434LX Cache (Section 5.1) and the 82434NX Cache (Section 5.2). The differences are in the following areas:

1. The 82434LX supports both write-through and write-back cache policies. The 82434NX only supports the write-back policy.
2. The 82434LX timings are for 60 and 66 MHz and the 82434NX timings are for 50, 60, and 66 MHz. Note that the cycle latencies for 60 and 66 MHz are the same for both devices.
3. When burst SRAMs are used to implement the secondary cache, address latches are not needed for the 82434NX type SRAM connectivity. However, a control bit has been added to the 82434NX that permits address latches for 82434LX type SRAM connectivity.
4. A low-power second level cache standby mode has been added to the 82434NX.
5. There are new or changed cache control bits as indicated by the shading in Section 3.0, Register Description. For example, the 82434NX supports zero wait-state cache at 50 MHz via the zero wait-state control bit.

NOTE:

- Second level cache sizes and organization are the same for the 82434LX and 82434NX.
- The general operation of the second level cache write-back policy is the same for the 82434LX and 82434NX. For example, the Valid and Modified bits operate the same for both devices. In addition, snoop operations are the same for both devices, as well as the handling of flush, flush acknowledge, and write-back special cycles.

5.1 82434LX Cache

The 82434LX PCMC integrates a high performance write-back/write-through second level cache controller providing integrated tags and a full first level and second level cache coherency mechanism. The second level cache controller can be configured to support either a 256-KByte cache or a 512 KByte cache using either synchronous burst SRAMs or standard asynchronous SRAMs. The cache is direct mapped and can be configured to support either a write-back or write-through write policy. Parity on the second level cache data SRAMs is optional.

The 82434LX contains 4096 address tags. Each tag represents a *sector* in the second level cache. If the second level cache is 256-KByte, each tag represents two cache lines. If the second level cache is 512-KByte, each tag represents four cache lines. Thus, in the 256-KByte configuration each sector contains two lines. In the 512-KByte configuration, each sector contains four lines. *Valid* and *modified* status bits are kept on a per line basis. Thus, in the case of a 256-KByte cache each tag has two valid bits and two modified bits associated with it. In the case of a 512-KByte cache each tag has four valid and four modified bits associated with it. Upon a CPU read cache miss, the PCMC inspects the valid and modified bits within the addressed sector and writes back to main memory only the lines marked both valid and modified. All of the lines in the sector are then invalidated. The line fill will then occur and the valid bit associated with the allocated line will be set. Only the requested line will be fetched from main memory and written into the cache. If no write-back is required, all of the lines in the sector are marked invalid. The line fill then occurs and the valid bit associated with the allocated line will be set. Lines are not allocated on write misses. When a CPU write hits a line in the second level cache, the modified bit for the line is set.

The second level cache is optional to allow the 82434LX PCMC to be used in a low cost configuration. A 256-KByte cache is implemented with a single bank of eight 32K x 9 SRAMs if parity is supported or 32K x 8 SRAMs if parity is not supported on the cache. A 512-KByte cache is implemented with four 64K x 18 SRAMs if parity is supported or 64K x 16 SRAMs if parity is not supported on the cache. Two 74AS373 latches complete the cache. Only main memory controlled by the PCMC DRAM interface is cached. Memory on PCI is not cached.

Figure 12 and Figure 13 depict the organization of the internal tags in the PCMC configured for a 256 KByte cache and a 512-KByte cache.

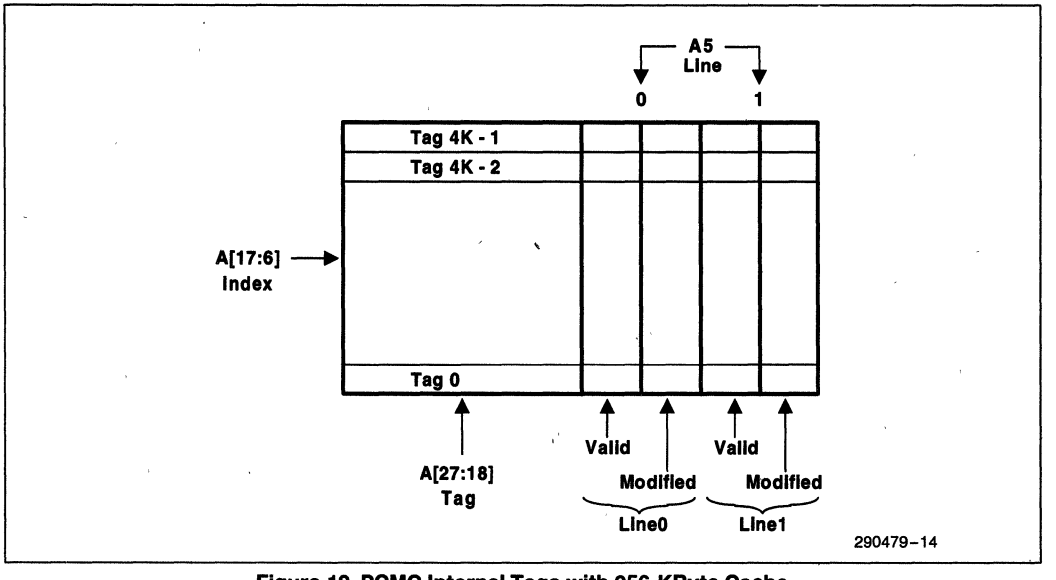


Figure 12. PCMC Internal Tags with 256-KByte Cache

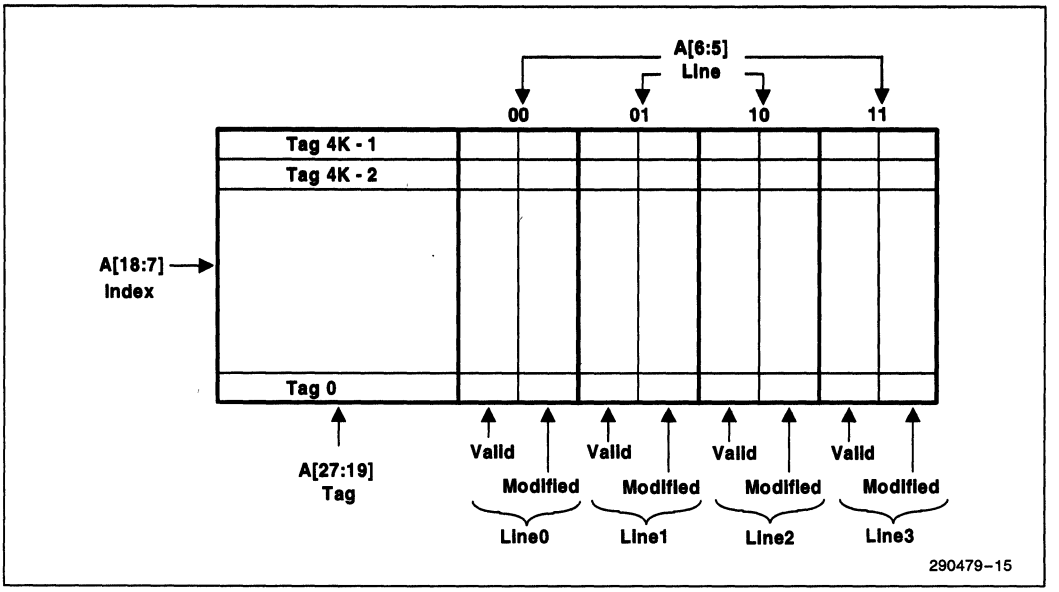


Figure 13. PCMC Internal Tags with 512-KByte Cache

In the 256-KByte cache configuration A[17:6] form the tag RAM index. The ten tag bits read from the tag RAM are compared against A[27:18] from the host address bus. Two valid bits and two modified bits are kept per tag in this configuration. Host address bit 5 is used to select between lines 0 and 1 within a sector. In the 512-KByte cache configuration A[18:7] form the tag RAM index. The nine bits read from the tag RAM are compared against A[27:19] from the host bus. Four valid bits and four modified bits are kept per tag. Host address bits 5 and 6 are used to select between lines 0, 1, 2 and 3 within a sector.

The Secondary Cache Controller Register at offset 52h in configuration space controls the secondary cache size, write and allocation policies, and SRAM type. The cache can also be enabled and disabled via this register.

Figure 14 through Figure 18 show the connections between the PCMC and the external cache data SRAMs and latches.

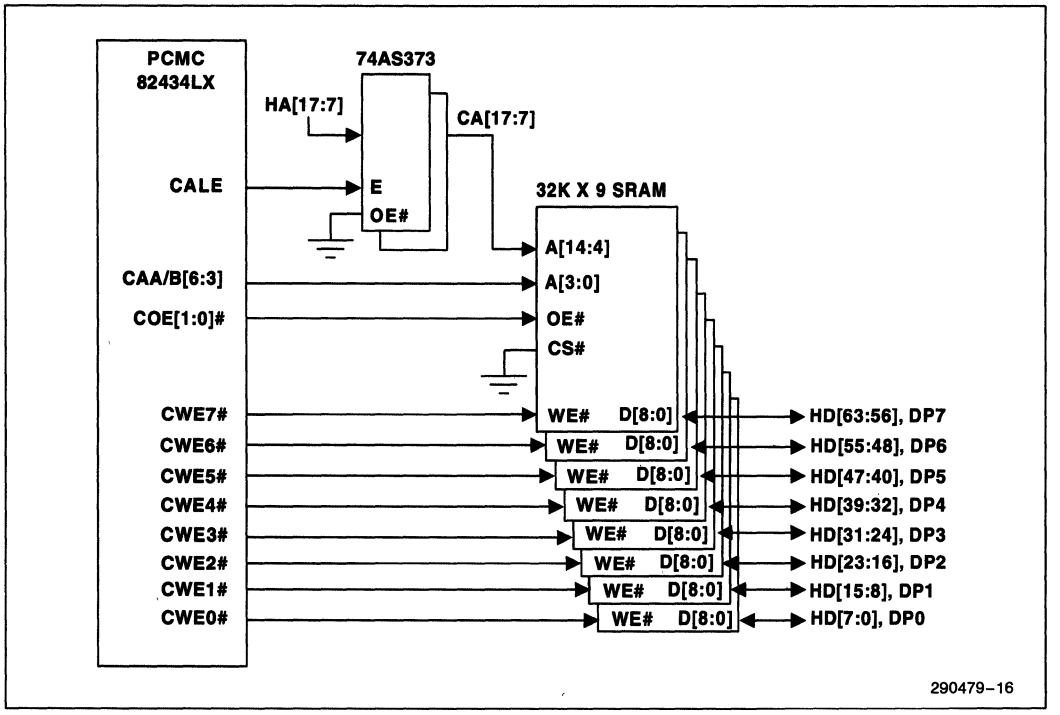


Figure 14. 82434LX Connections to 256-KByte Cache with Standard SRAM

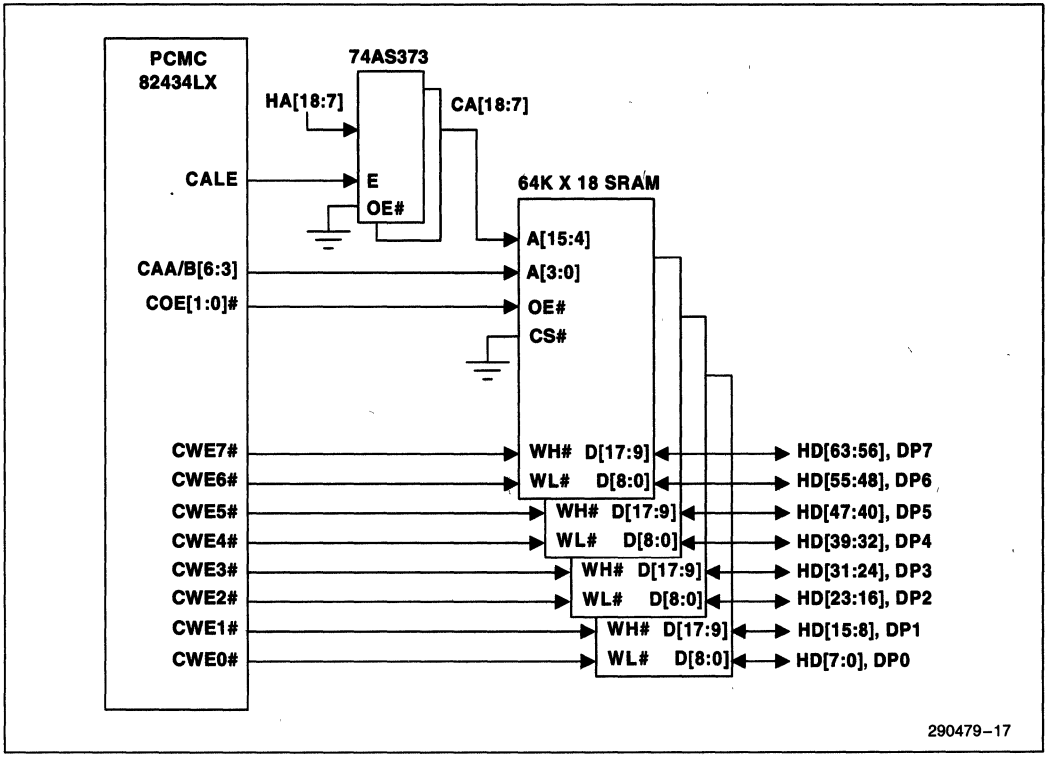


Figure 15. 82434LX Connections to 512-KByte Cache with Standard SRAM

2

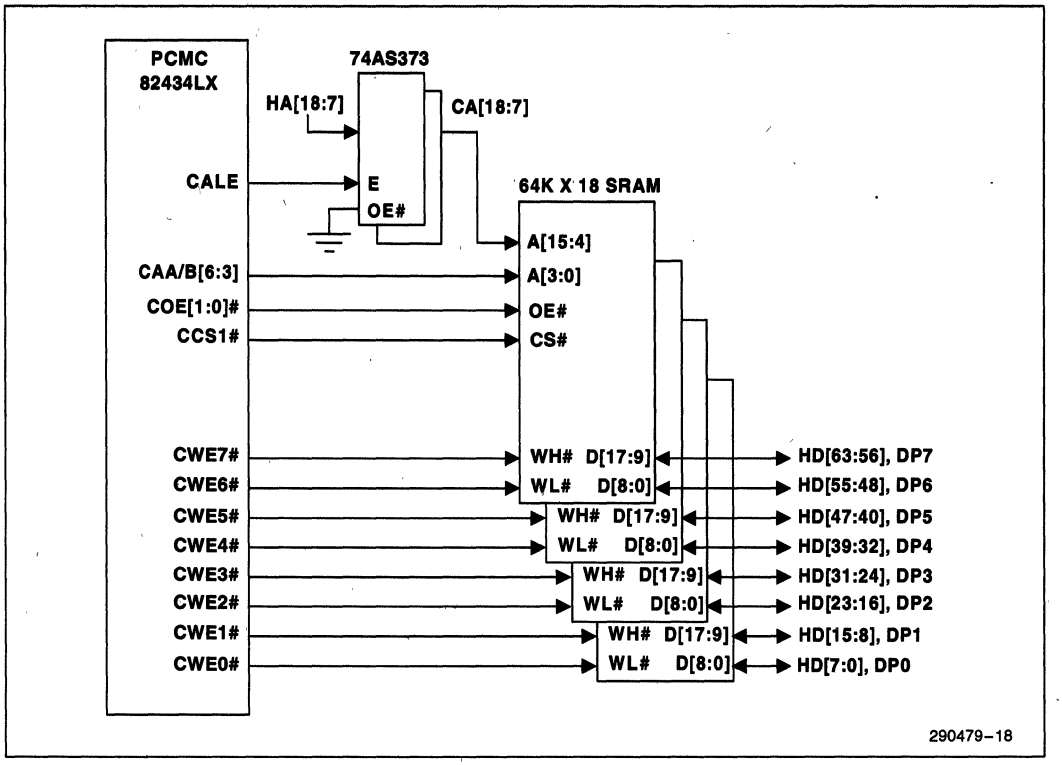


Figure 16. 82434LX Connections to 512-KByte Cache with Dual-Byte Select Standard SRAMs

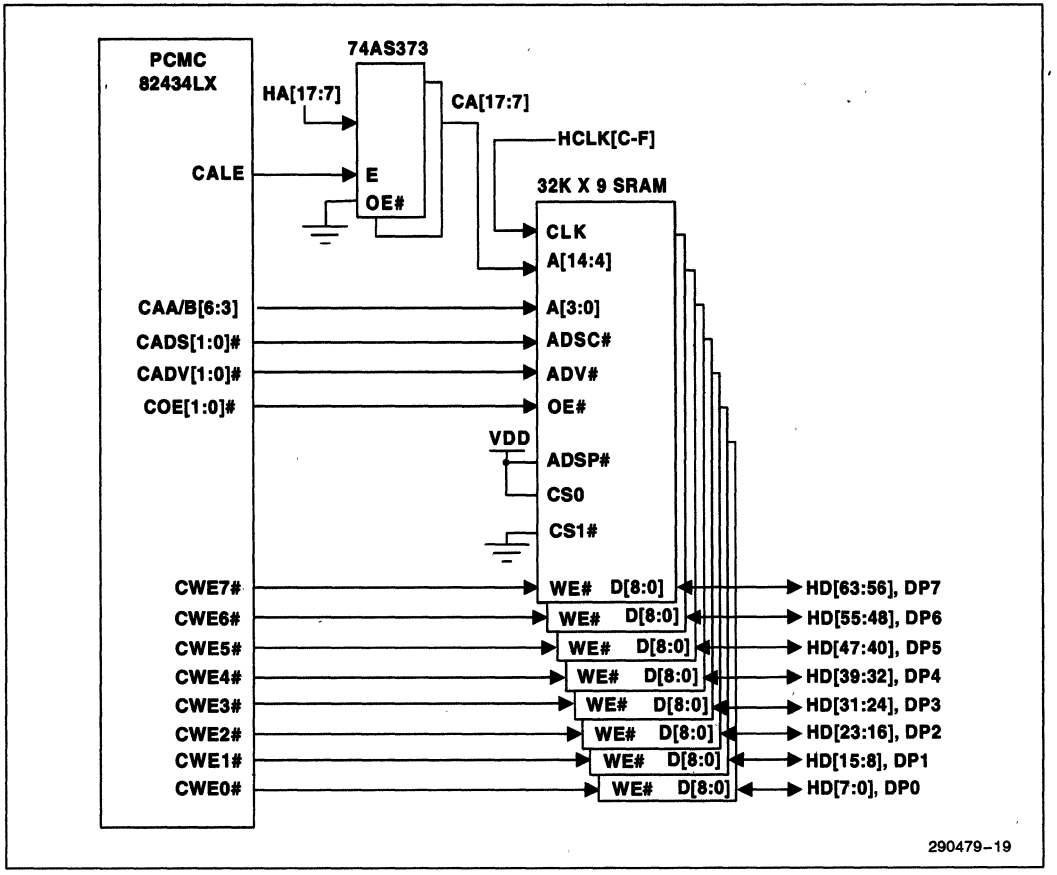


Figure 17. 82434LX Connections to 256-KByte Cache with Burst SRAM

2

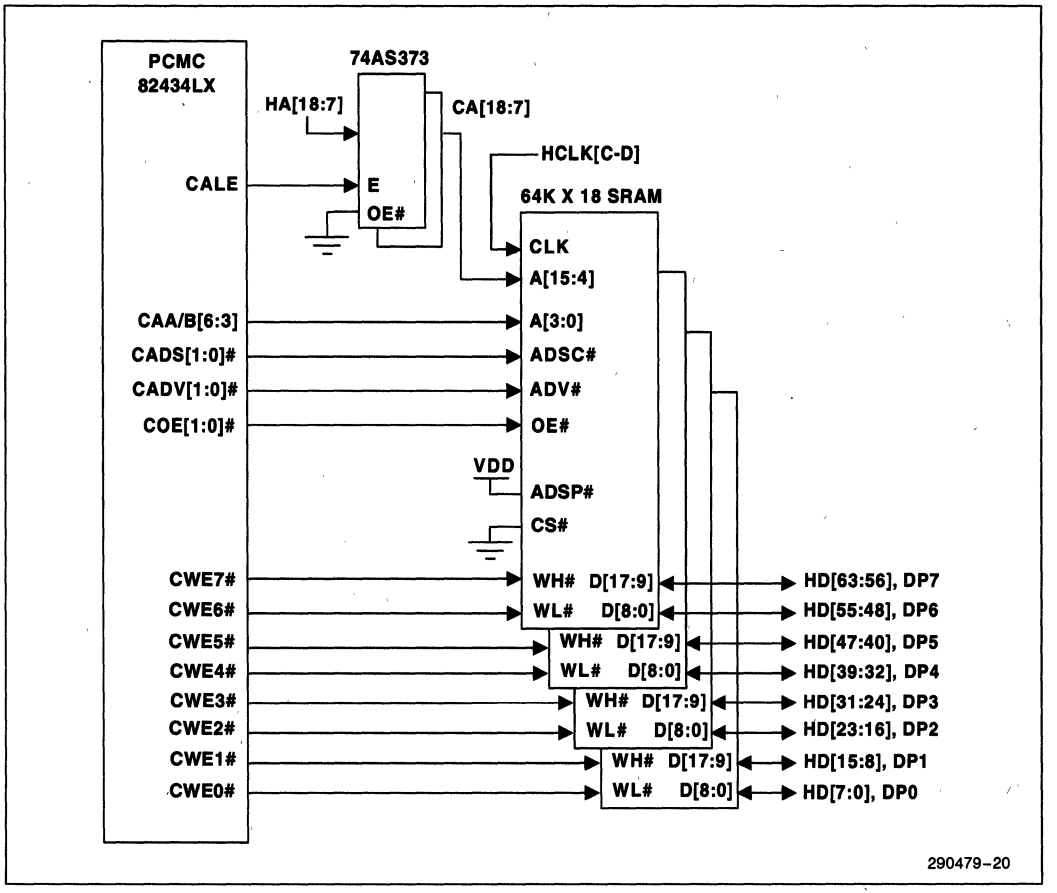


Figure 18. 82434LX Connections for 512-KByte Cache with Burst SRAM

When CALE is asserted, HA[18:7] flow through the address latch. When CALE is negated the address is captured in the latch allowing the processor to pipeline the next bus cycle onto the address bus. Two copies of CA[6:3], COE#, CADS# and CADV# are provided to reduce capacitive loading. Both copies should be used when the second level cache is implemented with eight 32K x 8 or 32K x 9 SRAMs. Either both copies or only one copy can be used with 64K x 18 or 64K x 16 SRAMs as determined by the system board layout and timing analysis. The two copies are always driven to the same logic level. CAA[4:3] and CAB[4:3] are used to count through the Pentium processor burst order when standard SRAMs are used to implement the cache.

With burst SRAMs, the address counting is provided inside the SRAMs. In this case, CAA[4:3] and CAB[4:3] are only used at the beginning of a cycle to load the initial low order address bits into the burst SRAMs. During CPU accesses, host address lines 6 and 5 are propagated to the CAA[6:5] and CAB[6:5] lines and are internally latched. When a CPU read cycle forces a line replacement in the second level cache, all modified lines within the addressed sector are written back to main memory. The PCMC uses CAA[6:5] and CAB[6:5] to select among the lines within the sector. The Cache Output Enables (COE[1:0]#) are asserted to enable the SRAMs to drive data onto the host data bus. The Cache Write Enables (CWE[7:0]#) allow byte control during CPU writes to the second level cache.

An asynchronous SRAM 512-KByte cache can be implemented with two different types of SRAM byte control. Figure 15 depicts the PCMC connections to a 512 KByte cache using 64K x 18 SRAMs or 64K x 16 SRAMs with two write enables per SRAM. Each SRAM has a high and low write enable. Figure 16 depicts the PCMC connections to a 512-KByte cache using 64K x 18 SRAMs or 64K x 16 SRAMs with two byte select lines per SRAM. Each SRAM has a high and low byte select.

The type of cache byte control (write enable or byte select) is programmed in the Cache Byte Control bit in the Secondary Cache Control Register at configuration space offset 52h. When this bit is set to 0, byte select control is used. In this mode, the CBS[7:0]# lines are multiplexed onto pins 90, 91, and 95-100 and CR/W[1:0]# pins are multiplexed onto pins 93 and 94. When this bit is set to 1, byte write enable control is used. In this mode, the CWE[7:0]# lines are multiplexed onto pins 90, 91, and 95-100. CADS[1:0]# and CADV[1:0]# are only used with burst SRAMs. The Cache Address Strobes (CADS[1:0]#) are asserted to cause the burst SRAMs to latch the cache address at the beginning of a second level cache access. CADS[1:0]# can be connected to either ADSP# or ADSC# on the SRAMs. The Cache Advance signals (CADV[1:0]#) are asserted to cause the burst SRAMs to advance to the next address of the burst sequence.

5.1.1 CLOCK LATENCIES (82434LX)

Table 5 and Table 6 list the latencies for various CPU transfers to or from the second level cache for standard SRAMs and burst SRAMs. Standard SRAM access times of 12 ns and 15 ns are recommended for 66 MHz and 60 MHz operation, respectively. Burst SRAM clock access times of 8 ns and 9 ns are recommended for 66 MHz and 60 MHz operation, respectively. Precise SRAM timing requirements should be determined by system board electrical simulation with SRAM I/O buffer models.

Table 5. Second Level Cache Latencies with Standard SRAM (82434LX)

Cycle Type	HCLK Count
Burst Read	3-2-2-2
Burst Write	4-2-2-2
Single Read	3
Single Write	4
Pipelined Back to Back Burst Reads	3-2-2-2/3-2-2-2
Burst Read followed by Pipelined Write	3-2-2-2/4

Table 6. Second Level Cache Latencies with Burst SRAM (82434LX)

Cycle Type	HCLK Count
Burst Read	3-1-1-1
Burst Write	3-1-1-1
Single Read	3
Single Write	3
Pipelined Back to Back Burst Reads	3-1-1-1/1-1-1-1
Read Followed by Pipelined Write	3-1-1-1/2

5.1.2 STANDARD SRAM CACHE CYCLES (82434LX)

The following sections describe the activity of the second level cache interface when standard asynchronous SRAMs are used to implement the cache.

5.1.2.1 Burst Read (82434LX)

Figure 19 depicts a burst read from the second level cache with standard SRAMs. The CPU initiates the read cycle by driving address and status onto the bus and asserting ADS#. Initially, the CA[6:3] are a propagation delay from the host address lines A[6:3]. Upon sampling W/R# active and M/IO# inactive, while ADS# is asserted, the PCMC asserts COE# to begin a read cycle from the SRAMs. CALE is negated, latching the address lines on the SRAM address inputs, allowing the CPU to pipeline a new address onto the bus. CA[4:3] cycle through the Pentium processor burst order, completing the cycle. PEN# is asserted with the first BRDY# and negated with the last BRDY# if parity is implemented on the second level cache data SRAMs and the MCHK DRAM/Second Level Cache Data Parity bit in the Error Command Register (offset 70h) is set.

Figure 20 depicts a burst read from the second level cache with standard 16- or 18-bit wide dual-byte select SRAMs. A single read cycle from the second level cache is very similar to the first transfer of a burst read cycle. CALE is not negated throughout the cycle. COE# is asserted as shown above, but is negated with BRDY#.

When the Secondary Cache Allocation (SCA) bit in the Secondary Cache Control Register is set to 1, the PCMC performs a line fill in the secondary cache, even if the CACHE# signal from the CPU is inactive. In this case, AHOLD is asserted to prevent the CPU from beginning a new cycle while the second level cache line fill is completing.

Back-to-back pipelined burst reads from the second level cache are shown in the Figure 21.

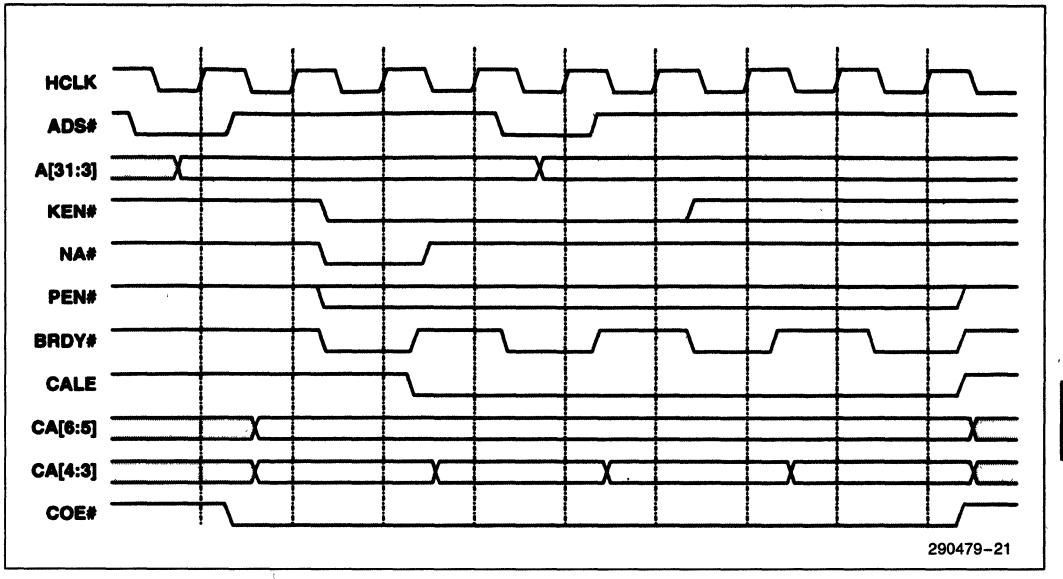


Figure 19. CPU Burst Read from Second Level Cache with Standard SRAM (82434LX)

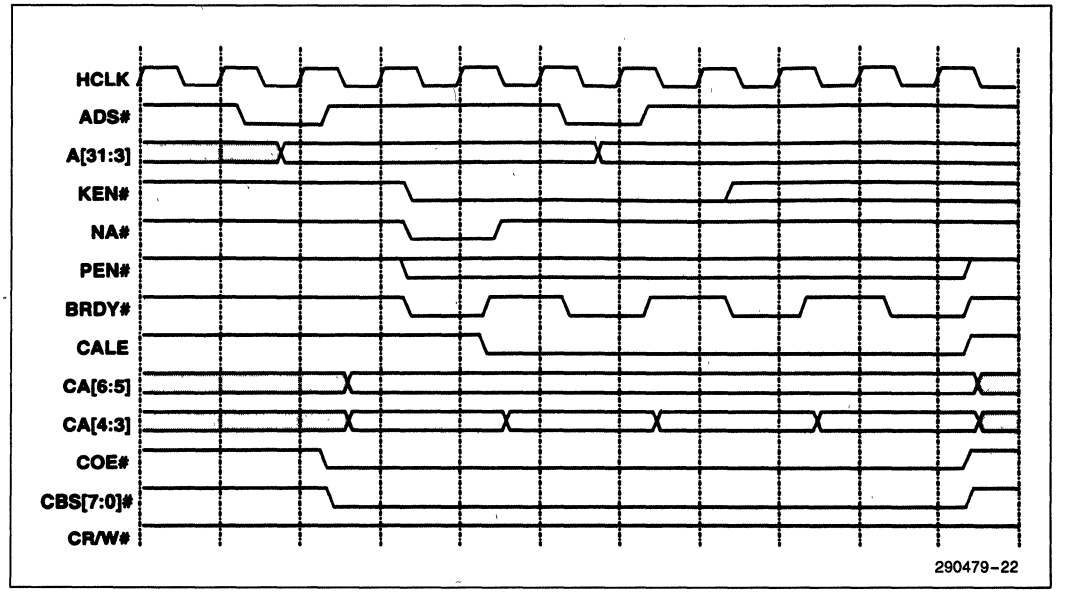


Figure 20. Burst Read from Second Level Cache with Dual-Byte Select SRAMs (82434LX)

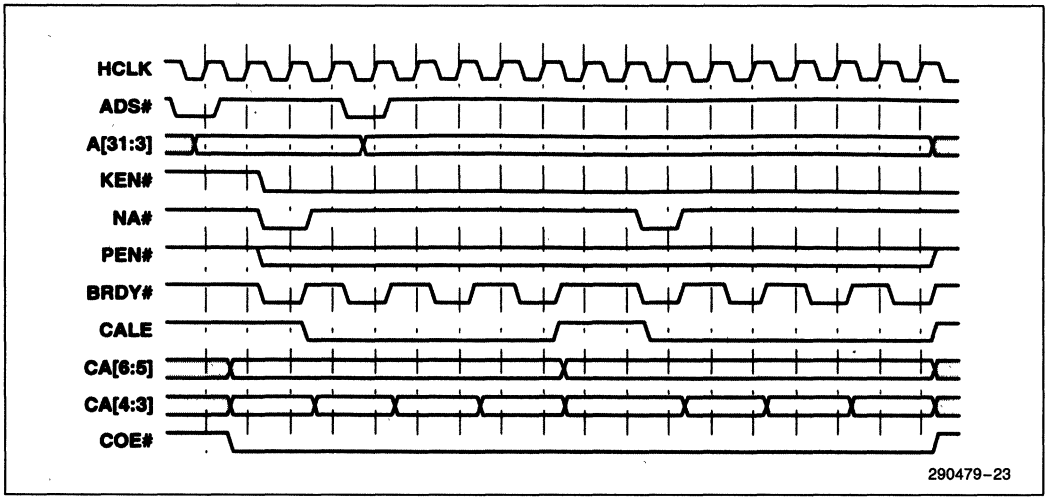


Figure 21. Pipelined Back-to-Back Burst Reads from Second Level Cache with Standard SRAM (82434LX)

Due to assertion of NA#, the CPU drives a new address onto the bus before the first cycle is complete. In this case, the second cycle is a hit in the second level cache. Immediately upon completion of the first read cycle, the PCMC begins the second cycle. When the first cycle completes, the PCMC drives the new address to the SRAMs on CA[6:3] and asserts CALE. The second cycle is very similar to the first, completing at a rate of 3-2-2-2. The cache address lines must be held at the SRAM address inputs until the first cycle completes. Only after the last BRDY# is returned, can CALE be asserted and CA[6:3] be changed. Thus, the pipelined cycle completes at the same rate as a non-pipelined cycle.

5.1.2.2 Burst Write (82434LX)

A burst write cycle is used to write back a cache line from the first level cache to either the second level cache or DRAM. Figure 22 depicts a burst write cycle to the second level cache with standard SRAMs. The CPU initiates the write cycle by driving address and status onto the bus and asserting ADS#. Initially, the CA[6:3] propagate from the host address lines A[6:3]. CALE is negated, latching the address lines on the SRAM address inputs, allowing the CPU to pipeline a new address onto the bus. Burst write cycles from the Pentium processor always begin with the low order Qword and advances to the high order Qword. CWE[7:0]# are generated from an internally delayed version of HCLK, providing address setup time to CWE[7:0]# falling and data setup time to CWE[7:0]# rising edges. HIG[4:0] are driven to PCMWQ (Post CPU to Memory Write Buffer Qword) only when the PCMC is programmed for a write-through write policy. When programmed for write-back mode, the modified bit associated with the line is set within the PCMC. The single write cycle is very similar to the first write of a burst write cycle. A burst read cycle followed by a pipelined write cycle with standard SRAMs is depicted in Figure 24.

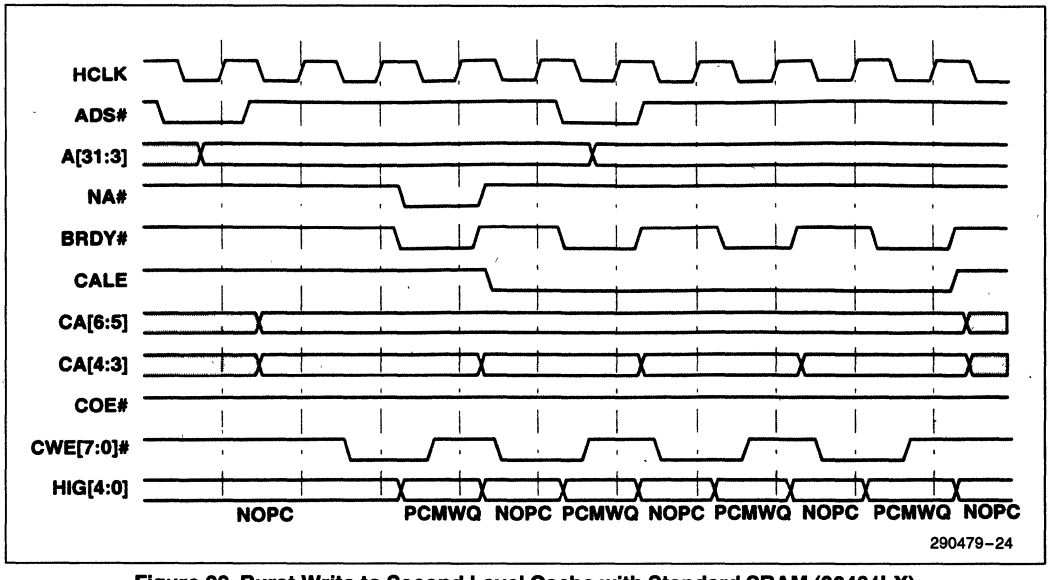


Figure 22. Burst Write to Second Level Cache with Standard SRAM (82434LX)

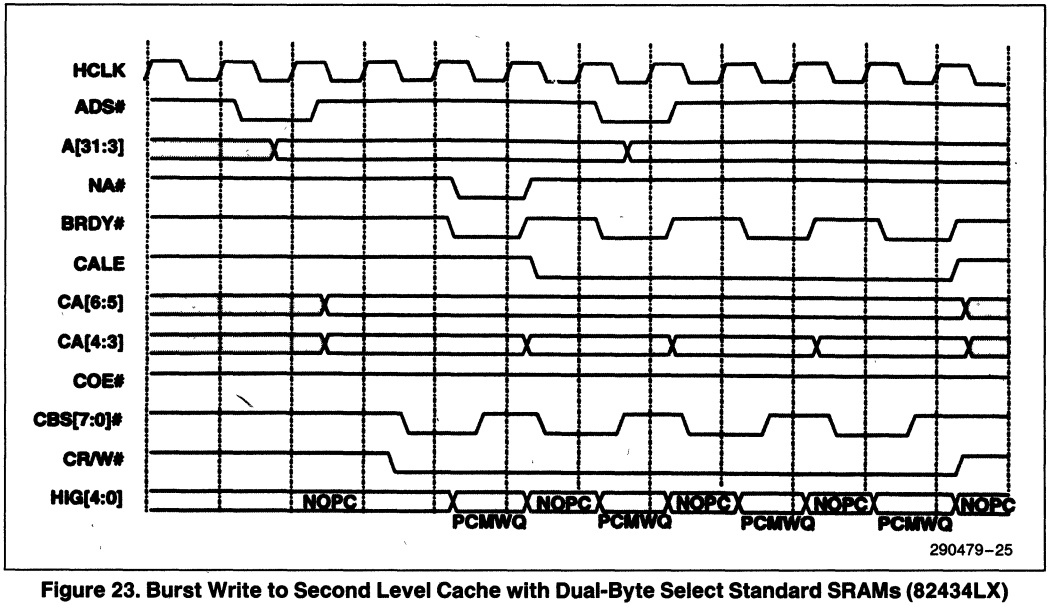


Figure 23. Burst Write to Second Level Cache with Dual-Byte Select Standard SRAMs (82434LX)

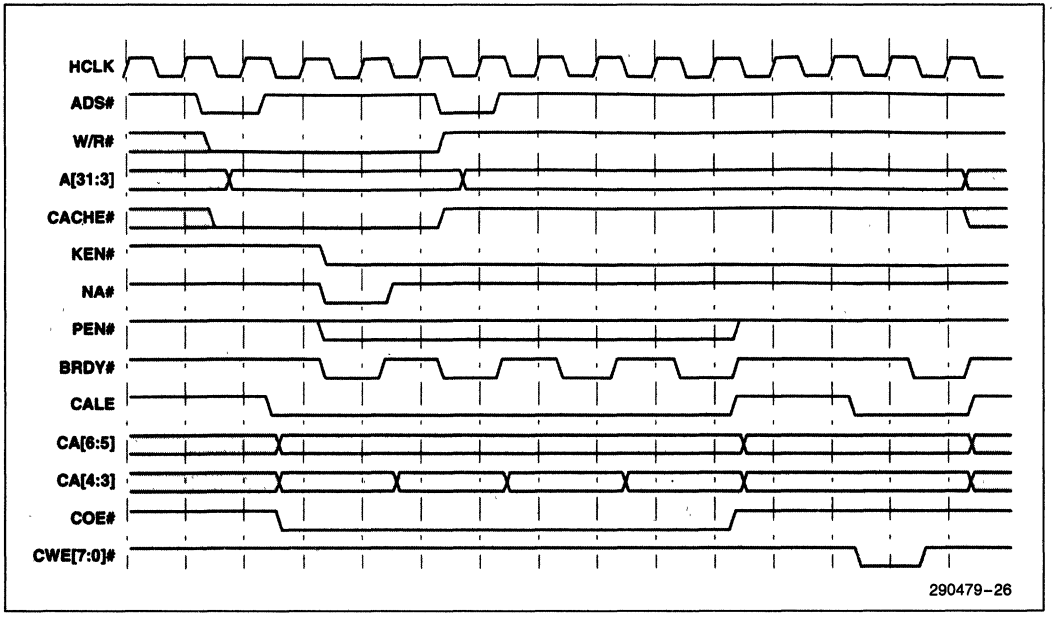


Figure 24. Burst Read Followed by Pipelined Write with Standard SRAM (82434LX)

5.1.2.3 Cache Line Fill (82434LX)

If the CPU issues a memory read cycle to cacheable memory that is not in the second level cache, a first and second level cache line fill occurs. Figure 25 depicts a CPU read cycle that results in a line fill into the first and second level caches.

Figure 27 depicts the host bus activity during a CPU read cycle that forces a write-back from the second level cache to the CPU-to-memory posted write buffer as the DRAM read cycle begins.

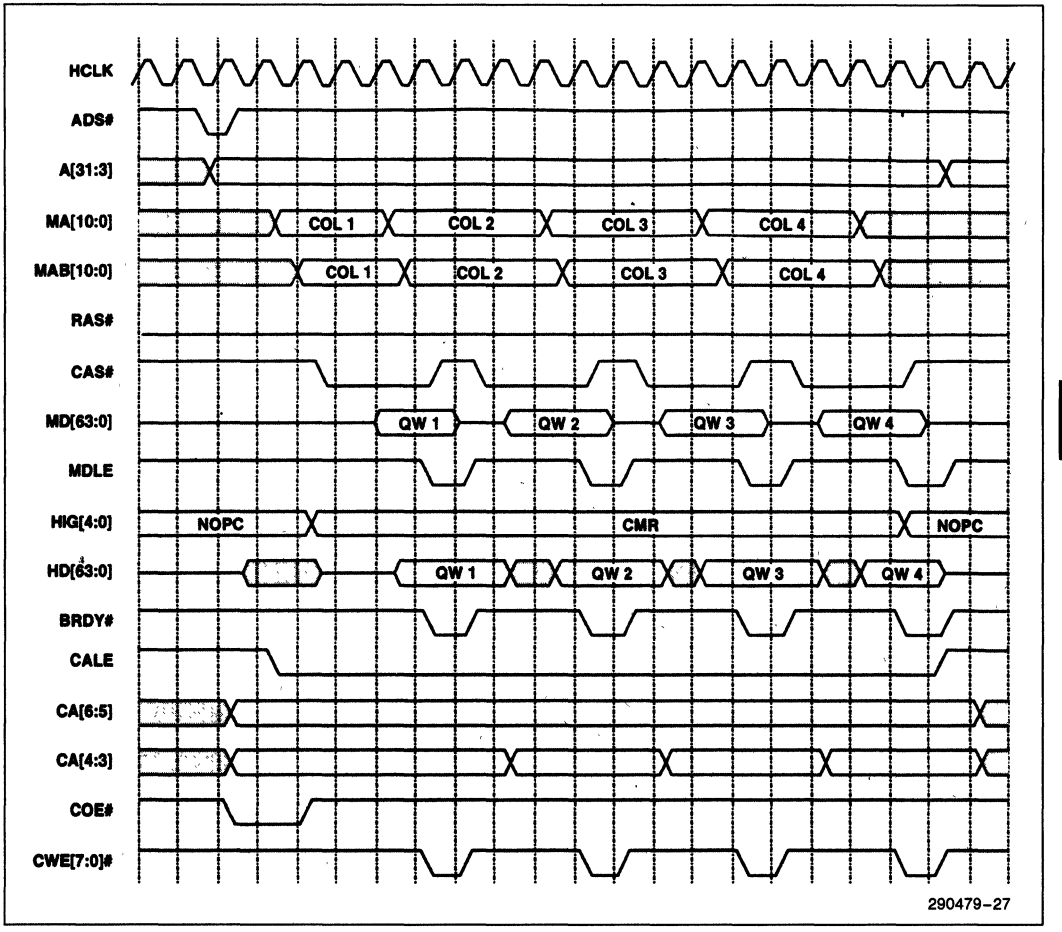


Figure 25. Cache Line Fill with Standard SRAM, DRAM Page Hit (82434LX)

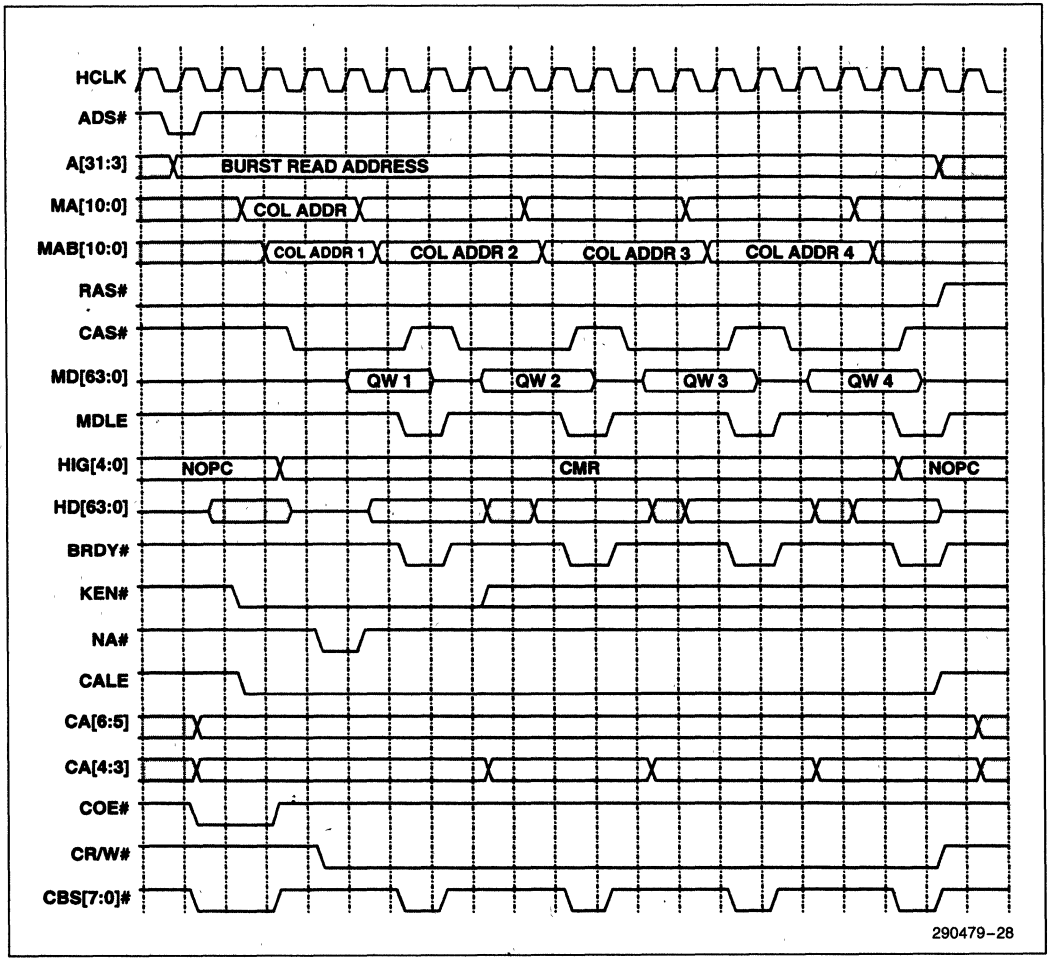


Figure 26. Cache Line Fill with Dual-Byte Select Standard SRAM, DRAM Page Hit (82434LX)

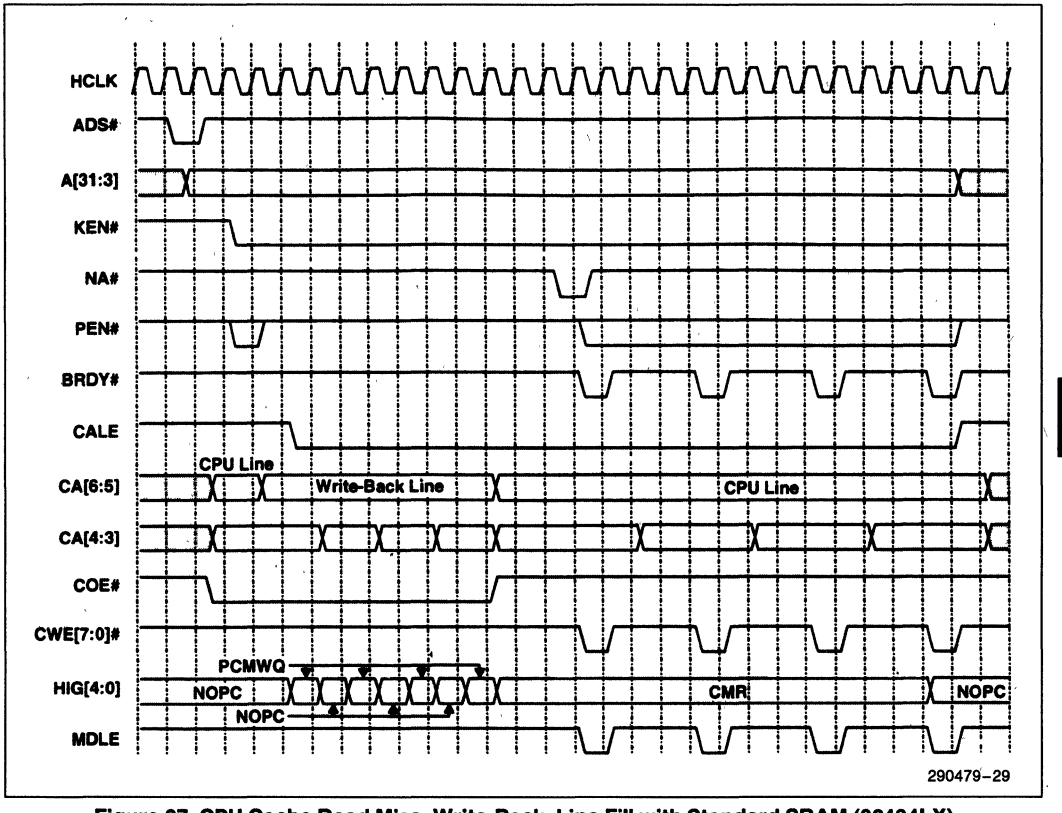


Figure 27. CPU Cache Read Miss, Write-Back, Line Fill with Standard SRAM (82434LX)

The CPU issues a memory read cycle that misses in the second level cache. In this instance, a modified line in the second level cache must be written back to main memory before the new line can be filled into the cache. The PCMC inspects the valid and modified bits for each of the lines within the addressed sector and writes back only the valid lines within the sector that are in the modified state. During the write-back cycle, CA[4:3] begin with the initial value driven by the Pentium processor and proceed in the Pentium processor burst order. CA[6:5] are used to count through the lines within the addressed sector. When two or more lines must be written back to main memory, CA[6:5] count in the direction from line 0 to line 3. CA[6:5] advance to the next line to be written back to main memory, skipping lines that are not modified. Figure 23 depicts the case of just one of the lines in a sector being written back to main memory. In this case, the entire line can be posted in the CPU-to-Main memory posted write buffer by driving the HIG[4:0] lines to the PCMWQ command as each Qword is read from the cache. At the same time, the required DRAM read cycle is beginning. As soon as the de-allocated line is written into the posted write buffer, the HIG[4:0] lines are driven to CMR (CPU Memory Read) to allow data to propagate from the DRAM data lines to the CPU data lines. The CWE[7:0]# lines are not generated from a delayed version of HCLK (as they are in the case of CPU to second level cache burst write), but from ordinary HCLK rising edges. CMR is driven on the HIG[4:0] lines

throughout the DRAM read portion of the cycle. With the fourth assertion of BRDY# the HIG[4:0] lines change to NOPC. The LBXs however, do not tri-state the host data lines until MDLE rises. CWE[7:0]# and MDLE track such that MDLE will not rise before CWE[7:0]#. Thus, the LBXs continue to drive the host data lines until CWE[7:0]# are negated. CA[6:3] remain at the valid values until the clock after the last BRDY#, providing address hold time to CWE[7:0]# rising.

PEN# is asserted as shown if the MCHK DRAM/L2 Cache Data Parity Error bit in the Error Command Register (offset 70h) is set. If the second level cache supports parity, PEN# is always asserted during CPU read cycles in the third clock in case the cycle hits in the cache.

If more than one line must be written back to main memory, the PCMC fills the CPU-to-Main Memory Posted Write Buffer and loads another Qword into the buffer as each Qword write completes into main memory. The writes into DRAM proceed as page hit write cycles from one line to the next, completing at a rate of X-4-4-4-5-4-4-4-5-4-4-4 for a three line write-back. All modified lines except for the last one to be written back are posted and written to memory before the DRAM read cycle begins. The last line to be written back is posted as the DRAM read cycle begins. Thus, the read data is returned to the CPU before the last line is retired to memory.

The line which was written into the second level cache is marked valid and unmodified by the PCMC. All the other lines in the sector are marked invalid. A subsequent CPU read cycle which hits in the same sector (but a different line) in the second level cache would then simply result in a line fill without any write-back.

5.1.3 BURST SRAM CACHE CYCLES (82434LX)

The following sections show the activity of the second level cache interface when burst SRAMs are used for the second level cache.

5.1.3.1 Burst Read (82434LX)

Figure 28 depicts a burst read from the second level cache with burst SRAMs.

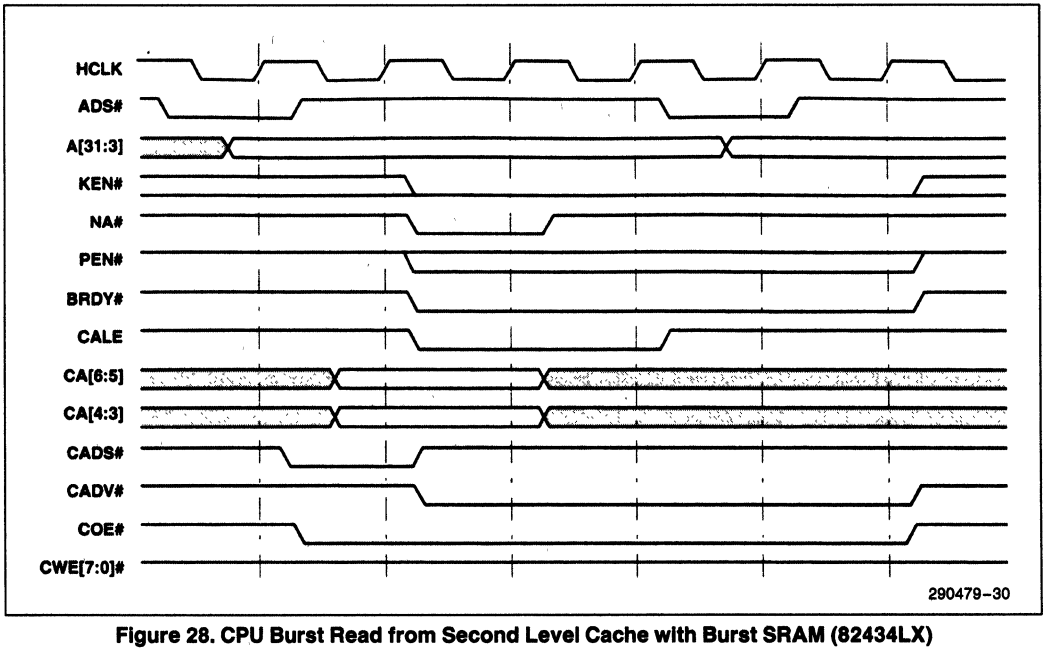


Figure 28. CPU Burst Read from Second Level Cache with Burst SRAM (82434LX)

The cycle begins with the CPU driving address and status onto Host Bus and asserting ADS#. The PCMC asserts CADS# and COE# in the second clock. After the address is latched by the burst SRAMs and the PCMC determines that no write-back cycles are required from the second level cache, CALE is negated. Back-to-back burst reads from the second level cache are shown in Figure 29.

When the Secondary Cache Allocation (SCA) bit in the Secondary Cache Control Register is set to 1, the PCMC performs a line fill in the secondary cache, even if the CACHE# signal from the CPU is negated. In this case, AHOLD is asserted to prevent the CPU from beginning a new cycle while the second level cache line fill is completing.

Back-to-back burst reads which hit in the second level cache complete at a rate of 3-1-1-1/1-1-1-1 with burst SRAMs. As the last BRDY# is being returned to the CPU, the PCMC asserts CADS# causing the SRAMs to latch the new address. This allows the data for the second cycle to be transferred to the CPU on the clock after the first cycle completes.

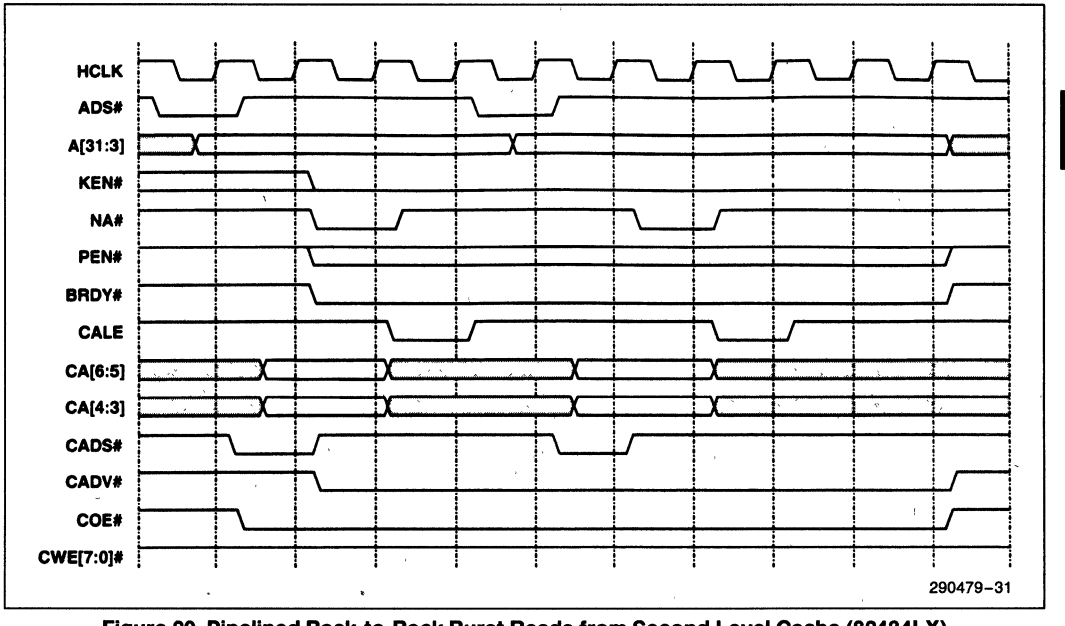


Figure 29. Pipelined Back-to-Back Burst Reads from Second Level Cache (82434LX)

2

5.1.3.2 Burst Write (82434LX)

A burst write cycle is used to write back a line from the first level cache to either the second level cache or DRAM. A burst write cycle from the first level cache to the second level cache is shown in Figure 30.

The Pentium processor always writes back lines starting with the low order Qword advancing to the high order Qword. CADS# is asserted in the second clock. CWE[7:0]# and BRDY# are asserted in the third clock. CADV# assertion is delayed by one clock relative to the burst read cycle. HIG[4:0] are driven to PCMWQ (Post CPU-to-Memory Write Buffer Qword) only when the PCMC is programmed for a write-through write policy. When programmed for write-back mode, the modified bit associated with the line is set within the PCMC. The single write is very similar to the first write in a burst write. CADS# is asserted in the second clock. BRDY# and CWE[7:0]# are asserted in the third clock. A burst read cycle followed by a pipelined single write cycle is depicted in Figure 31.

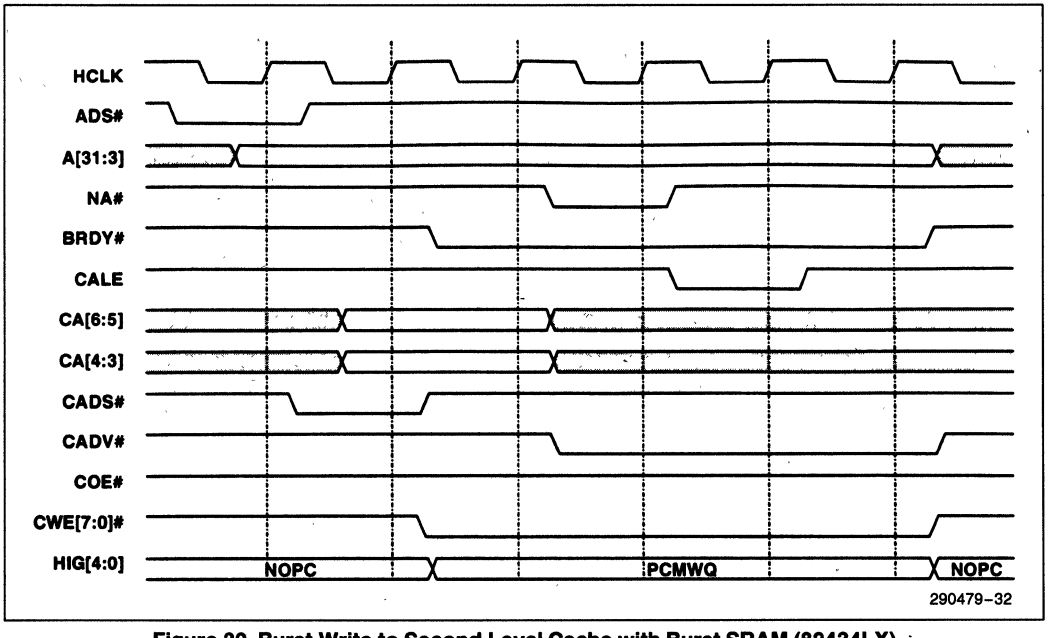


Figure 30. Burst Write to Second Level Cache with Burst SRAM (82434LX)

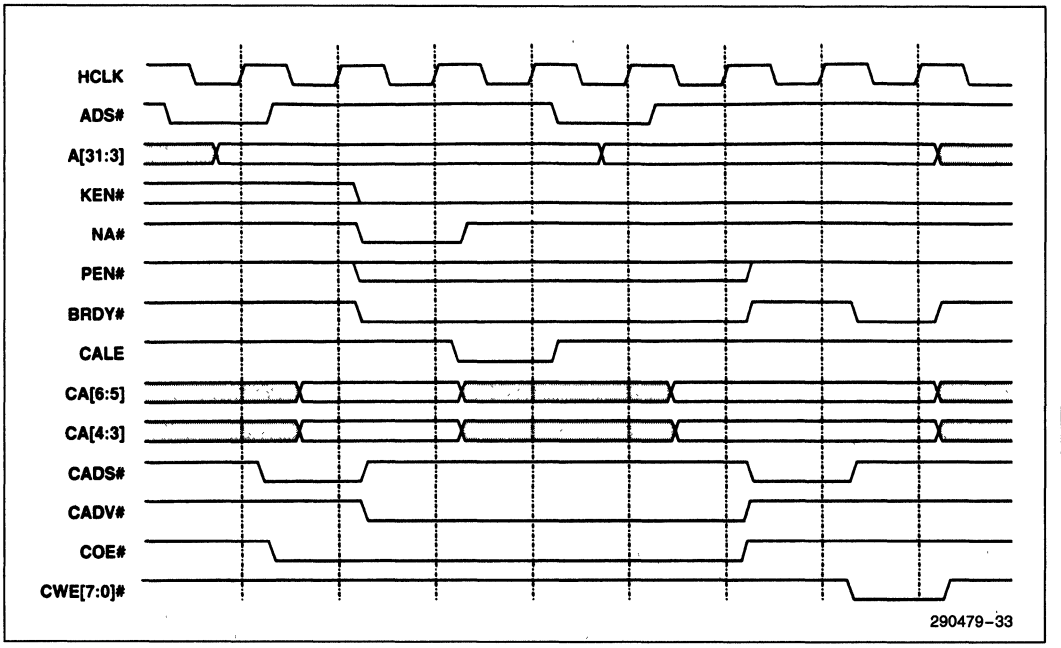


Figure 31. Burst Read Followed by Pipelined Single Write Cycle with Burst SRAM (82434LX)

5.1.3.3 Cache Line Fill (82434LX)

If the CPU issues a memory read cycle to cacheable memory which does not hit in the second level cache, a cache line fill occurs. Figure 32 depicts a first and second level cache line fill with burst SRAMs.

Figure 33 depicts a CPU read cycle which forces a write-back in the second level cache.

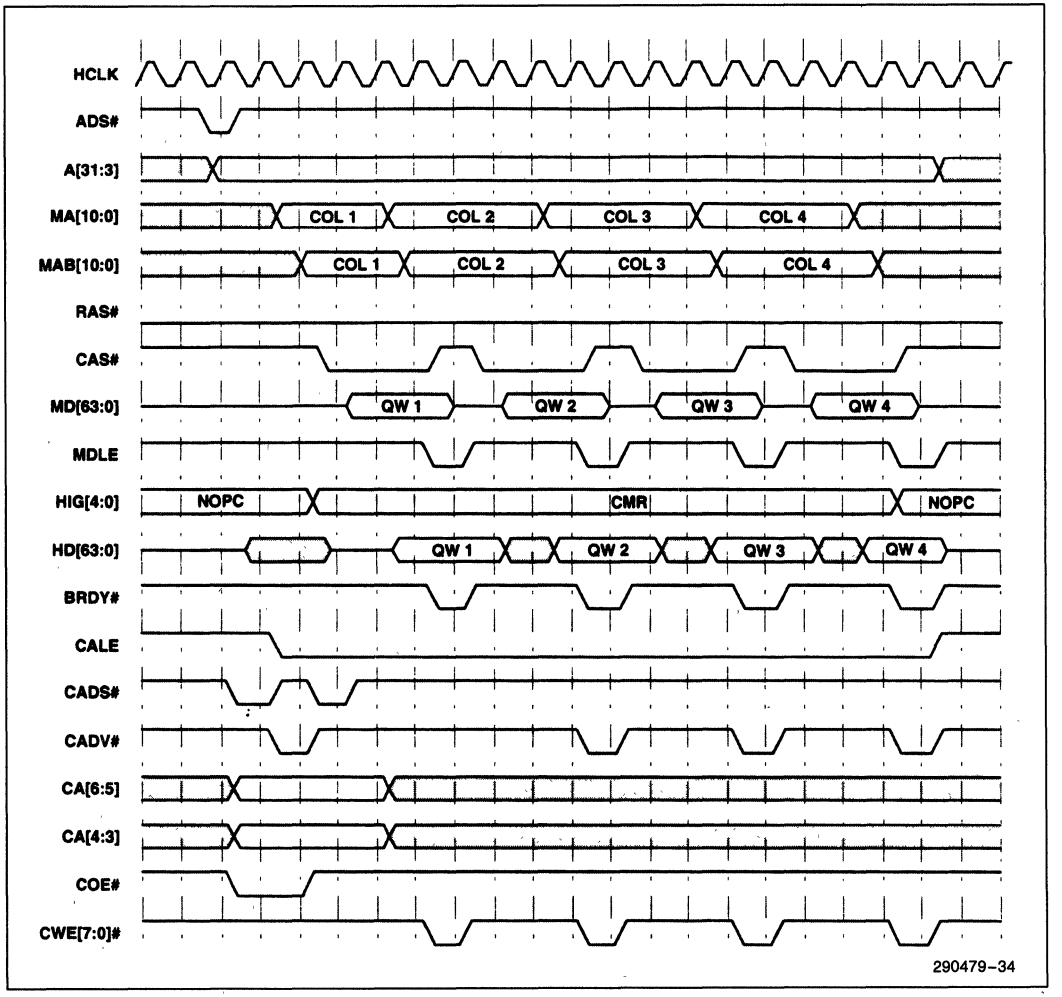


Figure 32. Cache Line Fill with Burst SRAM, DRAM Page Hit, 7-4-4-4 Timing (82434LX)

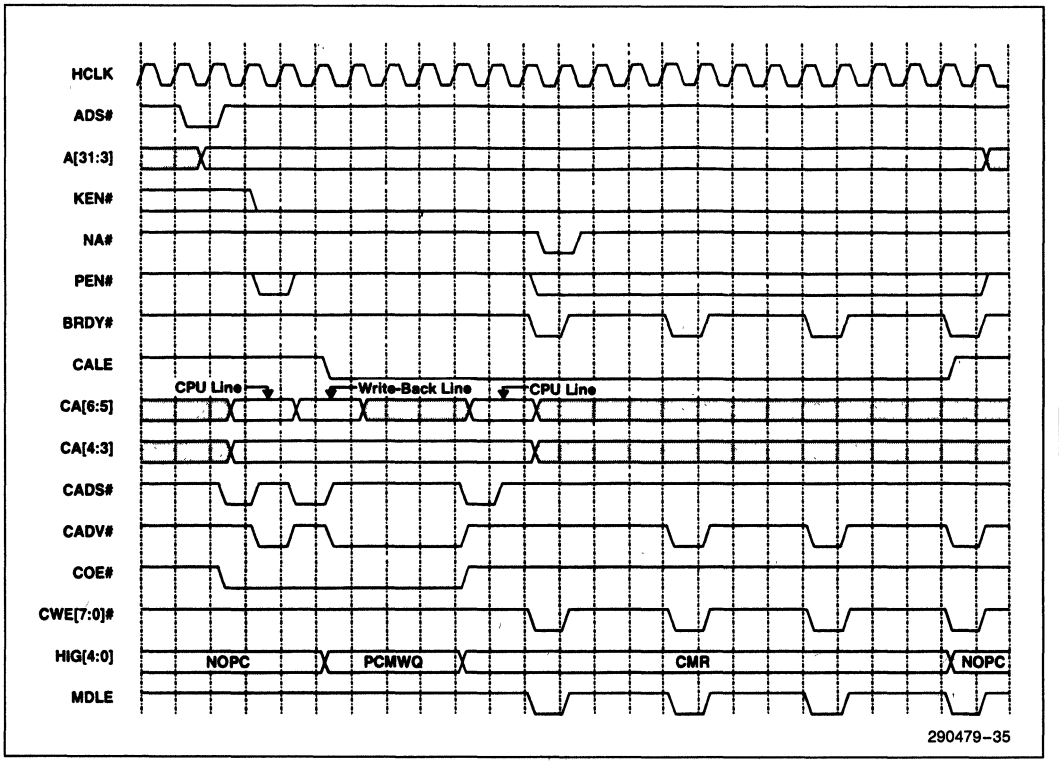


Figure 33. CPU Cache Read Miss, Write-Back, Line Fill with Burst SRAM (82434LX)

The CPU issues a memory read cycle which misses in the second level cache. In this instance, a modified line in the second level cache must be written back to main memory before the new line can be filled into the cache. The PCMC inspects the valid and modified bits for each of the lines within the addressed sector and writes back only the valid lines within the sector that are marked modified. CA[6:5] are used to count through the lines within the addressed sector. When two or more lines must be written back to main memory, CA[6:5] count in the direction from line 0 to line 3 after each line is written back. Figure 29 depicts the case of just one

of the lines in a sector being written back to main memory. In this case, the entire line can be posted in the CPU-to-Memory Posted Write Buffer by driving the HIG[4:0] lines to PCMWQ as each Qword is read from the cache. At the same time, the required DRAM read cycle is beginning. After the de-allocated line is written into the posted write buffer, the HIG[4:0] lines are driven to CMR (CPU Memory Read) to allow data to propagate from the DRAM data lines to the CPU data lines. Figure 29 assumes that the read from DRAM is a page hit and thus the first Qword is already read from the DRAMs when the transfer from cache to the CPU to Memory posting buffer is complete. The rest of the DRAM cycle completes at a -4-4-4 rate. CADV# is asserted with the last three BRDY# assertions. CMR is driven on the HIG[4:0] lines throughout the DRAM read portion of the cycle. Upon the fourth assertion of BRDY# the HIG[4:0] lines change to NOPC.

PEN# is asserted as shown if the MCHK DRAM/L2 Cache Data Parity Error bit in the Error Command Register (offset 70h) is set. If the second level cache supports parity, PEN# is always asserted during CPU read cycles in clock 3 in case the cycle hits in the cache.

If more than one line must be written back to main memory, the PCMC fills the CPU-to-Main Memory Posted Write Buffer and loads another Qword into the buffer as each Qword write completes into main memory. The writes into DRAM proceed as page hit write cycles from one line to the next, completing at a rate of X-4-4-4-5-4-4-4-5-4-4-4 for a three line write-back when programmed for X-4-4-4 DRAM write timing or X-3-3-3-4-3-3-3-4-3-3-3 when programmed for X-3-3-3 DRAM write timing. All modified lines except for the last one to be written back to memory are posted and retired to memory before the DRAM read cycle begins. The last line to be written back is posted as the DRAM read cycle begins. Thus, the read data is returned to the CPU before the last line is retired to memory.

The line which was written into the second level cache is marked valid and unmodified by the PCMC. All the other lines in the block are marked invalid. A subsequent CPU read cycle which hits the same sector (but a different line) in the second level cache results in a line fill without any write-back.

5.1.4 SNOOP CYCLES

Snoop cycles are the same for the 82434LX and 82434NX. The inquire cycle is used to probe the first level and second level caches when a PCI master attempts to access main memory. This is done to maintain coherency between the first and second level caches and main memory. When a PCI master first attempts to access main memory a snoop request is generated inside the PCMC. The PCMC supports up to two outstanding cycles on the CPU address bus at a time. Outstanding cycles include both CPU initiated cycles and snoop cycles. Thus, if the Pentium processor pipelines a second cycle onto the host address bus, the PCMC will not issue a snoop cycle until the first CPU cycle terminates. If the PCMC were to initiate a snoop cycle before the first CPU cycle were complete then for a brief period of time, three cycles would be outstanding. Thus, a snoop request is serviced with a snoop cycle only when either no cycle is outstanding on the CPU bus or one cycle is outstanding.

Snoop cycles are performed by driving the PCI master address onto the CPU address bus and asserting EADS#. The Pentium processor then performs a tag lookup to determine if the addressed memory is in the first level cache. At the same time the PCMC performs an internal tag lookup to determine if the addressed memory is in the second level cache. Table 7 describes how a PCI master read from main memory is serviced by the PCMC.

Table 7. Data Transfers for PCI Master Reads from Main Memory

Snoop Result		Action
First Level Cache	**Second Level Cache**	
Miss	Miss	Data is transferred from DRAM to PCI.
Miss	Hit Unmodified Line	Data is transferred directly from second level cache to PCI. The line remains valid and unmodified in the second level cache.
Miss	Hit Modified Line	Data is transferred directly from second level cache to PCI. Line remains valid and modified in the second level cache. The line is not written to DRAM.
Hit Unmodified Line	Miss	Data is transferred from DRAM to PCI.
Hit Unmodified Line	Hit Unmodified Line	Data is transferred directly from second level cache to PCI. The line remains valid and unmodified in the second level cache.
Hit Unmodified Line	Hit Modified Line	Data is transferred directly from second level cache to PCI. Line remains valid and modified in the second level cache. The line is not written to DRAM.
Hit Modified Line	Miss	A write-back from first level cache occurs. The data is sent to both PCI and the CPU-to-Memory Posted Write Buffer. The CPU-to-Memory Posted Write Buffer is then written to memory.
Hit Modified Line	Hit Unmodified Line	A write-back from first level cache occurs. The data is posted to PCI and written into the second level cache. When the second level cache is in write-back mode, the line is marked modified and is not written to DRAM. When the second level cache is in write-through mode, the line is posted and then written to DRAM.
Hit Modified Line	Hit Modified Line	A write-back from first level cache occurs. The data is posted to PCI and written into the second level cache. The line is not written to DRAM. This scenario can only occur when the second level cache is in write-back mode.

PCI master write cycles never result in a write directly into the second level cache. A snoop hit to a modified line in either the first level or second level cache results in a write-back of the line to main memory. The line is invalidated and the PCI write to main memory occurs after the write-back completes. The other lines in the sector are not written back to main memory or invalidated. A PCI master write snoop hit to an unmodified line in either the first level or second level cache results in the line being invalidated. Table 8 describes the actions taken by the PCMC when a PCI master writes to main memory.

Table 8. Data Transfers for PCI Master Writes to Main Memory

Snoop Result		Action
First Level Cache	**Second Level Cache**	
Miss	Miss	The PCI master write data is transferred from PCI to DRAM.
Miss	Hit Unmodified Line	The PCI master write data is transferred from PCI to DRAM. The line is invalidated in the second level cache.
Miss	Hit Modified Line	A write-back from second level cache to DRAM occurs. The PCI master write data is then written to DRAM. The line is invalidated in the second level cache.
Hit Unmodified Line	Miss	The first level cache line is invalidated. The PCI master write data is written to DRAM.
Hit Unmodified Line	Hit Unmodified Line	The line is invalidated in both the first level and second level caches. The PCI master write data is written to DRAM.
Hit Unmodified Line	Hit Modified Line	The first level cache line is invalidated. The second level cache line is written back to main memory and invalidated. The PCI master write data is then written to DRAM.
Hit Modified Line	Miss	The first level cache line is written back to DRAM and invalidated. The PCI master write data is then written to DRAM.
Hit Modified Line	Hit Unmodified Line	The first level cache line is written back to DRAM and invalidated. The second level cache line is invalidated. The PCI master write data is then written to DRAM.
Hit Modified Line	Hit Modified Line	The first level cache line is written back to DRAM and invalidated. The second level cache line is invalidated. The PCI master write data is then written to DRAM.

A snoop hit results in one of three transfers; a write-back from the first level cache posted to the LBXs, a write-back from the second level cache posted to the LBXs or a write-back from the first level cache posted to the LBXs and written to the second level cache. A snoop cycle that does not result in a write-back is depicted in Figure 34.

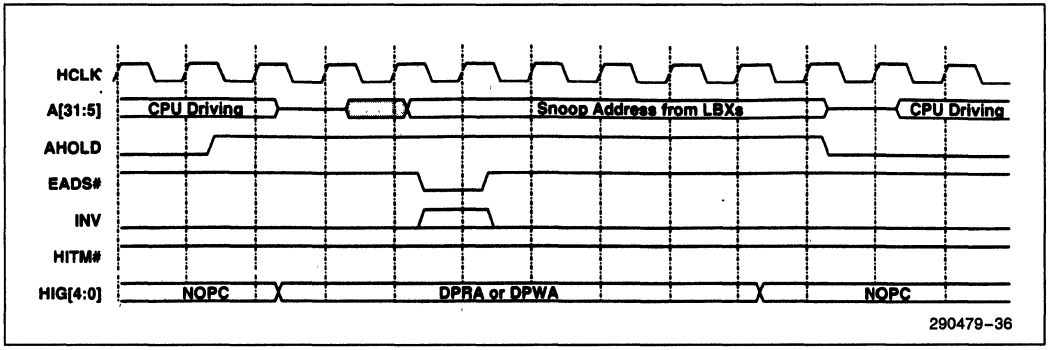


Figure 34. Snoop Hit to Unmodified Line in First Level Cache or Snoop Miss

The PCMC begins to service the snoop request by asserting AHOLD, causing the Pentium processor to tri-state the address bus in the clock after assertion. In the case of a PCI master read cycle, the PCMC drives the DPRA (Drive PCI Read Address) command onto the HIG[4:0] lines causing the LBXs to drive the PCI address onto the host address bus. For a write cycle, the PCMC drives the DPWA (Drive PCI Write Address to CPU Address Bus) command on the HIG[4:0] lines, also causing the LBXs to begin driving the host address bus. The PCMC then asserts EADS#, initiating the snoop cycle to the CPU. The INV signal is asserted by the PCMC only during snoops due to PCI master writes. INV remains negated during snoops due to PCI master reads. If the snoop results in a hit to a modified line in the first level cache, the Pentium processor asserts HITM#. The PCMC samples the HITM# signal two clocks after the CPU samples EADS# asserted to determine if the snoop hit in the first level cache. By this time the PCMC has completed an internal tag lookup to determine if the line is in the second level cache. Since this snoop does not result in a write-back, the NOPC command is driven on the HIG[4:0] lines, causing the LBXs to tri-state the address bus. The sequence ends with AHOLD negation.

If the Pentium processor asserts ADS# in the same clock as the PCMC asserts AHOLD, the PCMC will assert BOFF# in two cases. First, if the snoop cycle hits a modified line in the first level cache, the PCMC will assert BOFF# for 1 HCLK to re-order the write-back around the currently sending cycle. Second, if the snoop requires a write-back from the second level cache, the PCMC will assert BOFF# to enable the write-back from the secondary cache SRAMs.

Figure 35 depicts a snoop hit to a modified line in the first level cache due to a PCI master memory read cycle.

The snoop cycle begins when the PCMC asserts AHOLD causing the CPU to tri-state the address bus. The PCMC drives the DPRA (Drive PCI Read Address) command on to the HIG[4:0] lines causing the LBXs to drive the PCI address onto the host address bus. The PCMC then asserts EADS#, initiating the snoop to the first level cache. INV is not asserted since this is a PCI master read cycle. INV is only asserted with EADS# when the snoop cycle is in response to a PCI master write cycle. As the CPU is sampling EADS# asserted, the PCMC latches the address. Two clocks later, the PCMC completes the

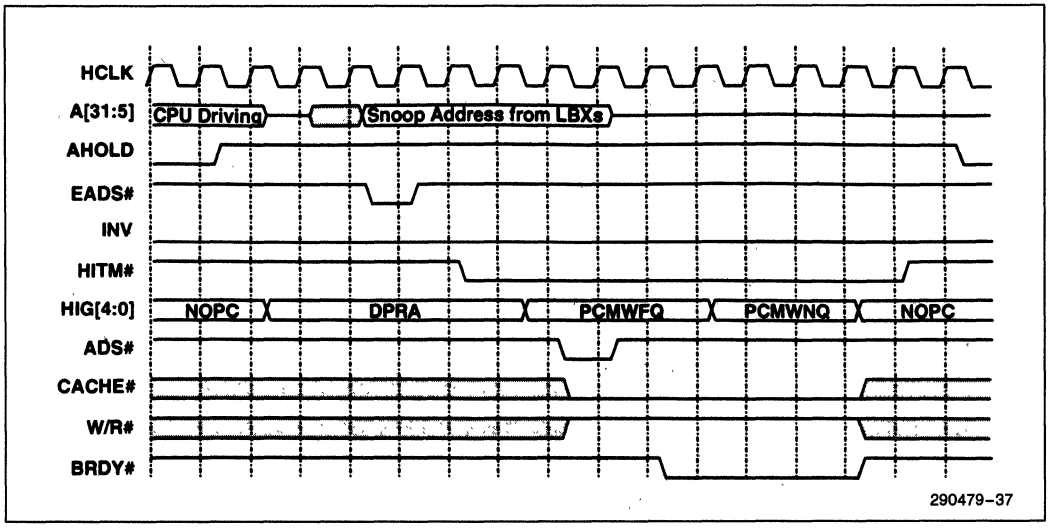


Figure 35. Snoop Hit to Modified Line in First Level Cache, Post Memory and PCI

internal tag lookup to determine if the line is in the second level cache. In this instance, the snoop hits a modified line in the first level cache and misses in the second level cache. Thus, the second level cache is not involved in the write-back cycle. The PCMC allows the LBXs to stop driving the address lines by driving NOPC command on the HIG[4:0] lines. The CPU then drives the write-back cycle onto the bus by asserting ADS# and driving the write-back data on the data lines even though AHOLD is still asserted. The write-back into the LBX buffers occurs at a rate of 3-1-1-1. The PCMC drives PCMWFQ on the HIG[4:0] lines for one clock causing the write data to be posted to both PCI and main memory. For the next three clocks, the HIG[4:0] lines are driven to PCMWNQ, posting the final three Qwords to both PCI and main memory.

A similar transfer from first level cache to the LBXs occurs when a snoop due to a PCI master write hits a modified line in the first level cache. In this case, the write-back is transferred to the CPU-to-Memory Posted Write Buffer. If the line is in the second level cache, it is invalidated. The cycle is similar to the snoop cycle shown above with two exceptions. The PCMC drives the DPWA command on the HIG[4:0] lines instead of the DPRA command. During the four clocks where the PCMC drives BRDY# active to the CPU, it also drives PCMWQ on the HIG[4:0] lines, causing the write to be posted to main memory.

In both of the above cases where a write-back from the first level cache is required, AHOLD is asserted until the write-back is complete. If the CPU has begun a read cycle directed to PCI and the snoop results in a hit to a modified line in the first level cache, BOFF# is asserted for one clock to abort the CPU read cycle and re-order the write-back cycle before the read cycle.

When a PCI master read or write cycle hits a modified line in the second level cache and either misses in the first level cache or hits an unmodified line in the first level cache, a write-back from the second level cache to the LBXs occurs. When a PCI master write snoop hits an unmodified line in the second level cache and either misses in the first level cache or hits an unmodified line in the first level cache, no data transfer from the second level cache occurs. The line is simply invalidated. In the case of a PCI master write cycle, the line is invalidated in both the first level and second level caches. In the case of a PCI master memory read cycle, neither cache is invalidated. A PCI master read from main memory which hits either a modified or unmodified line in the second level cache is shown in Figure 36.

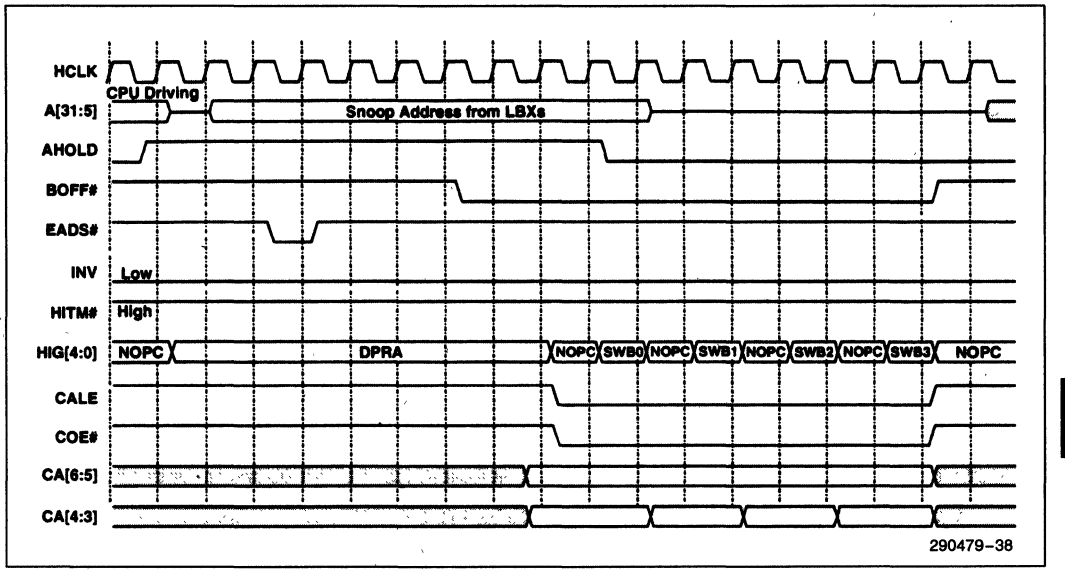


Figure 36. Snoop Hit to Modified Line in Second Level Cache, Store in PCI Read Prefetch Buffer

The snoop cycle begins with the PCMC asserting AHOLD, causing the CPU to tri-state the host address bus. The PCMC drives the DPRA command enabling the LBXs to drive the snoop address onto the host address bus. The PCMC asserts EADS#. INV is not asserted in this case since the snoop cycle is in response to a PCI master read cycle. If the snoop were in response to a PCI master write cycle then INV would be asserted with EADS#. Two clocks after the CPU samples EADS# active, the PCMC completes the internal tag lookup. In this case the snoop hit either an unmodified line or a modified line in the second level cache. Since HITM# is inactive, the snoop did not hit in the first level cache. The PCMC then schedules a read from the second level cache to be written to the LBXs. When the CPU burst cycle completes the PCMC negates the control signals to the second level cache and asserts CALE opening the cache address latch and allowing the snoop address to flow through to the SRAMs. The second level cache executes a read sequence which completes at 3-2-2-2 in the case of standard SRAMs and 3-1-1-1 in the case of burst SRAMs. During all snoop cycles where a write-back from the second level cache is required, BOFF# is asserted throughout the write-back cycle. This prevents the deadlock that would occur if the CPU is in the middle of a non-postable write and the data bus is required for the second level cache write-back.

When using burst SRAMs, the read from the SRAMs follows the Pentium processor burst order. However, the memory to PCI read prefetch buffer in the LBXs is organized as a FIFO and cannot accept data out of order. The SWB0, SWB1, SWB2 and SWB3 commands are used to write data into the buffer in ascending order. In the above example, the PCI master requests a data item which hits Qword 0 in the cache, thus CA[4:3] count through the following sequence: 0, 1, 2, 3 (00, 01, 10, 11). If the PCI mas-

ter requests a data item that hits Qword 1, the SWB0 command is sent via the HIG[4:0] lines to store Qword 1in the first buffer location. The next read from the cache is not in ascending order, thus a NOPC is sent on the HIG[4:0] lines. This Qword is not posted in the buffer. The next read from the cache is to Qword 3. SWB2 is sent on the HIG[4:0] lines. The final read from the cache is Qword 2. SWB1 is sent on the HIG[4:0] lines. Thus, Qword 1 is placed in entry 0 in the buffer, Qword 2 is placed in entry 1 in the buffer and Qword 3 is placed in entry 2 in the buffer. The ordering between the Qwords read from the cache and the HIG[4:0] commands when using burst SRAMs is summarized in Table 9.

Table 9. HIG[4:0] Command Sequence for Second Level Cache to PCI Master Read Prefetch Buffer Transfer

Burst Order from Cache	HIG[4:0] Command Sequence
0, 1, 2, 3	SWB0, SWB1, SWB2, SWB3
1, 0, 3, 2	SWB0, NOPC, SWB2, SWB1
2, 3, 0, 1	SWB0, SWB1, NOPC, NOPC
3, 2, 1, 0	SWB0, NOPC, NOPC, NOPC

When using standard asynchronous SRAMs, the read from the SRAMs occurs in a linear burst order. Thus, CAA[4:3] and CAB[4:3] count in a linear burst order and the Store Write Buffer commands are sent in linear order. The burst ends at the cache line boundary and does not wrap around and continue with the beginning of the cache line.

A PCI master write cycle which hits a modified line in the second level cache and either hits an unmodified line in the first level cache or misses in the first level cache will also cause a transfer from the second level cache to the LBXs. In this case, the read from the SRAMs is posted to main memory and the line is invalidated in the second level cache. The cycle would differ only slightly from the above cycle. INV would be asserted with EADS#. Instead of the DPRA command, the PCMC would use the DPWA command to drive the snoop address onto the host address bus. The write would be posted to the DRAM, thus the PCMC would drive the PCMWQ command on the HIG[4:0] lines to post the write to DRAM.

A snoop cycle can result in a write-back from the first level cache to both the second level and LBXs in the case of a PCI master read cycle which hits a modified line in the first level cache and hits either a modified or unmodified line in the second level cache. The line is written to both the second level cache and the memory to PCI read prefetch buffer. The cycle is shown in Figure 37.

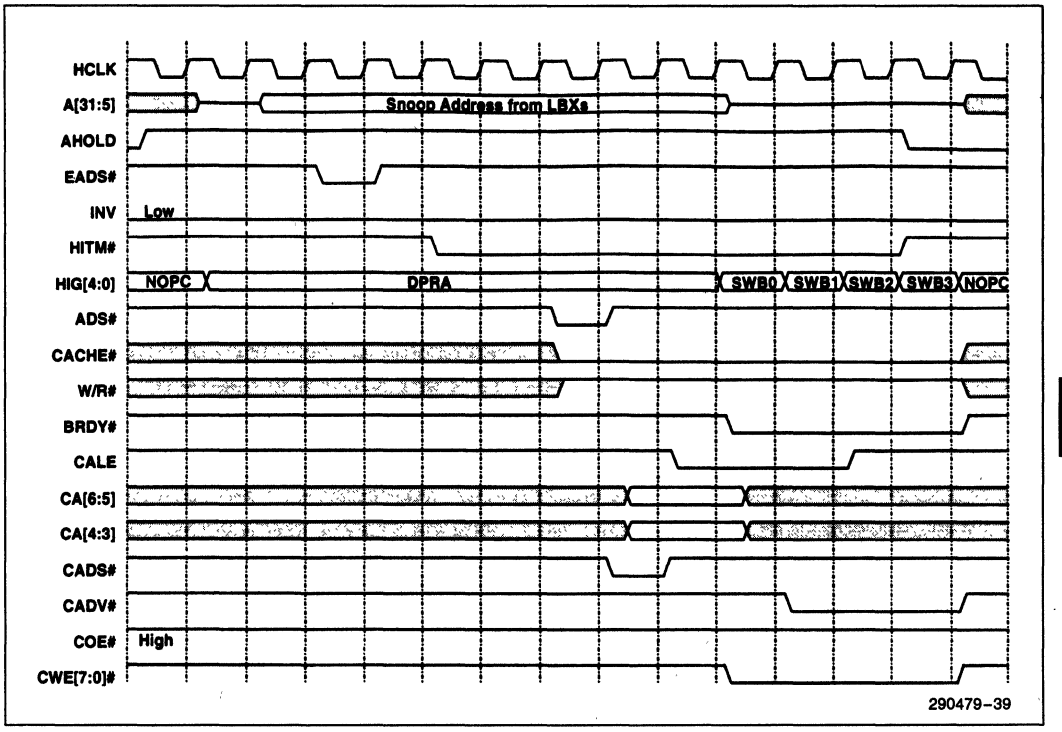


Figure 37. Snoop Hit to Modified Line in First Level Cache, Write-Back from First Level Cache to Second Level Cache and Send to PCI

This cycle is shown for the case of a second level cache with burst SRAMs. In this case, as it completes the second level cache tag lookup, the PCMC samples HITM# active. The write-back is written to the second level cache and simultaneously stored in the memory to PCI prefetch buffer. In the case shown in Figure 33, the PCI master requests a data item which is contained in Qword 0 of the cache line. Note that a write-back from the first level cache always starts with Qword 0 and finishes with Qword 3. Thus the HIG[4:0] lines are sequenced through the following order: SWB0, SWB1, SWB2, SWB3. If the PCI master requests a data item which is contained in Qword 1, the HIG[4:0] lines sequence through the following order: NOPC, SWB0, SWB1, SWB2. If the PCI master requests a data item which is contained in Qword 2, the HIG[4:0] lines sequence through the following order: NOPC, NOPC, SWB0, SWB1. If the PCI master requests a data item which is contained in Qword 3, the HIG[4:0] lines sequence through the following order: NOPC, NOPC, NOPC, SWB0. AHOLD is negated after the write-back cycle is complete.

If the CPU has begun a read cycle directed to PCI and the snoop results in a hit to a modified line in the first level cache, BOFF# is asserted for one clock to abort the CPU read cycle and re-order the write-back cycle before the pending read cycle.

5.1.5 FLUSH, FLUSH ACKNOWLEDGE AND WRITE-BACK SPECIAL CYCLES

There are three special cycles that affect the second level cache, flush, flush acknowledge, and write-back. If the processor executes an INVD instruction, it will invalidate all unmodified first level cache lines and issue a flush special cycle. If the processor executes a WBINVD instruction, it will write back all modified first level cache lines, invalidate the first level cache, and issue a write-back special cycle followed by a flush special cycle. If the Pentium processor FLUSH# pin is asserted, the CPU will write-back all modified first level cache lines, invalidate the first level cache, and issue a flush acknowledge special cycle.

The second level cache behaves the same way in response to the flush special cycle and flush acknowledge special cycle. Each tag is read and the valid and modified bits are examined. If the line is both valid and modified it is written back to main memory and the valid bit for that line is reset. All valid and unmodified lines are simply marked invalid. The PCMC advances to the next tag when all lines within the current sector have been examined. BRDY# is returned to the Pentium processor after all modified lines in the second level cache have been written back to main memory and all of the valid bits for the second level cache are reset. The sequence of write-back cycles will only be interrupted to service a PCI master cycle.

The write-back special cycle is ignored by the PCMC because all modified lines will be written back to main memory by the following flush special cycle. Upon decoding a write-back special cycle, the PCMC simply returns BRDY# to the Pentium processor.

5.2 82434NX Cache

The 82434NX PCMC integrates a high performance write-back second level cache controller, tag RAM and a full first and second level cache coherency mechanism. The cache is either 256 KBytes or 512 KBytes using either synchronous burst SRAMs or standard asynchronous SRAMs. Parity on the data SRAMs is optional. The cache uses a write-back write policy. Write-through mode is not supported.

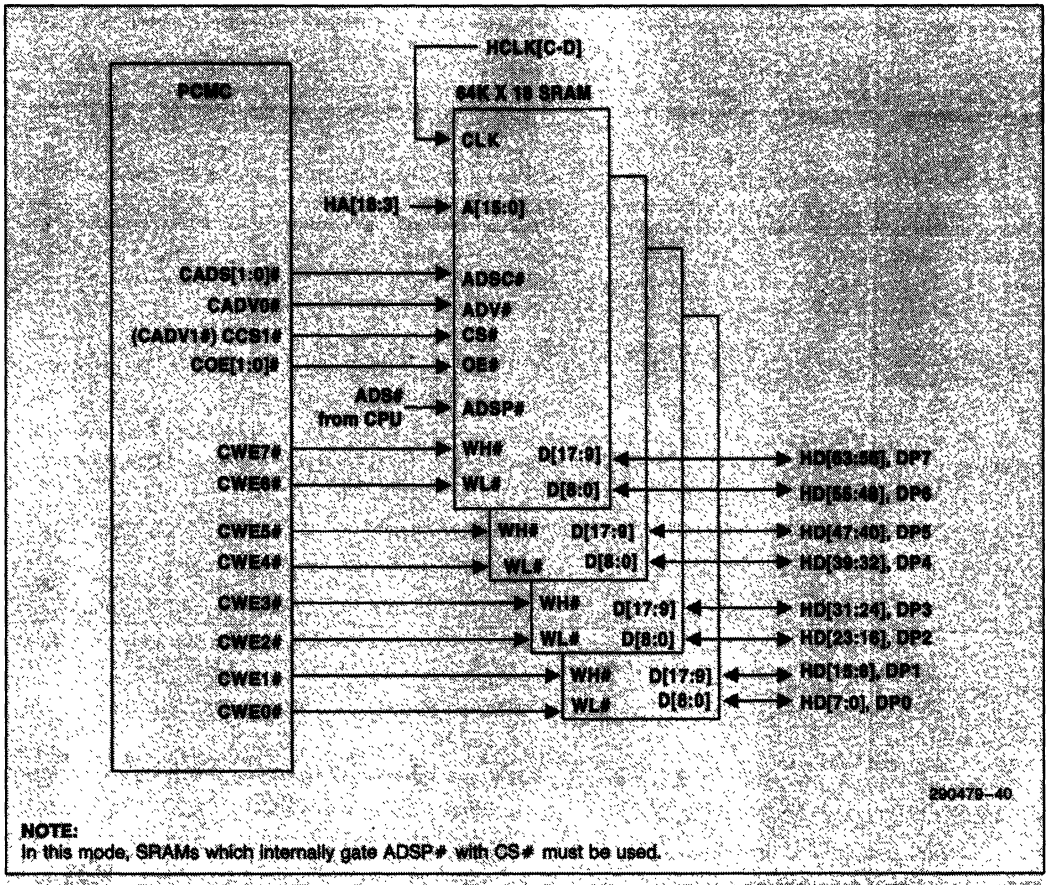
The 82434NX PCMC supports a direct mapped secondary cache. The PCMC contains 4096 tags. Each tag represents a sector in the cache. If the cache is 512 KB, each sector contains four cache lines. If the cache is 256 KB, each sector contains two cache lines. *Valid* and *Modified* bits are kept on a per line basis. The 82434NX Tag RAM is 1 bit wider than the 82434LX Tag RAM.

The PCMC can be configured to cache main memory on read cycles even when CACHE# is not asserted. When bit 4 in the Secondary Cache Control Register (offset 52h) is set to 1, all accesses to main memory, except those to SMM memory or any range marked non-cacheable via the PAM registers, are cached in the secondary cache. Accesses with CACHE# asserted result in a line fill in both the first and second level cache while accesses with CACHE# negated result in a line fill only in the second level cache. When bit 4 in the SCC Register is set to 0, only access with CACHE# asserted can generate a first and second level cache line fill.

When a Halt or Stop Grant Special Cycle is detected from the CPU, the 82434NX PCMC places the second level cache into the low power stand-by mode by deselecting the SRAMs and then generates the corresponding special cycle on PCI. (i.e., if the CPU cycle was a halt special cycle then the PCMC generates a halt special cycle on PCI and if the CPU cycle is a stop grant special cycle the PCMC generates a stop grant special cycle on PCI).

When a burst SRAM secondary cache is implemented, bit 2 of the Secondary Cache Control Register (offset 52h) is used to select between 82434LX SRAM connectivity and the new 82434NX SRAM connectivity. When set to 0, the secondary cache interface is in 82430-compatible mode. (i.e., the four low order address lines on the SRAMs are connected to CAA/B[6:3] on the PCMC. When set to 1, second level cache stand-by is enabled and no latch is used between the host CPU address lines and the SRAM address lines. All of the SRAM address lines are then connected directly to the CPU address lines. Write-back addresses are driven by the PCMC over the host address lines. When a standard SRAM secondary cache is implemented, bit 2 of the Secondary Cache Control Register (offset 52h) is used to enable second level cache stand-by. The default value of this bit is 0.

Figure 38 and Figure 41 show the connections between the PCMC and the external cache data SRAMs and latch for the case of an asynchronous SRAM cache.



NOTE:
In this mode, SRAMs which internally gate ADSP# with CS# must be used.

Figure 38. 512 KByte Secondary Cache, Synchronous Burst SRAM (82434NX)

2

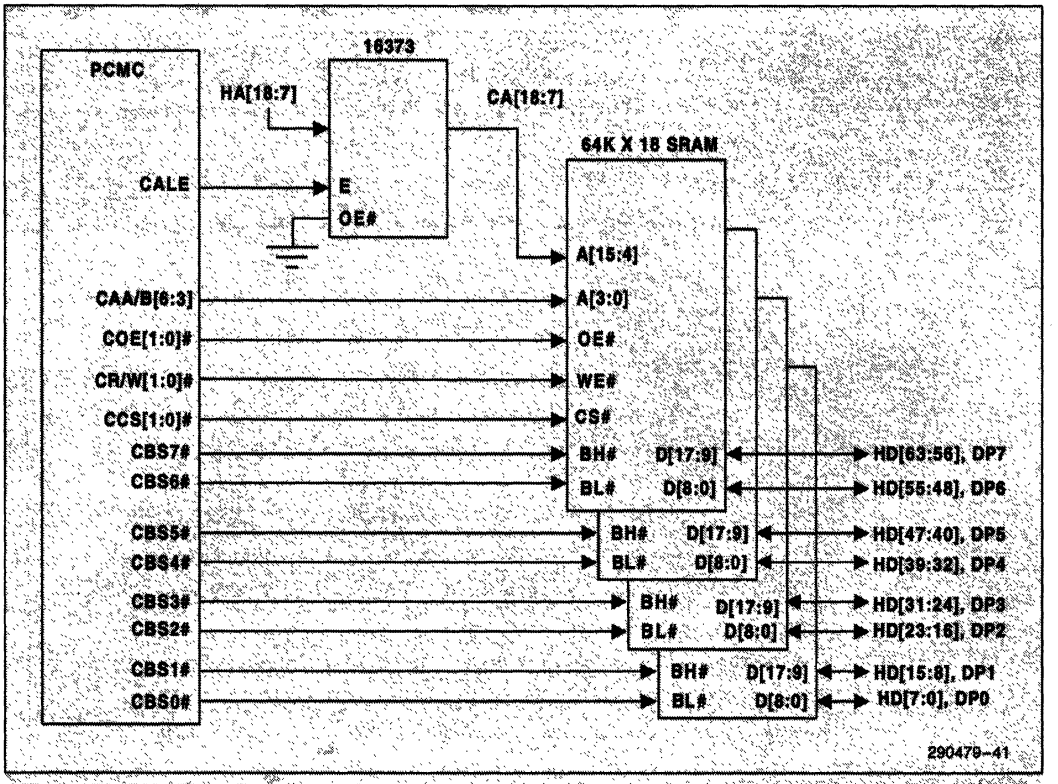


Figure 39. 512 KByte Secondary Cache, Standard Dual-Byte-Select (Asynch) SRAM, 50, 60 & 66 MHz

Figure 38 depicts the PCMC connections to a 512 KByte burst SRAM secondary cache when the PCMC is configured for 50, 60, or 66 MHz operation. Host address lines HA[18:3] are connected directly to the SRAM address lines, A[15:0]. ADS# from the CPU is connected to ADSP# on the SRAMs. CADV0# implements the address advance (ADV#) functionality. A new signal, CCS#, is multiplexed onto the CADV1# pin. When bit 2 in the SCC register is set to 1, SRAMs containing logic which gates ADSP# with CS# must be used. When negated, CCS# prevents the SRAMs from latching a new address due to a pipelined ADS# from the CPU during cache line fills. Note that, unlike the burst SRAM configuration with the 82430 PCIset, no external latch is used between the CPU address bus and the SRAM address lines. The SRAM Connectivity bit (bit 2) in the Secondary Cache Control register (offset 52h) must be set to 1 when using this cache configuration.

If the tag lookup results in a miss in the cache and the sector to be replaced contains one or more modified lines, the PCMC drives the write-back address from the A[18:3] lines on the host bus. Although not used in the write-back, A[31:19] (or A[31:18] in the case of a 256 KB cache) are driven to valid logic levels by the PCMC.

Figure 39 depicts the 82434NX PCMC connections to a 512 KByte standard asynchronous SRAM secondary cache. Figure 40 depicts the 82434NX connections to a 256 KByte asynchronous SRAM secondary cache. Host address lines HA[18:7] are driven through an external latch to form the upper SRAM address lines, CA[18:7]. CA[6:3] are driven from the PCMC. Figure 41 depicts the 82434NX PCMC connections to a 512 KByte standard SRAM secondary cache with dual-write-enable SRAMs.

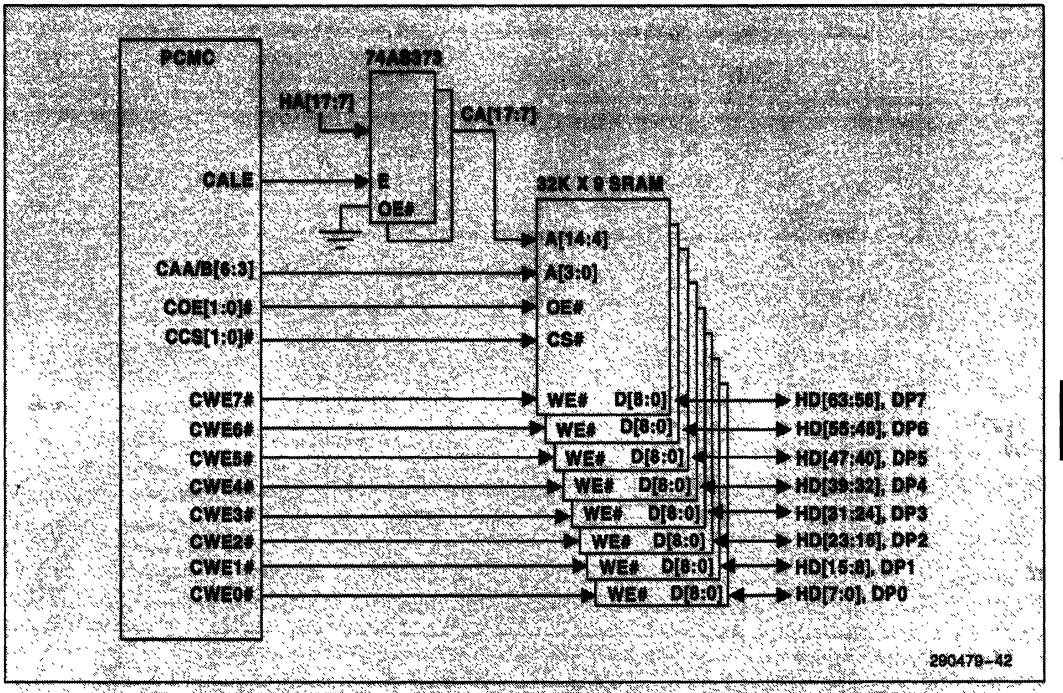


Figure 40. 82434NX Connections to 256 KByte Cache with Standard SRAM

2

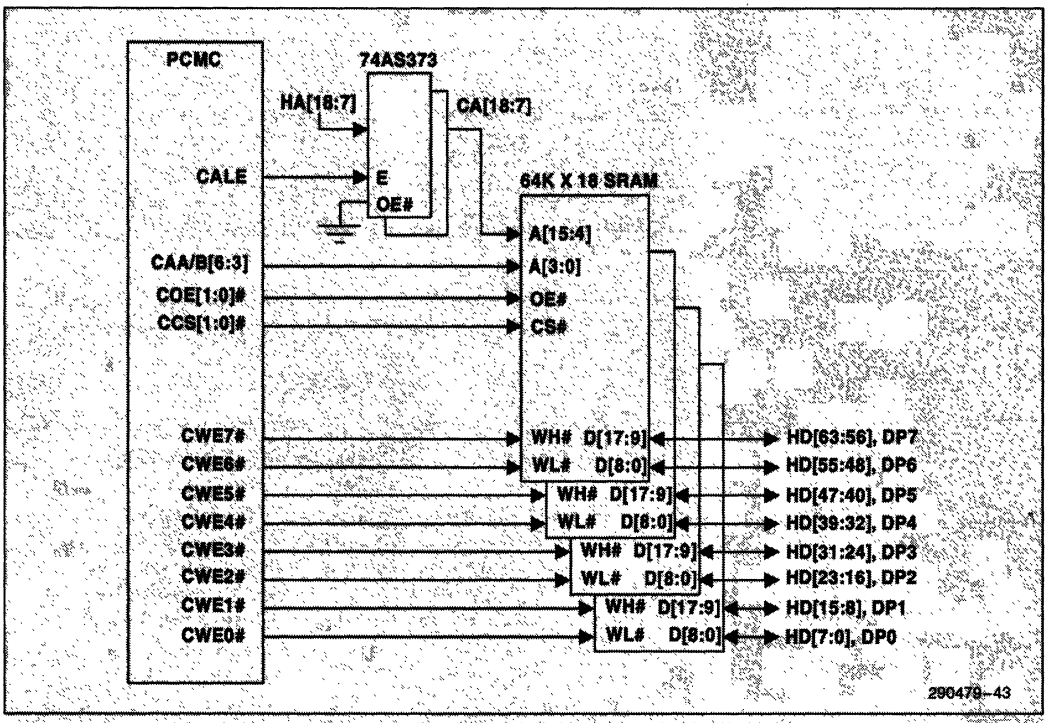


Figure 41. 82434NX Connections to 512 KByte Cache with Standard SRAM

5.2.1 CYCLE LATENCY SUMMARY (82434NX)

Table 10 and Table 11 summarize the clock latencies for CPU memory cycles which hit in the secondary cache.

Table 10. Secondary Cache Latencies with Synchronous Burst SRAM

Cycle Type	50, 60 and 66 MHz
Burst Read	3-1-1-1
Burst Write	3-1-1-1
Single Read	3
Single Write	3
Pipelined Back-to-Back Burst Reads	3-1-1-1-1-1-1-1
Burst Read Followed by Pipelined Write	3-1-1-1-2

Table 11. Secondary Cache Latencies with Standard Asynchronous SRAM (82434NX)

Cycle Type	50, 60 and 66 MHz
Burst Read	3-2-2-2
Burst Write	4-2-2-2
Single Read	3
Single Write	4
Pipelined Back-to-Back Burst Reads	3-2-2-2-3-2-2-2
Burst Read Followed by Pipelined Write	3-2-2-2-4

The 60 MHz and 66 MHz asynchronous SRAM latencies require 15 ns and 12 ns SRAMs, respectively. The 82434NX PCMC supports asynchronous SRAMs at 50 MHz. The 50 MHz (1 wait-state) timings require 20 ns SRAMs. The burst SRAMs speeds for 66 MHz, 60 MHz and 50 MHz operation are 8 ns, 9 ns, and 13 ns clock-to-output valid into a 0 pF test load. The SRAM access times listed in this paragraph are recommendations. Actual access time requirements are a function of system board layout and routing and should be validated with electrical simulation.

5.2.2 STANDARD SRAM CACHE CYCLES (82434NX)

At 50, 60 and 66 MHz, the timing of the second level cache interface with standard asynchronous SRAMs is identical to the timing in the 82430LX PCIset. Compared to the 82434LX second level cache, one additional connection can be made from the PCMC to the SRAMs. The CCS[1:0]# pins, in the case of asynchronous SRAMs, are multiplexed onto the CADV[1:0]# pins. These are then connected to the SRAM CS# pins. The two copies are functionally identical. The two copies are provided for timing reasons. These pins allow the PCMC to deselect the SRAMs, putting them into standby mode. When a halt special cycle or a stop grant special cycle is detected from the CPU, the PCMC negates CCS[1:0]#, placing the SRAMs into the low power standby mode. The PCMC then generates a halt or stop grant special cycle on PCI.

5.2.3 SECOND LEVEL CACHE STANDBY

When the PCMC detects a halt or stop grant special cycle from the CPU, it first places the second level cache into the low power stand-by mode by deselecting the SRAMs and then generates a halt or stop grant special cycle on PCI.

With a standard SRAM secondary cache, a halt or stop grant special cycle from the CPU causes the PCMC to negate CCS[1:0]#, deselecting the SRAMs and placing them in a low power standby mode. When the cache is in stand-by mode, the first bus cycle from the CPU brings the cache out of stand-by and into active mode, enabling the SRAMs to service the cycle in the case of a hit to the cache. The PCMC asserts CCS[1:0]# as a propagation delay from the falling edge of ADS#. CCS[1:0]# are then left asserted until the next halt or stop grant special cycle is occurs. When exiting the powerdown state, the PCMC ignores the Secondary Cache Leadoff wait-states bit and executes a 3-2-2-2 read or 4-2-2-2 write in order to allow the SRAMs time to power up. In the case of a read cycle, COE[1:0]# are asserted in clock two as in the case of ordinary read cycles.

When the SRAMs are powered down, the PCMC asserts CCS[1:0]# when performing a snoop cycle, regardless of whether the cycle hits in the second level cache. The PCMC then negates CCS# after the snoop cycle is complete.

With a burst SRAM secondary cache, a halt or stop grant special cycle from the CPU causes the PCMC to negate CCS# and assert CADS[1:0]#, deselecting the SRAMs, placing them in a low power standby mode. CCS# is then asserted and is left asserted by the PCMC. Thus, when the first cycle is driven from the CPU, the SRAMs sample ADSP# and CS# active, placing them in active mode and initiating the first access.

If the SRAMs are required to service a snoop, they are brought out of power-down when the PCMC asserts CADS[1:0]#. The PCMC always asserts CADS[1:0]# with CCS# negated after a snoop cycle is complete, regardless of whether the SRAMs were powered down prior to the snoop cycle.

5.2.4 SNOOP CYCLES

For snoop operations, refer to Section 5.1, 82434LX Cache.

5.2.5 FLUSH, FLUSH ACKNOWLEDGE, AND WRITE-BACK SPECIAL CYCLES

For flush, flush acknowledge, and write-back special cycles, refer to Section 5.1, 82434LX Cache.

6.0 DRAM INTERFACE

This section describes the DRAM interface for the 82434LX DRAM Interface (Section 6.1) and the 82434NX DRAM Interface (Section 6.2). The differences are in the following areas:

1. Increased maximum DRAM memory size to 512 MBytes. An extra address line (MA11) has been added to the 82434NX.
2. Two additional RAS# lines for a total of eight (RAS[0:7]#.
3. Addition of 50 MHz host-bus optimized DRAM timing sets. Thus, the 82434LX supports 60 and 66 MHz frequencies and the 82434NX supports 50, 60, and 66 MHz.

6.1 82434LX DRAM Interface

The 82434LX PCMC integrates a high performance DRAM controller supporting from 2–192 MBytes of main memory. The PCMC generates the RAS#, CAS#, WE# and multiplexed addresses for the DRAM array, while the data path to DRAM is provided by two 82433LX LBXs. The DRAM controller interface is fully configurable through a set of control registers. Complete descriptions of these registers are given in Section 3.0, Register Description. A brief overview of the registers which configure the DRAM interface is provided in this section.

The 82434LX controls a 64-bit memory array (72-bit including parity) ranging in size from 2 MBytes up to 192 MBytes using industry standard 36-bit wide memory modules with fast page-mode DRAMs. Both single- and double-sided SIMMs are supported. The eleven multiplexed address lines, MA[10:0] allow the PCMC to support 256K x 36, 1M x 36, and 4M x 36 SIMMs. The PCMC has six RAS# lines enabling the support of up to six rows of DRAM. Eight CAS# lines allow byte control over the array during read and write operations. The PCMC supports 70 and 60 ns DRAMs. The PCMC DRAM interface is synchronous to the CPU clock and supports page mode accesses to efficiently transfer data in bursts of four Qwords.

The DRAM interface of the PCMC is configured by the DRAM Control Mode Register (offset 57h) and the six DRAM Row Boundary (DRB) Registers (offsets 60h–65h). The DRAM Control Mode Register contains bits to configure the DRAM interface for RAS# modes and refresh options. In addition, DRAM Parity Error Reporting and System Management RAM space can be enabled and disabled. When System Management RAM is enabled, if SMIACT# from the Pentium processor is not asserted, all CPU read and write accesses to SMM memory are directed to PCI. The SMRAM Space Register at configuration space offset 72h provides additional control over the SMRAM space. The six DRB Registers define the size of each row in the memory array, enabling the PCMC to assert the proper RAS# line for accesses to the array.

CPU-to-Memory write posting and read-around-write operations are enabled and disabled via the Host Read/Write Buffer Control Register (offset 53h). PCI-to-Memory write posting is enabled and disabled via the PCI Read/Write Buffer Control Register (offset 54h). PCI master reads from main memory always result in the PCMC and LBXs reading the requested data and prefetching the next seven Dwords.

Seven Programmable Attribute Map (PAM) Registers (offsets 59h–5Fh) are used to specify the cacheability and read/write status of the memory space between 512 KBytes and 1 MByte. Each PAM Register defines a specific address area enabling the system to selectively mark specific memory ranges as cacheable, read-only, write-only, read/write or disabled. When a memory range is disabled, all CPU accesses to that range are directed to PCI.

Two other registers also affect the DRAM interface, the Memory Space Gap Register (offsets 78h–79h) and the Frame Buffer Range Register (offsets 7Ch–7Fh). The Memory Space Gap Register is used to place a logical hole in the memory space between 1 MByte to 16 MBytes to accommodate memory mapped ISA boards. The Frame Buffer Range Register, is used to map a linear frame buffer into the Memory Space Gap or above main memory. When enabled, accesses to these ranges are never directed to the DRAM interface, but are always directed to PCI.

6.1.1 DRAM CONFIGURATIONS

Figure 42 illustrates a 12-SIMM configuration which supports single-sided SIMMs. A row in the DRAM array is made up of two SIMMs which share a common RAS# line. SIMM0 and SIMM1 are connected to RAS0# and therefore, comprise row 0. SIMM10 and SIMM11 form row 5. Within any given row, the two SIMMs must be the same size. Among the six rows, SIMM densities can be mixed in any order. That is, there are no restrictions on the ordering of SIMM densities among the six rows.

The low order LBX (LBXL) is connected to byte lanes 5, 4, 1, and 0 of the host and memory data buses, and the lower two bytes of the PCI AD bus. The high order LBX (LBXH) is connected to byte lanes 7, 6, 3, and 2 of the host and memory data buses, and the upper two bytes of the PCI AD bus. Thus, SIMMs connected to LBXL are connected to CAS[5:4,1:0]# and SIMMs connected to LBXH are connected to CAS[7:6, 3:2]#.

The MA[10:0] and WE# lines are externally buffered to drive the large capacitance of the memory array. Three buffered copies of the MA[10:0] and WE# signals are required to drive the six row array.

Figure 43 illustrates a 6-SIMM configuration that supports either single- or double-sided SIMMs. In this configuration, single- and double-sided SIMMs can be mixed. For example, if single-sided SIMMs are installed into the sockets marked SIMM0 and SIMM1, then RAS0# is connected to the SIMMs and RAS1# is not connected. Row 0 is then populated and row 1 is empty. Two double-sided SIMMs could then be installed in the sockets marked SIMM2 and SIMM3, populating rows 2 and 3.

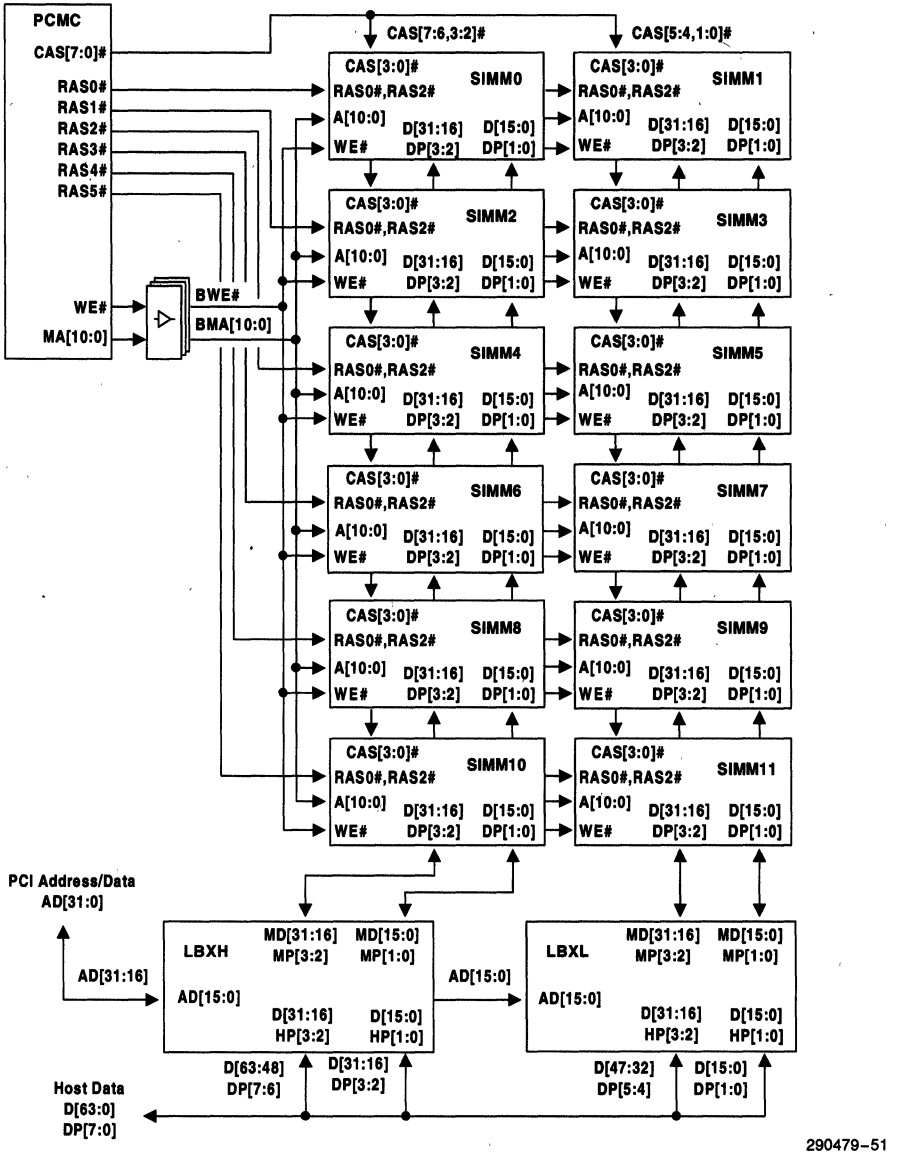
6.1.2 DRAM ADDRESS TRANSLATION

The 82434LX multiplexed row/column address to the DRAM memory array is provided by the MA[10:0] signals. The MA[10:0] bits are derived from the host address bus as defined by Table 12.

MA[10:0] are translated from the host address A[24:3] for all memory accesses, except those targeted to memory that has been remapped as a result of the creation of a memory space gap in the lower extended memory area. In the case of a cycle targeting remapped memory, the least significant bits come directly from the host address, while the more significant bits depend on the memory space gap start address, gap size, and the size of main memory.

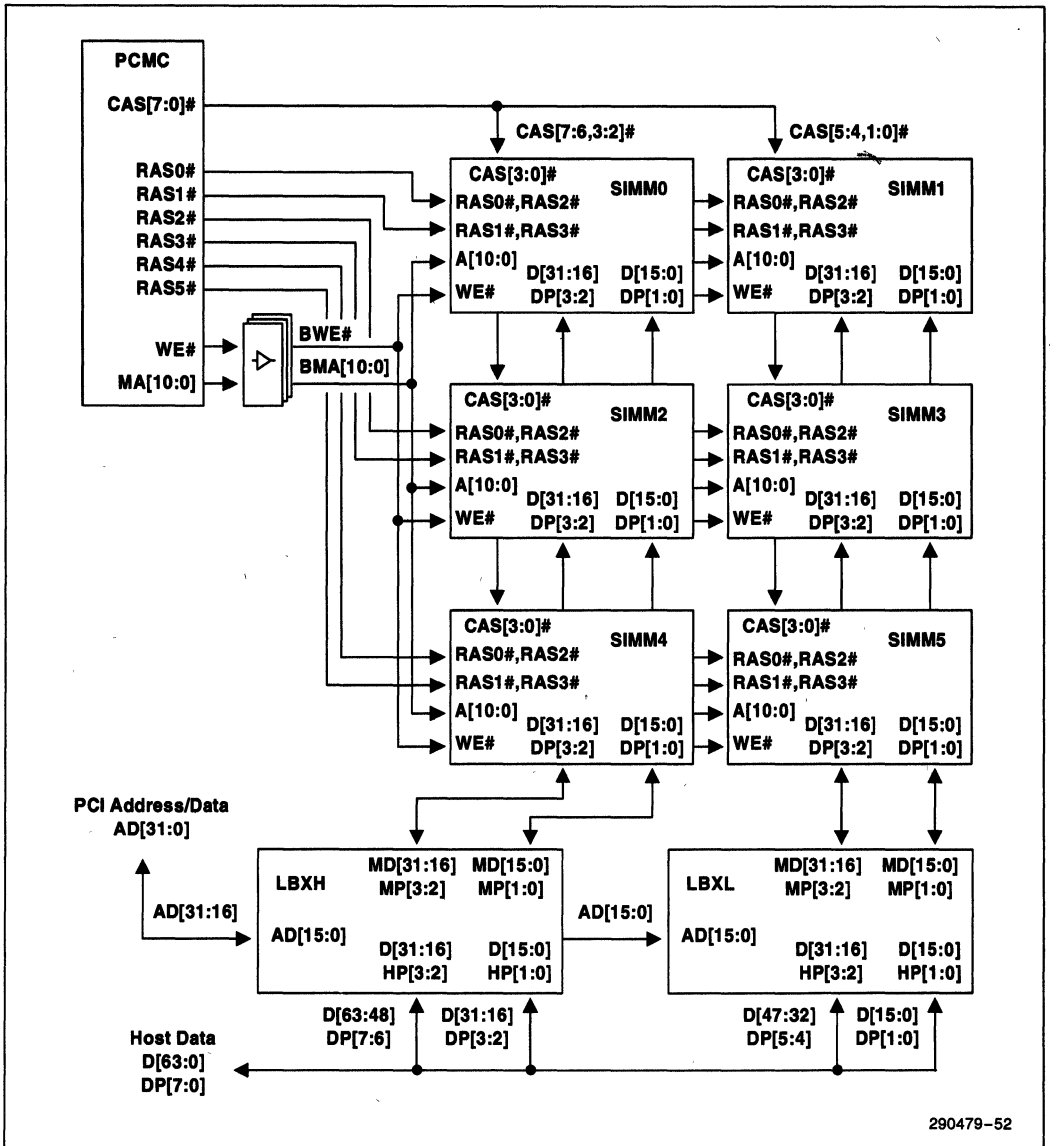
Table 12. DRAM Address Translation

Memory Address, MA[10:0]	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Row Address	A24	A22	A20	A19	A18	A17	A16	A15	A14	A13	A12
Column Address	A23	A21	A11	A10	A9	A8	A7	A6	A5	A4	A3



NOTE:
The figure shows the connections for the 82434LX. For the 82434NX, there are two additional RAS lines (RAS[7:6]#) and one additional address line (MA11).

Figure 42. 82434LX DRAM Configuration Supporting Single-Sided SIMMs



NOTE:
The figure shows the connections for the 82434LX. For the 82434NX, there are two additional RAS lines (RAS[7:6]#) and one additional address line (MA11).

Figure 43. 82434LX DRAM Configuration Supporting Single- or Double-Sided SIMMs

6.1.3 CYCLE TIMING SUMMARY

The 82434LX PCMC DRAM performance is summarized in Table 13 for all CPU read and write cycles.

Table 13. CPU to DRAM Performance Summary

Cycle Type	Burst, x-4-4-4 Timing	Single, x-4-4-4 Timing
Read Page Hit	7-4-4-4	7
Read Row Miss	11-4-4-4	11
Read Page Miss	14-4-4-4	14
Posted Write, WT L2	3-1-1-1	3
Posted Write, WB L2	4-1-1-1	4
Write Page Hit	12-4-4-4	12
Write Row Miss	13-4-4-4	13
Write Page Miss	16-4-4-4	16
0-Active RAS# Mode Read	10-4-4-4	10
0-Active RAS# Mode Write	12-4-4-4	12

CPU writes to the CPU-to-Memory Posted Write Buffer are completed at 3-1-1-1 when the second level cache is configured for write-through mode and 4-1-1-1 when the cache is configured for write-back mode. Table 14 shows the refresh performance in CPU clocks.

Table 14. Refresh Cycle Performance

Refresh Type	Hidden Refresh	RAS# only Refresh	CAS# before RAS#
Single	12	13	14
Burst of Four	48	52	56

6.1.4 CPU TO DRAM BUS CYCLES

This section describes the CPU-to-DRAM cycles for the 82434LX.

6.1.4.1 Read Page Hit

Figure 44 depicts a CPU burst read page hit from DRAM. The 82434LX PCMC decodes the CPU address as a page hit and drives the column address onto the MA[10:0] lines. CAS[7:0]# are then asserted to cause the DRAMs to latch the column address and begin the read cycle. CMR (CPU Memory Read) is driven on the HIG[4:0] lines to enable the memory data to host data path through the LBXs. The PCMC advances the MA[1:0] lines through the Pentium processor burst order, negating and asserting CAS[7:0]# to read each Qword. The host data is latched on the falling edge of MDLE, when CAS[7:0]# are negated. The latch is opened again when MDLE is sampled asserted by the LBXs. The LBXs tri-state the host data bus when HIG[4:0] change to NOPC and MDLE rises. A single read page hit from DRAM is similar to the first read of this sequence. The HIG[4:0] lines are driven to NOPC when BRDY# is asserted.

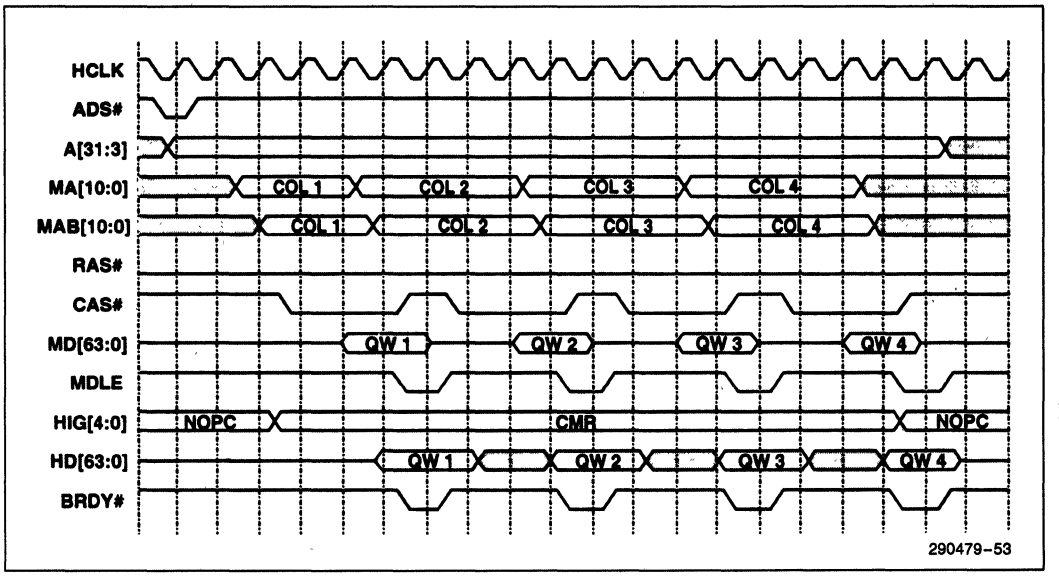


Figure 44. Burst DRAM Read Cycle-Page Hit

2

6.1.4.2 Read Page Miss

Figure 45 depicts a CPU burst read page miss from DRAM. The 82434LX decodes the CPU address as a page miss and switches from initially driving the column address to driving the row address on the MA[10:0] lines. RAS# is then negated to precharge the DRAMs and then asserted to cause the DRAMs to latch the new row address. The PCMC then switches the MA[10:0] lines to drive the column address and asserts CAS[7:0]#. CMR (CPU Memory Read) is driven on the HIG[4:0] lines to enable the memory data to host data path through the LBXs. The PCMC advances the MA[1:0] lines through the Pentium processor burst order, negating and asserting CAS[7:0]# to read each Qword. The host data is latched on the falling edge of MDLE, when CAS[7:0]# are negated. The latch is opened again when MDLE is sampled asserted by the LBXs. The LBXs tri-state the host data bus when HIG[4:0] change to NOPC and MDLE rises. A single read page miss from DRAM is similar to the first read of this sequence. The HIG[4:0] lines are driven to NOPC when BRDY# is asserted.

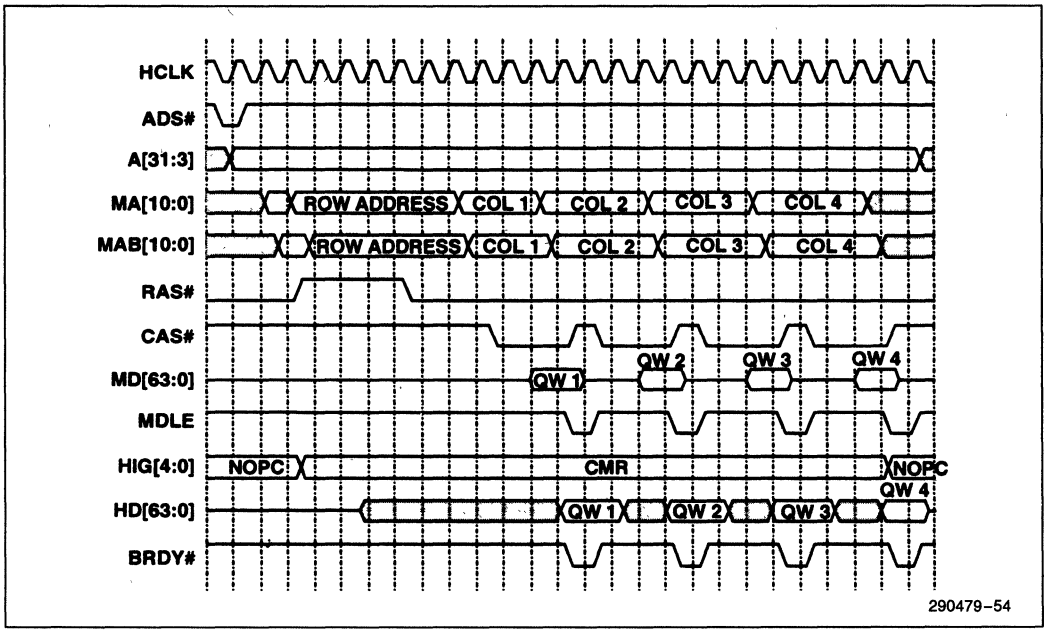


Figure 45. DRAM Read Cycle-Page Miss

6.1.4.3 Read Row Miss

Figure 46 depicts a CPU burst read row miss from DRAM. The 82434LX decodes the CPU address as a row miss and switches from initially driving the column address to driving the row address on the MA[10:0] lines. The RAS# signal that was asserted is negated and the RAS# for the currently accessed row is asserted. The PCMC then switches the MA[10:0] lines to drive the column address and asserts CAS[7:0]#. CMR (CPU Memory Read) is driven on the HIG[4:0] lines to enable the memory data to host data path through the LBXs. The PCMC advances the MA[1:0] lines through the Pentium processor burst order, negating and asserting CAS[7:0]# to read each Qword. The host data is latched on the falling edge of MDLE, when CAS[7:0]# are negated. The latch is opened again when MDLE is sampled asserted by the LBXs. The LBXs tri-state the host data bus when HIG[4:0] change to NOPC and MDLE rises. A single read row miss from DRAM is similar to the first read of this sequence. The HIG[4:0] lines are driven to NOPC when BRDY# is asserted.

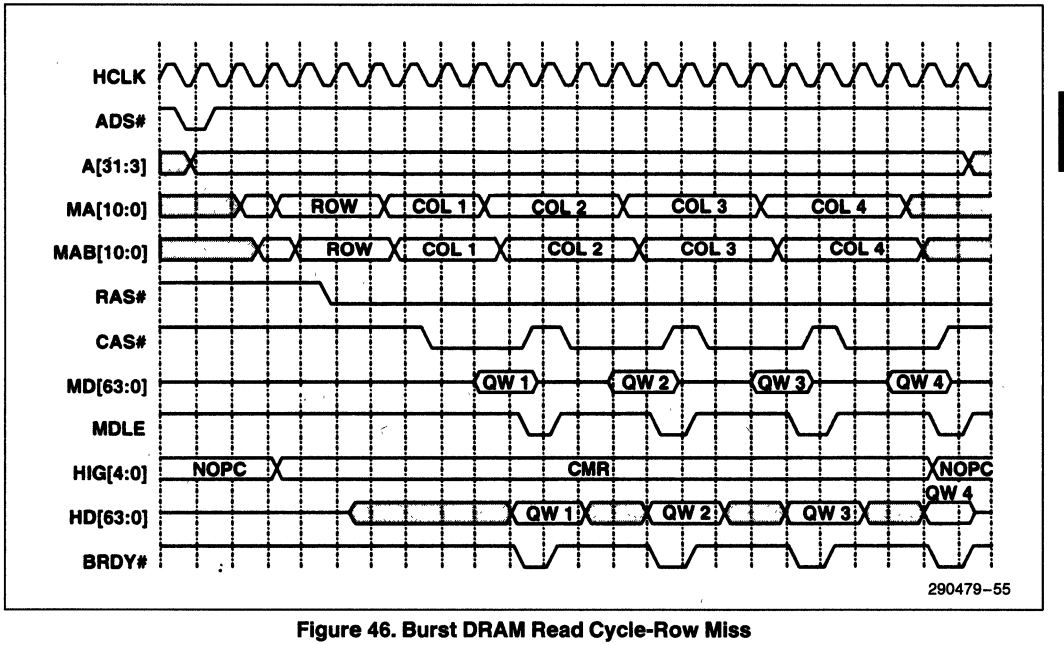


Figure 46. Burst DRAM Read Cycle-Row Miss

6.1.4.4 Write Page Hit

Figure 47 depicts a CPU burst write page hit from DRAM. The 82434LX decodes the CPU write cycle as a DRAM page hit. The HIG[4:0] lines are driven to PCMWQ to post the write to the LBXs. In the figure, the write cycle is posted to the CPU-to-Memory Posted Write Buffer at 4-1-1-1. The write is posted at 4-1-1-1 when the second level cache is configured for a write-back policy. The write is posted to DRAM at 3-1-1-1 when the second level cache is configured for a write-through policy. When the cycle is decoded as a page hit, the PCMC asserts WE# and drives the RCMWQ command on MIG[2:0] to enable the LBXs to drive the first Qword of the write onto the memory data lines. MEMDRV is then driven to cause the LBXs to continue to drive the first Qword for three more clocks. CAS[7:0]# are then negated and asserted to perform the writes to the DRAMs as the MA[1:0] lines advance through the Pentium processor burst order. A single write is similar to the first write of the burst sequence. MIG[2:0] are driven to NOPM in the clock after CAS[7:0]# are asserted.

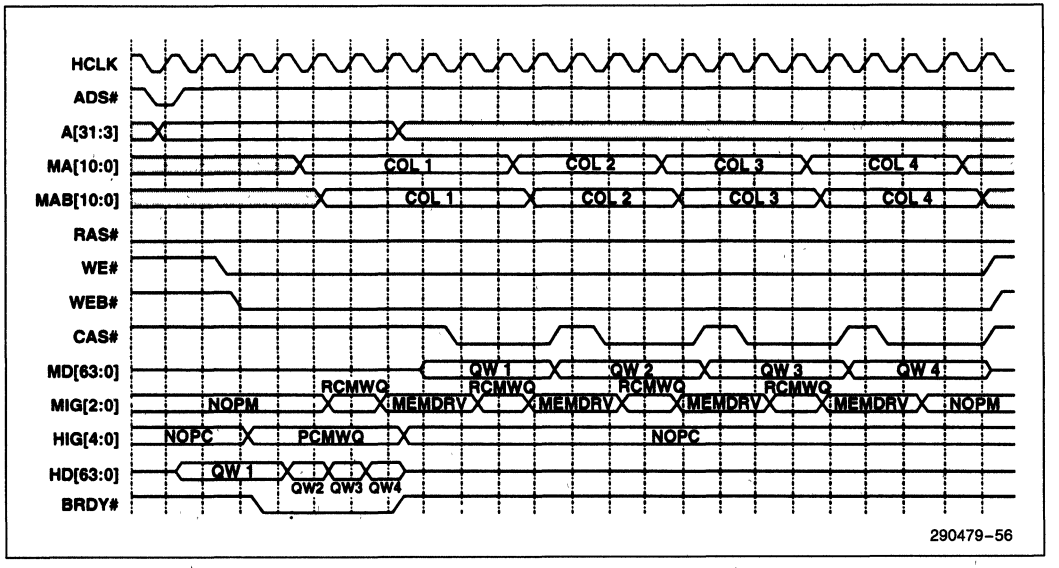


Figure 47. Burst DRAM Write Cycle-Page Hit

6.1.4.5 Write Page Miss

Figure 48 depicts a CPU burst write page miss to DRAM. The 82434LX decodes the CPU write cycle as a DRAM page miss. The HIG[4:0] lines are driven to PCMWQ to post the write to the LBXs. In the figure, the write cycle is posted to the CPU-to-Memory Posted Write Buffer at 4-1-1-1. The write is posted at 4-1-1-1 when the second level cache is configured for a write-back policy. The write is posted to DRAM at 3-1-1-1 when the second level cache is configured for a write-through policy. When the cycle is decoded as a page miss, the PCMC switches the MA[10:0] lines from the column address to the row address and asserts WE#. The PCMC drives the RCMWQ command on MIG[2:0] to enable the LBXs to drive the first Qword of the write onto the memory data lines. MEMDRV is then driven to cause the LBXs to continue to drive the first Qword. The RAS# signal for the currently decoded row is negated to precharge the DRAMs. RAS# is then asserted to cause the DRAMs to latch the row address. The PCMC then switches the MA[10:0] lines to the column address and asserts CAS[7:0]# to initiate the first write. CAS[7:0]# are then negated and asserted to perform the writes to the DRAMs as the MA[1:0] lines advance through the Pentium processor burst order. A single write is similar to the first write of the burst sequence. MIG[2:0] are driven to NOPM in the clock after CAS[7:0]# are asserted.

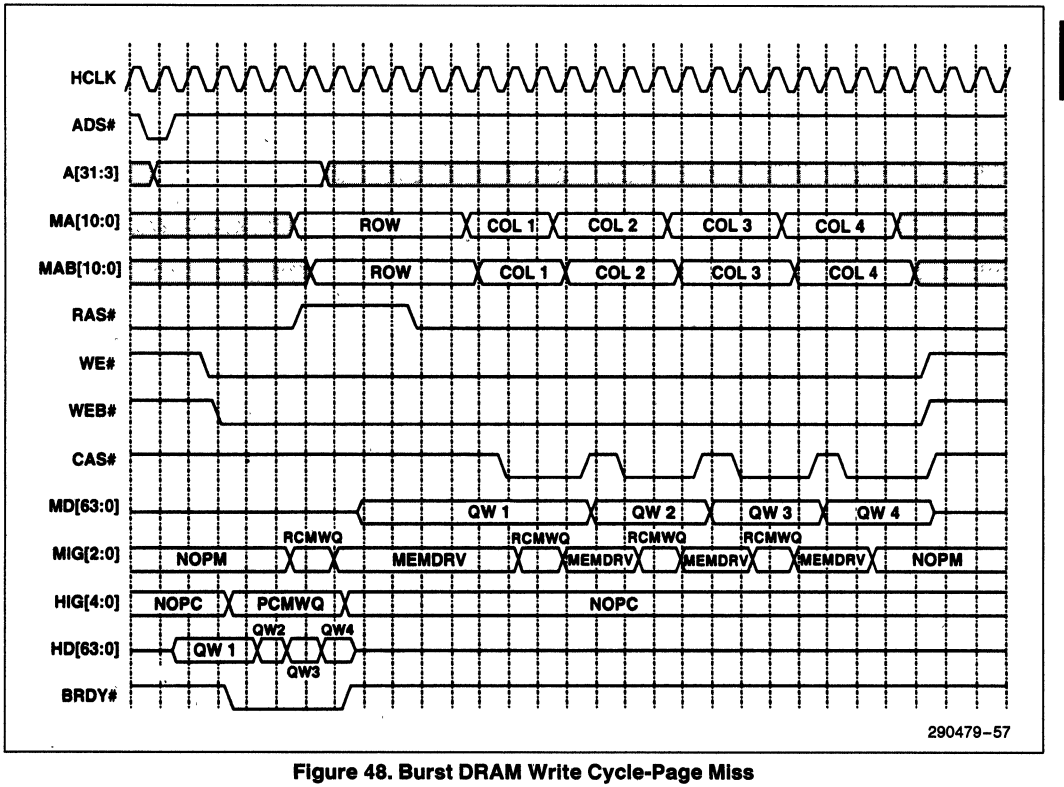


Figure 48. Burst DRAM Write Cycle-Page Miss

6.1.4.6 Write Row Miss

Figure 49 depicts a CPU burst write row miss to DRAM. The 82434LX decodes the CPU write cycle as a DRAM row miss. The HIG[4:0] lines are driven to PCMWQ to post the write to the LBXs. In the figure, the write cycle is posted to the CPU-to-Memory Posted Write Buffer at 4-1-1-1. The write is posted at 4-1-1-1 when the second level cache is configured for a write-back policy. The write is posted to DRAM at 3-1-1-1 when the second level cache is configured for a write-through policy. When the cycle is decoded as a row miss, the PCMC negates the already active RAS# signal, switches the MA[10:0] lines from the column address to the row address and asserts the RAS# signal for the currently decoded row. The PCMC asserts WE# and drives the RCMWQ command on MIG[2:0] to enable the LBXs to drive the first Qword of the write onto the memory data lines. MEMDRV is then driven to cause the LBXs to continue to drive the first Qword. The PCMC then switches the MA[10:0] lines to the column address and asserts CAS[7:0]# to initiate the first write. CAS[7:0]# are then negated and asserted to perform the writes to the DRAMs as the MA[1:0] lines advance through the Pentium processor burst order. A single write is similar to the first write of the burst sequence. MIG[2:0] are driven to NOPM in the clock after CAS[7:0]# are asserted.

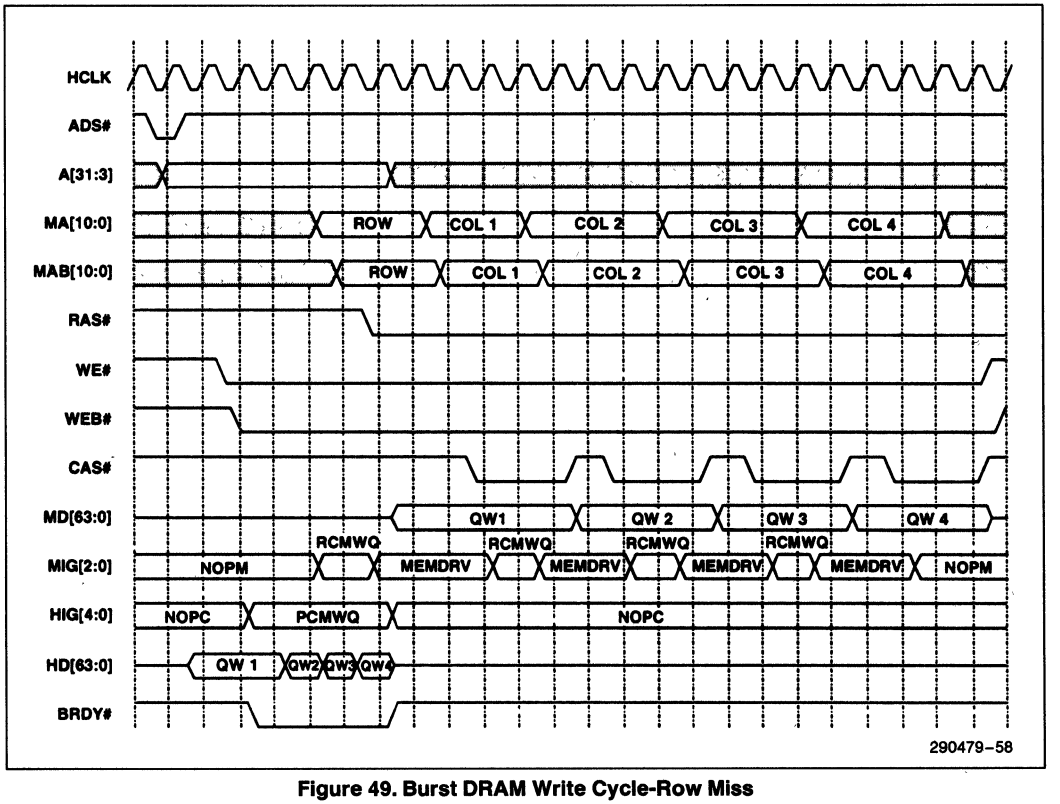


Figure 49. Burst DRAM Write Cycle-Row Miss

6.1.4.7 Read Cycle, 0-Active RAS# Mode

When in 0-active RAS# mode, every CPU cycle to DRAM results in a RAS# and CAS# sequence. RAS# is always negated after a cycle completes. Figure 50 depicts a CPU burst read cycle from DRAM where the 82434LX is configured for 0-active RAS# mode. When in 0-active RAS# mode, the PCMC defaults to driving the row address on the MA[10:0] lines. The PCMC asserts the RAS# signal for the currently decoded row causing the DRAMs to latch the row address. The PCMC then switches the MA[10:0] lines to drive the column address and asserts CAS[7:0]#. CMR (CPU Memory Read) is driven on the HIG[4:0] lines to enable the memory data to host data path through the LBXs. The PCMC advances the MA[1:0] lines through the Pentium processor burst order, negating and asserting CAS[7:0]# to read each Qword. The host data is latched on the falling edge of MDLE, when CAS[7:0]# are negated. The latch is opened again when MDLE is sampled asserted by the LBXs. The LBXs tri-state the host data bus when HIG[4:0] change to NOPC and MDLE rises. A single read row miss from DRAM is similar to the first read of this sequence. The HIG[4:0] lines are driven to NOPC when BRDY# is asserted. RAS# is negated with CAS[7:0]#.

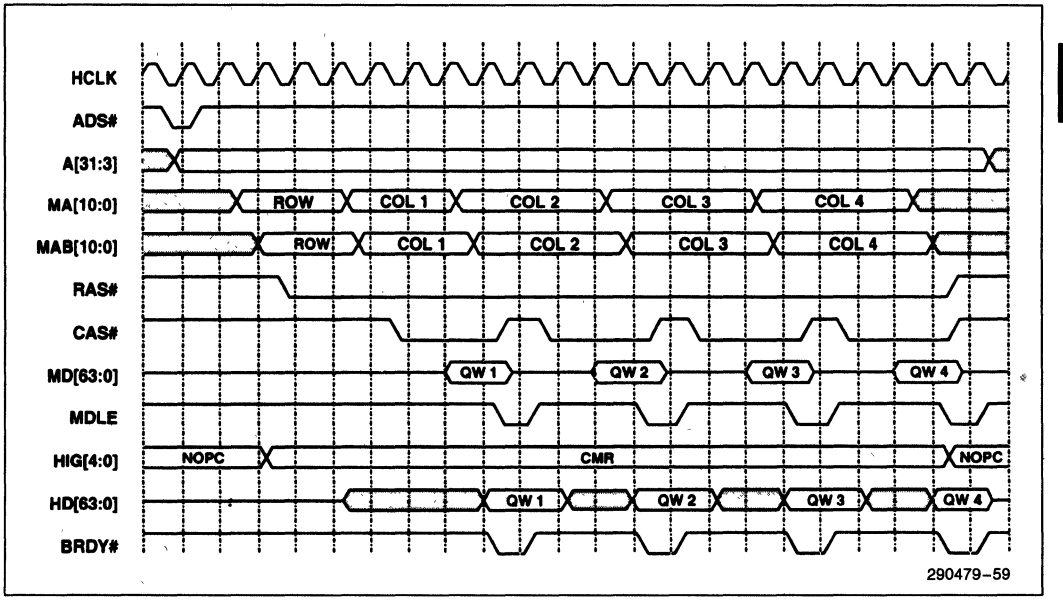


Figure 50. Burst DRAM Read Cycle, 0-Active RAS# Mode

6.1.4.8 Write Cycle, 0-Active RAS# Mode

When in 0-active RAS# mode, every CPU cycle to DRAM results in a RAS# and CAS# sequence. RAS# is always negated after a cycle completes. Figure 51 depicts a CPU Burst Write Cycle to DRAM where the 82434LX is configured for 0-active RAS# mode. The HIG[4:0] lines are driven to PCMWQ to post the write to the LBXs. In the figure, the write cycle is posted to the CPU-to-Memory Posted Write Buffer at 4-1-1-1. The write is posted at 4-1-1-1 when the second level cache is configured for a write-back policy. The write is posted to DRAM at 3-1-1-1 when the second level cache is configured for a write-through policy. When in 0-active RAS# mode, the PCMC defaults to driving the row address on the MA[10:0] lines. The PCMC asserts the RAS# signal for the currently decoded row causing the DRAMs to latch the row address. The PCMC asserts WE# and drives the RCMWQ command on MIG[2:0] to enable the LBXs to drive the first Qword of the write onto the memory data lines. MEMDRV is then driven to cause the LBXs to continue to drive the first Qword. The PCMC then switches the MA[10:0] lines to the column address and asserts CAS[7:0]# to initiate the first write. CAS[7:0]# are then negated and asserted to perform the writes to the DRAMs as the MA[1:0] lines advance through the Pentium processor burst order. A single write is similar to the first write of the burst sequence. MIG[2:0] are driven to NOPM in the clock after CAS[7:0] are asserted.

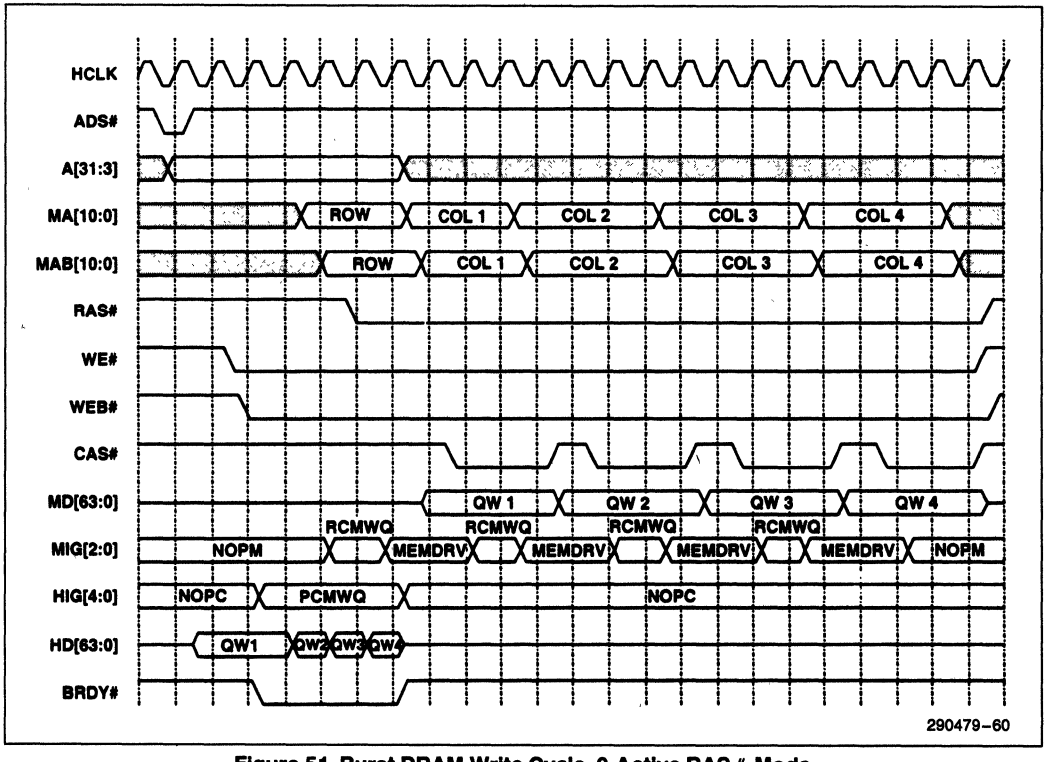


Figure 51. Burst DRAM Write Cycle, 0-Active RAS# Mode

6.1.5 REFRESH

The refresh of the DRAM array can be performed by either using RAS#-only or CAS#-before-RAS# refresh cycles. When programmed for CAS#-before-RAS# refresh, hidden refresh cycles are initiated when possible. RAS# only refresh can be used with any type of second level cache configuration (i.e., no second level cache is present, or either a burst SRAM or standard SRAM second level cache is implemented). CAS#-before-RAS# refresh can be enabled when either no second level cache is present or a burst SRAM second level cache is implemented. CAS#-before-RAS# refresh should not be used when a standard SRAM second level cache is implemented. The timing of internally generated refresh cycles is derived from HCLK and is independent of any expansion bus refresh cycles.

The DRAM controller contains an internal refresh timer which periodically requests the refresh control logic to perform either a single refresh or a burst of four refreshes. The single refresh interval is 15.6 μs. The interval for burst of four refreshes is four times the single refresh interval, or 62.4 μs. The PCMC is configured for either single or burst of four refresh and either RAS#-only or CAS#-before-RAS# refresh via the DRAM Control Register (offset 57h).

To minimize performance impact, refresh cycles are partially deferred until the DRAM interface is idle. The deferment of refresh cycles is limited by the DRAM maximum RAS# low time of 100 μs. Refresh cycles are initiated such that the RAS# maximum low time is never violated.

Hidden refresh cycles are run whenever all eight CAS# lines are active when the refresh cycle is internally requested. Normal CAS#-before-RAS# refresh cycles are run whenever the DRAM interface is idle when the refresh is requested, or when any subset of the CAS# lines is inactive as the refresh is internally requested.

To minimize the power surge associated with refreshing a large DRAM array the DRAM interface staggers the assertion of the RAS# signals during both CAS#-before-RAS# and RAS#-only refresh cycles. The order of RAS# edges is dependent on which RAS# was most recently asserted prior to the refresh sequence. The RAS# that was active will be the last to be activated during the refresh sequence. All RAS[5:0]# lines are negated at the end of refresh cycles, thus, the first DRAM cycle after a refresh sequence is a row miss.

2

6.1.5.1 RAS#-Only Refresh-Single

Figure 52 depicts a RAS#-only refresh cycle when the 82434LX is programmed for single refresh cycles. The diagram shows a CPU read cycle completing as the refresh timing inside the PCMC generates a refresh request. The refresh address is driven on the MA[10:0] lines. Since the CPU cycle was to row 0, RAS0# is negated. RAS1# is the first to be asserted. RAS2# through RAS5# are then asserted sequentially while RAS0# is driven high, precharging the DRAMs in row 0. RAS0# is then asserted after RAS5#. Each RAS# line is asserted for six host clocks.

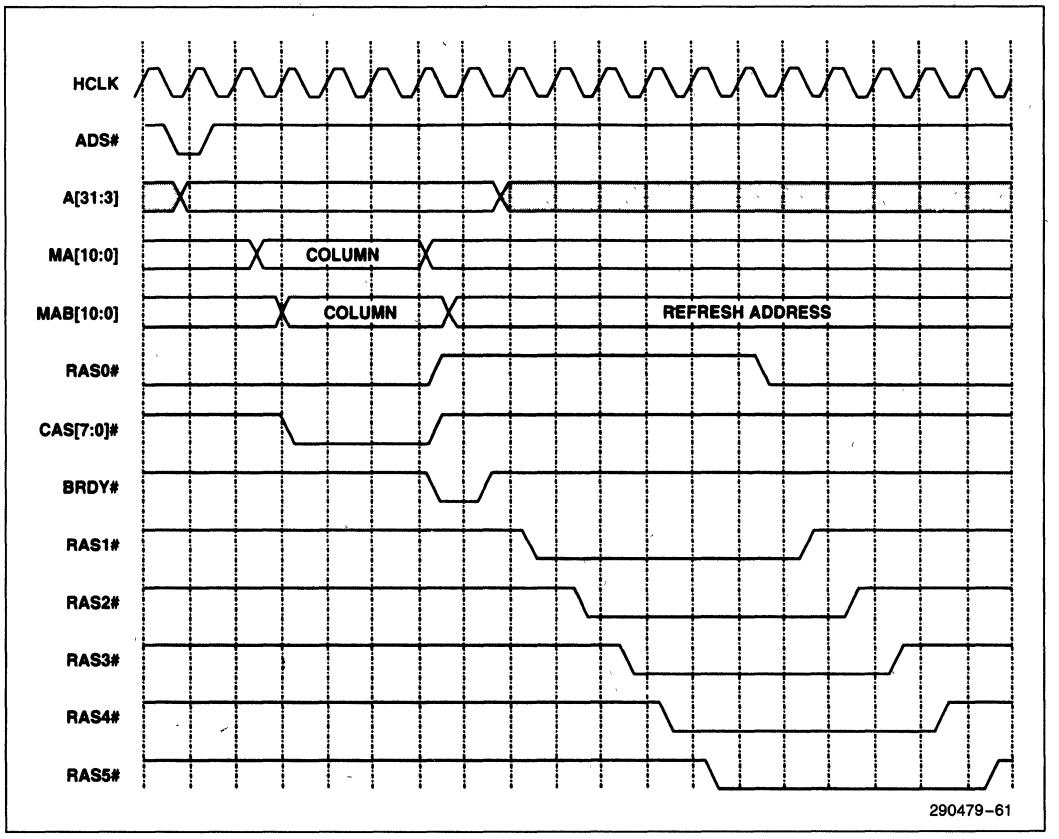


Figure 52. RAS# Only Refresh-Single

6.1.5.2 CAS#-before-RAS# Refresh-Single

Figure 53 depicts a CAS#-before-RAS# refresh cycle when the 82434LX is programmed for single refresh cycles. The diagram shows a CPU read cycle completing as the refresh timing inside the PCMC generates a refresh request. The CPU read cycle is less than a Qword, therefore a hidden refresh is not initiated. After the CPU read cycle completes, all of the RAS# and CAS# lines are negated. The PCMC then asserts CAS[7:0]# and then sequentially asserts the RAS# lines, starting with RAS1# since RAS0# was the last RAS# line asserted. Each RAS# line is asserted for six clocks.

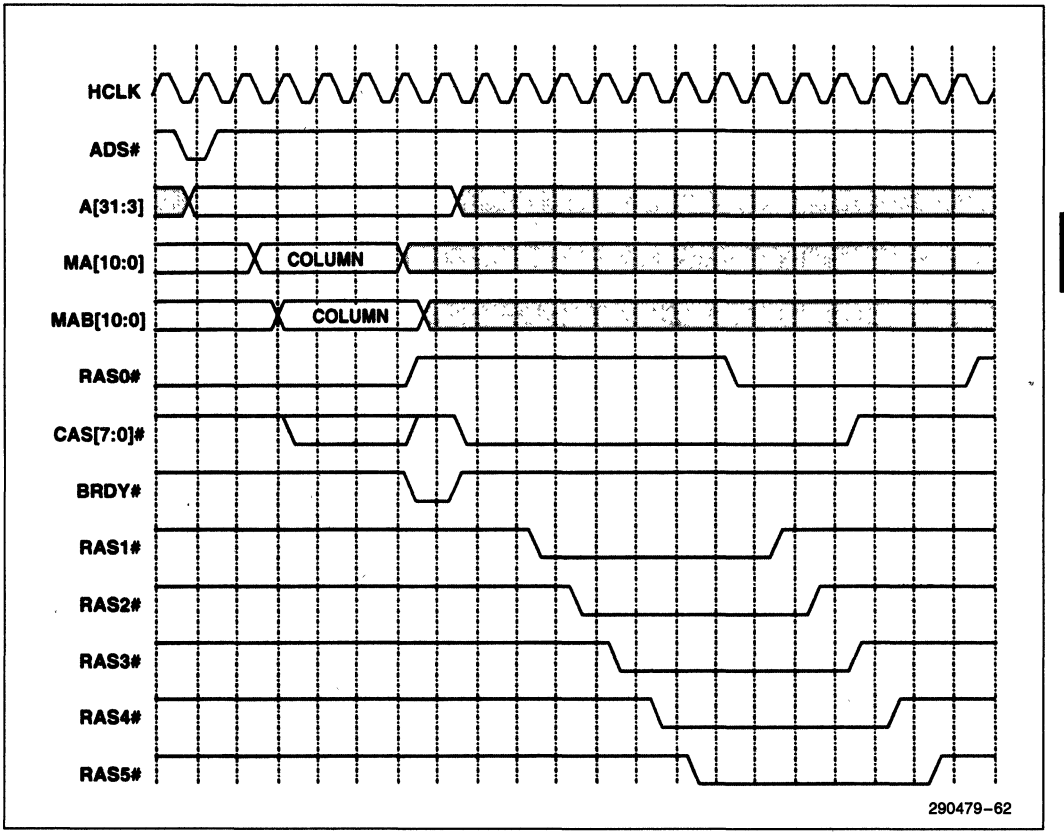


Figure 53. CAS#-before-RAS# Refresh-Single

6.1.5.3 Hidden Refresh-Single

Figure 54 depicts a hidden refresh cycle which takes place after a DRAM read page hit cycle. The diagram shows a CPU read cycle completing as the refresh timing inside the 82434LX generates a refresh request. The CPU read cycle is an entire Qword, therefore a hidden refresh is initiated. After the CPU read cycle completes, RAS# is negated, but all eight CAS# lines remain asserted. The PCMC then sequentially asserts the RAS# lines, starting with RAS1# since RAS0# was the last active RAS# line. Each RAS# line is asserted for six clocks.

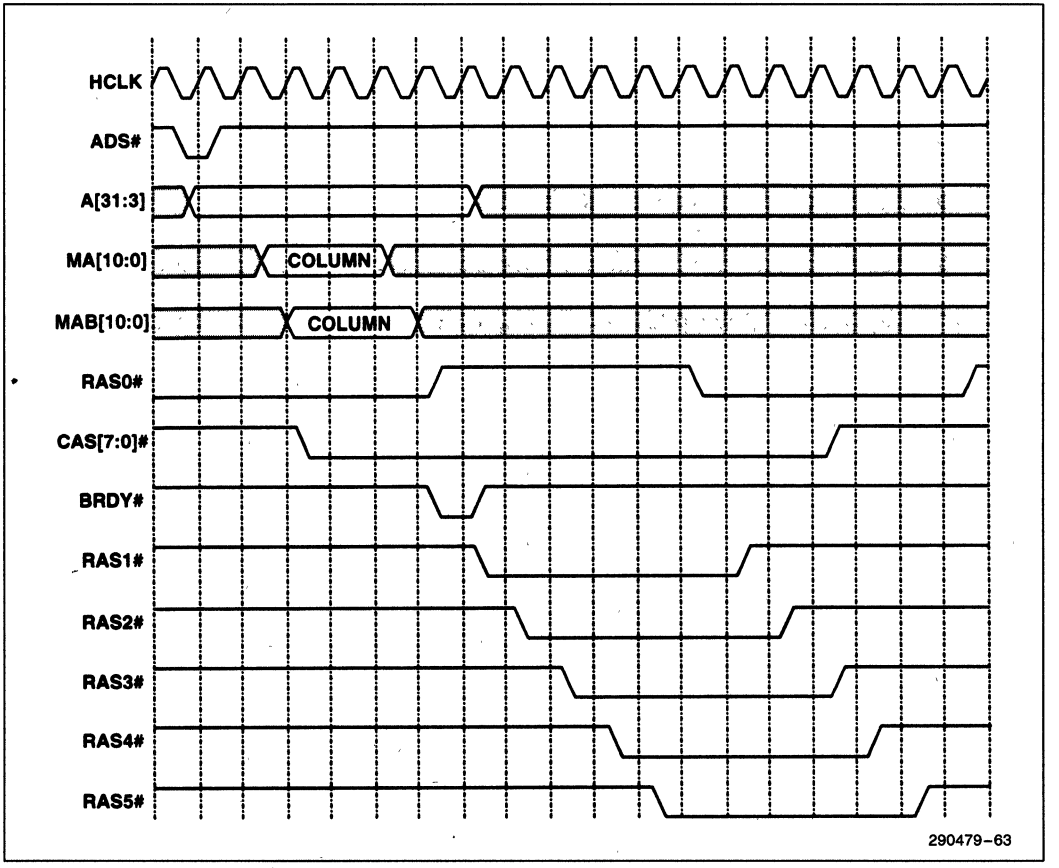


Figure 54. Hidden Refresh-Single

6.2 82434NX DRAM Interface

This section describes the 82434NX DRAM Interface. Changes in the 82430NX PCIset from the 82430 PCIset include:

1. Increased maximum DRAM memory size to 512 MBytes. The 82430NX PCIset increases the maximum memory array size from 192 MBytes to 512 MBytes.
2. Two additional row address lines (RAS[7:6]#) for a total of eight (RAS[7:0]#).
3. Addition of 50 MHz host-bus optimized DRAM timing sets.
4. Three additional registers are added to support the increased memory sizeDRAM Row Boundary Registers 6 and 7 (DRB[7:6]) and the DRAM Row Boundary Extension (DRBE) Register.
5. Modified MA[11:0] timing to provide more MA[11:0] setup time to CAS[7:0]# assertion.

6.2.1 DRAM ADDRESS TRANSLATION

The MA[11:0] lines are translated from the host address lines A[26:3] for all memory accesses, except those targeted to memory that has been remapped as a result of the creation of a memory space gap in the lower extended memory area. In the case of a cycle targeting remapped memory, the least significant bits come directly from the host address, while the more significant bits depend on the memory space gap start address, gap size, and the size of main memory.

2

Table 15. DRAM Address Translation

Memory Address MA[11:0]	11	10	9	8	7	6	5	4	3	2	1	0
Column Address	A25	A23	A21	A11	A10	A9	A8	A7	A6	A5	A4	A3
Row Address	A26	A24	A22	A20	A19	A18	A17	A16	A15	A14	A13	A12

6.2.2 CYCLE TIMING SUMMARY

The 82434NX PCMC DRAM performance for 50 MHz Host bus clock is summarized in Table 13 for all CPU read and write cycles. The 60/66 MHz MA[11:0] timings when in X-4-4-4 mode have one difference from the 82434LX MA[11:0] timings. The MA lines switch to the next address in the burst sequence one clock sooner than in the 82434LX, providing more MA[11:0] setup time to CAS[7:0]# assertion. The 60/66 MHz DRAM timings for write cycles have been improved by 1 clock for all leadoffs. The 50 MHz timings shown below are selected by HOF=00, DBT=11, RWS=0 and CWS=0.

Table 16. CPU to DRAM Performance Summary for 50 MHz Host Bus Clock

Cycle Type	x-3-3-3 Timing(1)
Read (Page Hit/Row Miss/ Page Miss)	6/10/12-3-3-3
Posted Write	4-1-1-1
Write (Page Hit/Row Miss/ Page Miss)	10/11/13-3-3-3
0-Active RAS# Mode Reads	9-3-3-3
0-Active RAS# Mode Writes	9-3-3-3

NOTES:
1. Single cycle timings are identical to these leadoff timings.

Table 17. Refresh Cycle Performance (Independent of CPU frequency)

Refresh Type	Hidden Refresh	RAS# Only Refresh	CAS#-Before-RAS#
Single	16	17	18
Burst of Four	64	68	72

6.2.3 CPU TO DRAM BUS CYCLES

In this section, all timing diagrams are for 50 MHz DRAM timing, 1-Active RAS mode. The 60/66 MHz MA[11:0] timings when in X-4-4-4 mode have one difference from the 82434LX MA[11:0] timings. The MA lines switch to the next address in the burst sequence one clock sooner than in the 82434LX. The write cycle leadoffs are 1 clock earlier for 82430NX than 82430 (the MIGs and CAS timings improved by 1 clock). The 0-Active RAS# modes closely resemble the row miss cases. In 0-Active RAS# mode, RAS# is asserted one clock sooner than is shown in the row miss timing diagrams.

6.2.3.1 Burst DRAM Read Page Hit

Figure 55 depicts a CPU burst read page hit to DRAM. The 82434NX decodes the CPU address as a page hit and drives the column address onto the MA[11:0] lines. CAS[7:0]# are then asserted for two CLKs and negated for one CLK. CMR (CPU Memory Read) is driven on the HIG[4:0] lines to enable the memory data to host data path through the LBXs. The PCMC advances the MA[1:0] lines through the processor burst order, negating and asserting CAS[7:0]# to read each Qword. The MD[63:0] data is sampled with HCLK in the LBXs when MDLE is asserted, and driven on the host bus the following cycle to meet the setup time of the CPU. BRDY# is then asserted. When MDLE is negated, the LBX continues to drive the latched HD[63:0] to ensure that the data hold time to CWE[7:0]# is met for standard SRAMs. The LBXs tri-state the host data bus when HIG[4:0] change to NOPC and MDLE rises. A single read page hit from DRAM is similar to the first read of this sequence. The HIG[4:0] lines are driven to NOPC when the last BRDY# is asserted.

The diagram also shows the typical control signal timing for a burst SRAM line fill operation. Note that CCS# inactive will mask any new ADS# (caused by the NA# assertion) to the burst SRAMs.

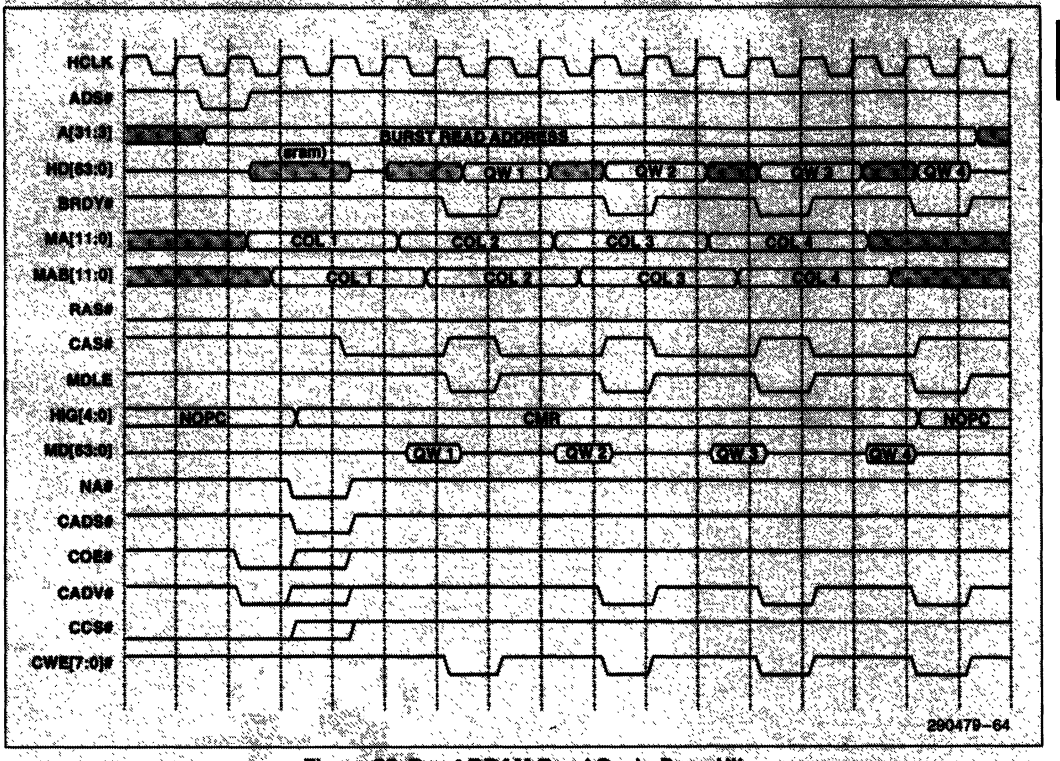


Figure 55. Burst DRAM Read Cycle-Page Hit

6.2.3.2 Burst DRAM Read Page Miss

Figure 56 depicts a CPU to DRAM burst read page miss cycle. The 82434NX decodes the CPU address as a page miss and switches from initially driving the column address to driving the row address on the MA[11:0] lines. RAS# is then negated to precharge the DRAMs and then asserted to latch the new DRAM row address. The PCMC then switches the MA[11:0] lines to drive the column address and asserts CAS[7:0]#. CMR (CPU Memory Read) is driven on the HIG[4:0] lines to enable the memory data to host data path through the LBXs. The PCMC advances the MA[1:0] lines through the microprocessor burst order, negating and asserting CAS[7:0]# to read each Qword. The MD[63:0] data is sampled with HCLK in the LBXs when MDLE is asserted, and driven on the host bus the following cycle to meet the setup time of the CPU. BRDY# is then asserted. When MDLE is negated, the LBX continues to drive the latched HD[63:0] to ensure that the data hold time to CWE[7:0]# is met for standard SRAMs. The LBXs tri-state the host data bus when HIG[4:0] change to NOPC and MDLE rises. The HIG[4:0] lines are driven to NOPC when the last BRDY# is asserted.

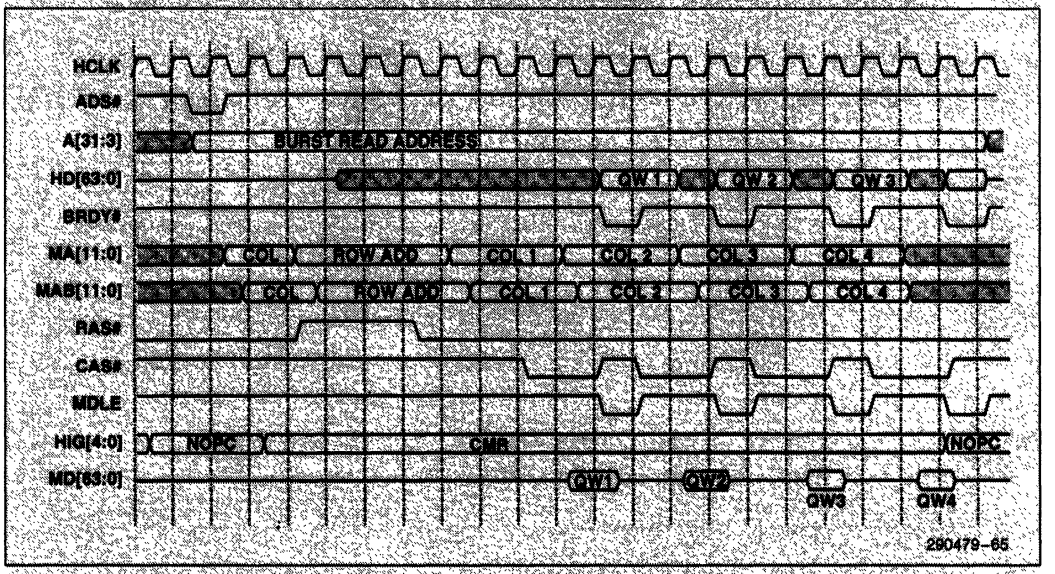


Figure 56. Burst DRAM Read Cycle-Page Miss

6.2.3.3 Burst DRAM Read Row Miss

Figure 57 depicts a CPU to DRAM burst read row miss cycle. The 82434NX decodes the CPU address as a row miss and switches from initially driving the column address to driving the row address on the MA[11:0] lines. The RAS# signal that was asserted is negated and the RAS# for the currently accessed row is asserted (RAS# is asserted 1 clock earlier in 0-Active RAS# Mode.) The PCMC then switches the MA[11:0] lines to drive the column address and asserts CAS[7:0]#. CMR (CPU Memory Read) is driven on the HIG[4:0] lines to enable the memory data to host data path through the LBXs. The PCMC advances the MA[1:0] lines through the microprocessor burst order, negating and asserting CAS[7:0]# to read each Qword. The MD[63:0] data is sampled with HCLK in the LBXs when MDLE is asserted, and driven on the host bus the following cycle to meet the setup time of the CPU. BRDY# is then asserted. When MDLE is negated, the LBX continues to drive the latched HD[63:0] to ensure that the data hold time to CWE[7:0]# is met for standard SRAMs. The LBXs tri-state the host data bus when HIG[4:0] change to NOPC and MDLE rises. A single read row miss from DRAM is similar to the first read of this sequence. The HIG[4:0] lines are driven to NOPC when the last BRDY# is asserted.

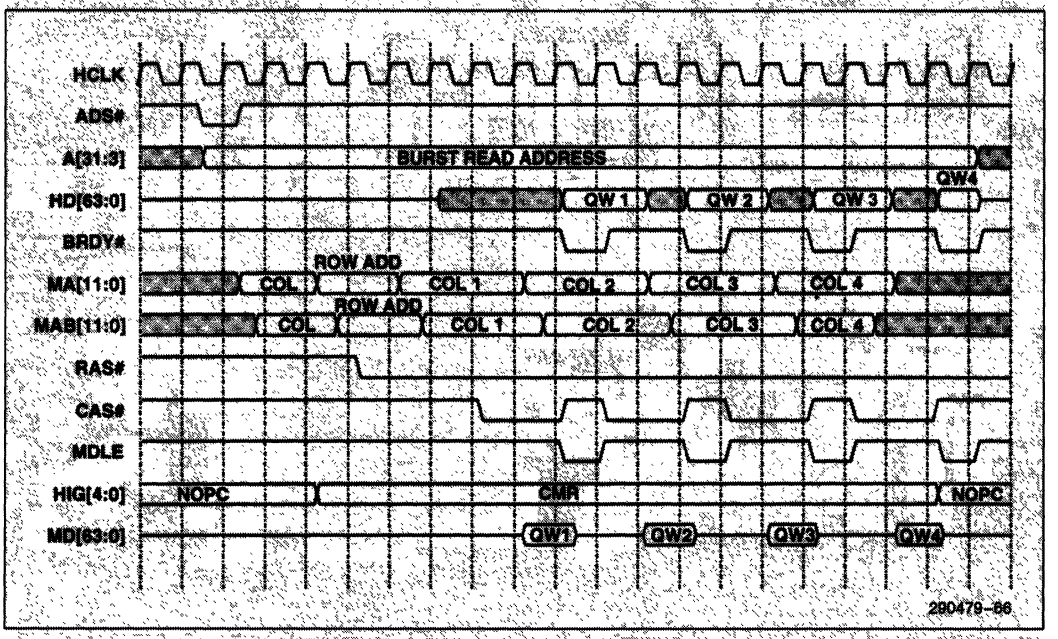


Figure 57. Burst DRAM Read Cycle-Row Miss

6.2.3.4 Burst DRAM Write Page Hit

Figure 58 depicts a CPU burst write page hit to DRAM. The 82434NX decodes the CPU write cycle as a DRAM page hit. The HIG[4:0] lines are driven to PCMWQ to post the write to the LBXs. In the figure, the write cycle is posted to the CPU-to-Memory Posted Write Buffer at 3-1-1-1. When the cycle is decoded as a page hit, the PCMC asserts WE# and drives the RCMWQ command on MIG[2:0] to enable the LBXs to drive the first Qword of the write onto the memory data lines. MEMDRV is then driven to cause the LBXs to continue to drive the first Qword for two more clocks. CAS[7:0]# are then negated and asserted to perform the writes to the DRAMs as the MA[1:0] lines advance through the Pentium processor burst order. A single write is similar to the first write of the burst sequence. The MIG[2:0] lines are driven to NOPM in the clock when the last CAS[7:0]# are asserted.

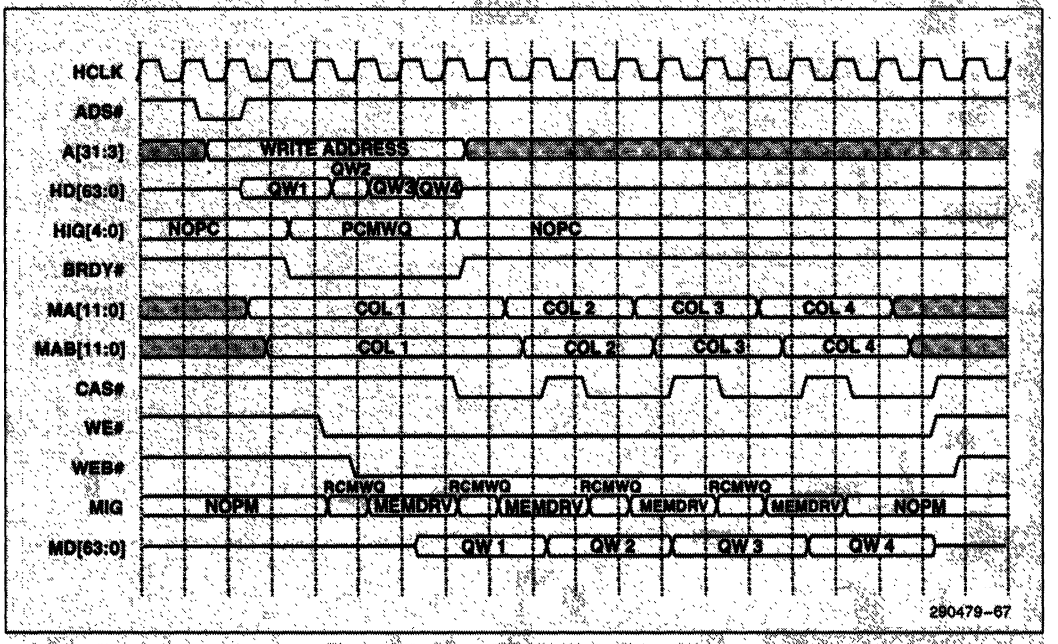


Figure 58. Burst DRAM Write Page Miss

6.2.3.5 Burst DRAM Write Page Miss

Figure 59 depicts a CPU burst write page miss to DRAM. The 82434NX decodes the CPU write cycle as a DRAM page miss and drives the PCMWQ command [HIG[4:0] lines] to post the write data to the LBXs. In the figure, the write cycle is posted to the CPU-to-Memory Posted Write Buffer at 3-1-1-1. When the cycle is decoded as a page miss, the PCMC switches the MA[11:0] lines from the column address to the row address and asserts WE# in clock 4. The PCMC drives the RCMWQ command on MIG[2:0] to enable the LBXs to drive the first Qword of the write onto the memory data lines. MEMDRV is then driven to cause the LBXs to continue to drive the first Qword. The RAS# signal for the currently decoded row is negated to precharge the DRAMs. RAS# is then asserted to cause the DRAMs to latch the row address. The PCMC then switches the MA[11:0] lines to the column address and asserts CAS[7:0]# to initiate the first write. CAS[7:0]# are then negated and asserted to perform the writes to the DRAMs as the MA[1:0] lines advance through the Pentium processor burst order. A single write is similar to the first write of the burst sequence. The MIG[2:0] lines are driven to NOPM in the clock when the last CAS[7:0]# are asserted.

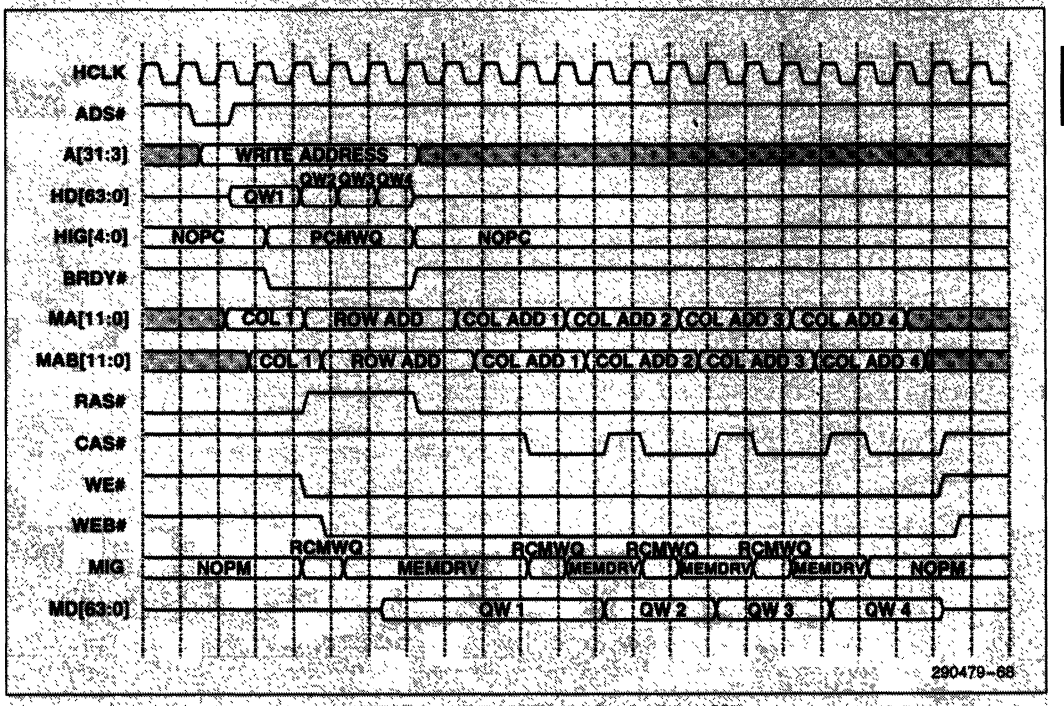


Figure 59. Burst DRAM Write Cycle-Page Miss

2

6.2.3.6 Burst DRAM Write Row Miss

Figure 60 depicts a CPU burst write row miss to DRAM. The 82434NX decodes the CPU write cycle as a DRAM row miss and the HIG[4:0] lines are driven to PCMWQ to post the write data into LBXs. When the cycle is decoded as a row miss, the PCMC negates the already active RAS# signal, switches the MA[11:0] lines from the column address to the row address and asserts the RAS# signal for the currently decoded row. The PCMC asserts WE# and drives the RCMWQ command on MIG[2:0] to enable the LBXs to drive the first Qword of the write onto the memory data lines. MEMDRV is then driven to cause the LBXs to continue to drive the first Qword. The PCMC then switches the MA[11:0] lines to the column address and asserts CAS[7:0]# to initiate the first write. CAS[7:0]# are then negated and asserted to perform the writes to the DRAMs as the MA[1:0] lines advance through the microprocessor burst order. A single write is similar to the first write of the burst sequence. The MIG[2:0] lines are driven to NOPM in the clock when the last CAS[7:0]# are asserted.

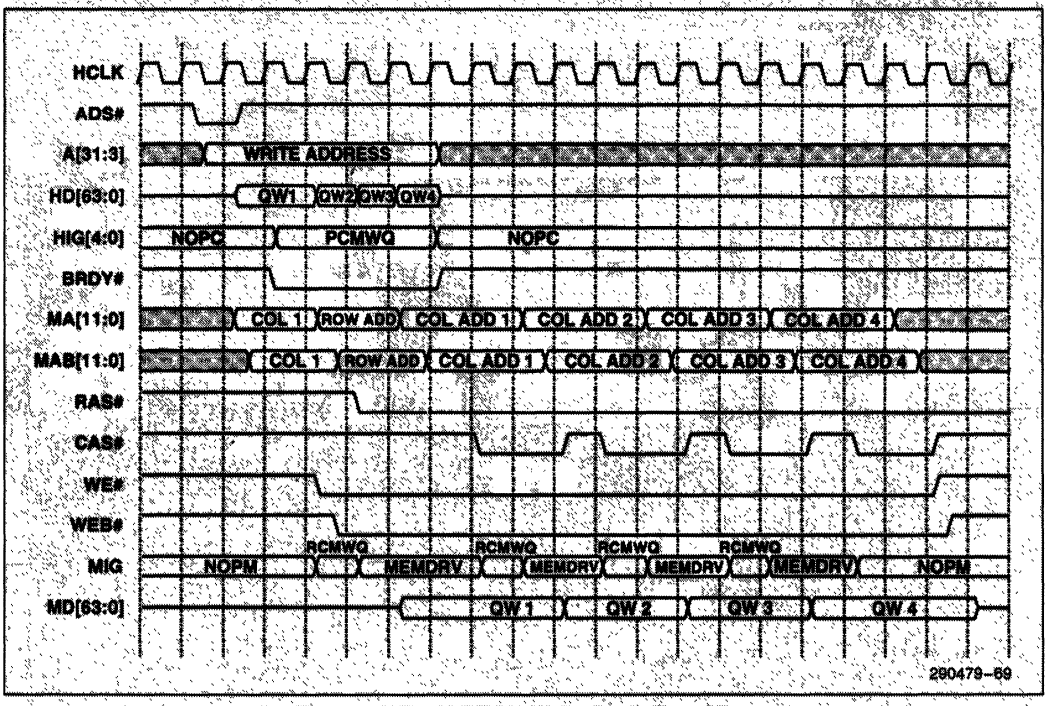


Figure 60. Burst DRAM Write Cycle-Row Miss

6.2.4 REFRESH

The refresh of the DRAM array can be performed by either using RAS#-only or CAS#-before-RAS# refresh cycles. When programmed for CAS#-before-RAS# refresh, hidden refresh cycles are initiated when possible. The timing of internally generated refresh cycles is derived from HCLK and is independent of any expansion bus refresh cycles.

The DRAM controller contains an internal refresh timer which periodically requests the refresh control logic to perform either a single refresh or a burst of four refreshes. The single refresh interval is 15.6 µs. The interval for burst of four refreshes is four times the single refresh interval, or 62.4 µs. The PCMC is configured for either single or burst of four refresh and either RAS#-only or CAS#-before RAS# refresh via the DRAM Control Register (offset 57h).

To minimize performance impact, refresh cycles are partially deferred until the DRAM interface is idle. Refresh cycles are initiated such that the RAS# maximum active time is never violated.

Hidden refresh cycles are run whenever all eight CAS# lines are active at the end of a read transaction when the refresh cycle is internally requested. Normal CAS#-before-RAS# refresh cycles are run whenever the DRAM interface is idle when the refresh is requested, or when any subset of the CAS# lines is inactive as the refresh is internally requested.

To minimize the power surge for refreshing a large DRAM array, the DRAM interface staggers the assertion and negation of the RAS# signals during both CAS#-before-RAS# and RAS#-only refresh cycles. The order of RAS# edges is dependent on which RAS# was most recently asserted prior to the refresh sequence. The RAS# that was active will be the last to be activated during the refresh sequence. All RAS[7:0]# lines are negated at the end of refresh cycles, making the first DRAM cycle after a refresh sequence a row miss.

6.2.4.1 RAS#-Only Refresh—Single

Figure 61 depicts a RAS#-only refresh cycle when the 82434NX is programmed for single refresh cycles. The diagram shows a cycle completing as the refresh timer inside the PCMC generates a refresh request. The refresh address is driven on the MA[11:0] lines. Since the cycle was to row 0, RAS0# is negated. RAS1# is the first to be asserted. RAS2# through RAS7# are then asserted sequentially while RAS0# is driven high, precharging the DRAMs in row 0. RAS0# is then asserted after RAS7#. Each RAS# line is asserted for eight host clocks.

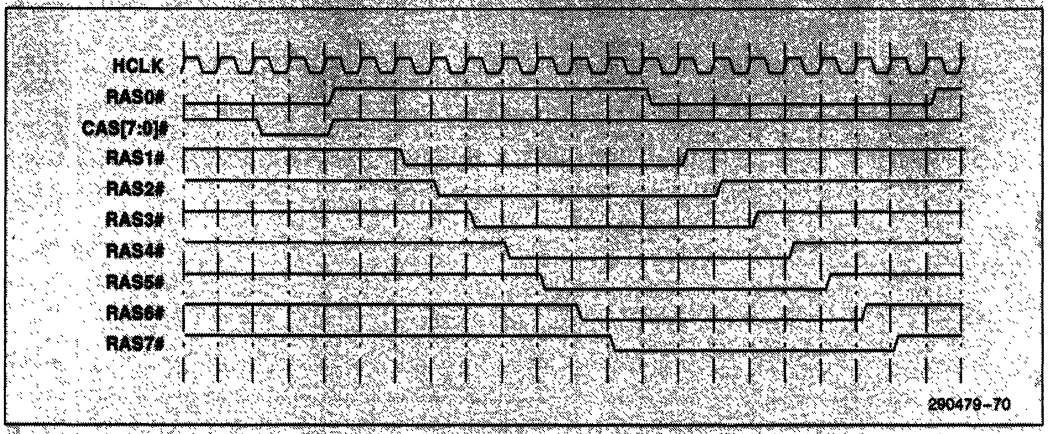


Figure 61. RAS#-Only Refresh—Single

6.2.4.2 CAS#-before-RAS# Refresh—Single

Figure 62 depicts a CAS#-before-RAS# refresh cycle when the 82434NX is programmed for single refresh cycles. The diagram shows a write cycle completing as the refresh timer inside the PCMC generates a refresh request. The cycle is less than a Qword, therefore a hidden refresh is not initiated. After the cycle completes, all of the RAS# and CAS# lines are negated. The PCMC then asserts CAS[7:0]# and then sequentially asserts the RAS# lines, starting with RAS1# since RAS0# was the last RAS# line asserted. Each RAS# line is asserted for eight clocks.

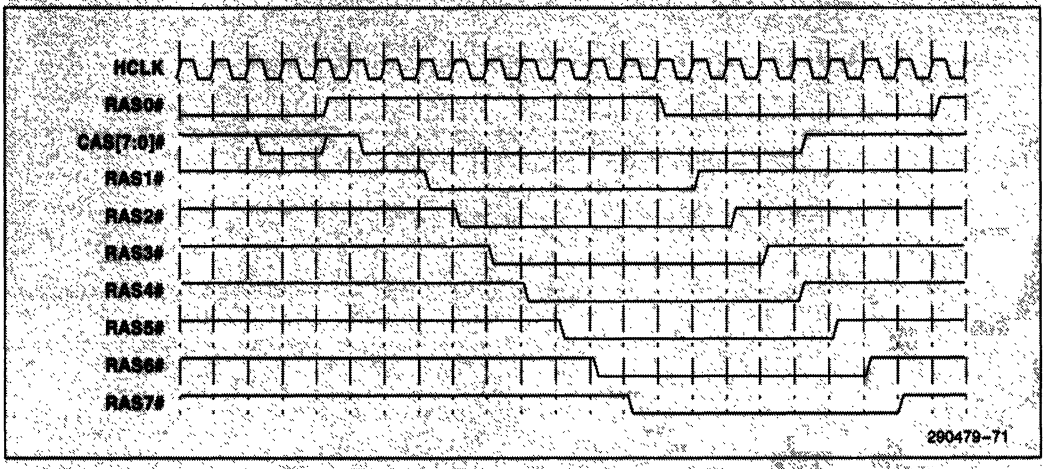


Figure 62. CAS#-Before-RAS# Refresh—Single

6.2.4.3 Hidden Refresh-Single

Figure 63 depicts a hidden refresh cycle which takes place after a DRAM read page hit cycle. The diagram shows a read cycle completing as the refresh timing inside the 82434NX PCMC generates a refresh request. The cycle is an entire Qword, therefore, a hidden refresh is initiated. After the cycle completes, RAS# is negated, but all eight CAS# lines remain asserted. The PCMC then sequentially asserts the RAS# lines, starting with RAS1# since RAS0# was the last active RAS# line. Each RAS# line is asserted for eight clocks.

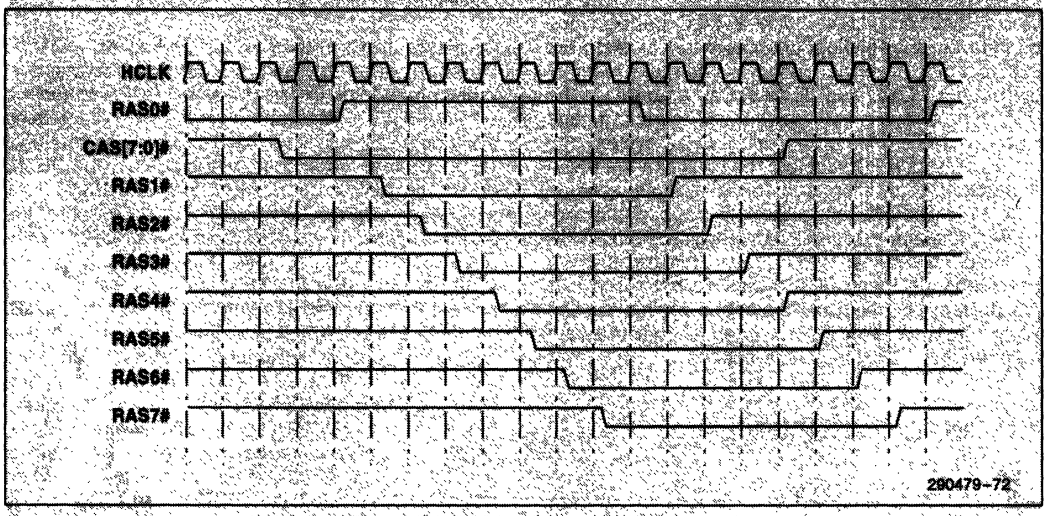


Figure 63. Hidden Refresh—Single

7.0 PCI INTERFACE

The description in this section applies to both the 82434LX and 82434NX.

7.1 PCI Interface Overview

The PCMC and LBXs form a high performance bridge from the Pentium processor to PCI and from PCI to main memory. During PCI-to-main memory cycles, the PCMC and LBXs act as a target on the PCI Bus, allowing PCI masters to read from and write to main memory. During CPU cycles, the PCMC acts as a PCI master. The CPU can then read and write I/O, memory and configuration spaces on PCI. When the CPU accesses I/O mapped and configuration space mapped PCMC registers, the PCMC intercepts the cycles and does not forward them to PCI. Although these CPU cycles do not result in a PCI bus cycle, they are described in this section since most of the PCMC internal registers are mapped into PCI configuration space.

7.2 CPU-to-PCI Cycles

7.2.1 CPU WRITE TO PCI

Figure 64 depicts a series of CPU memory writes which are posted to PCI. The CPU initiates the cycles by asserting ADS# and driving the memory address onto the host address lines. The PCMC asserts NA# in the clock after ADS# allowing the Pentium processor to drive another cycle onto the host bus two clocks later. The PCMC decodes the memory address and drives PCPWL on the HIG[4:0] lines, posting the host address bus and the low Dword of the data bus to the LBXs. The PCMC asserts BRDY#, terminating the CPU cycle with one wait state. Since NA# is asserted in the second clock of the first cycle, the Pentium processor does not insert an idle cycle after this cycle completes, but immediately drives the next cycle onto the bus. Thus, the Pentium processor maximum Dword write bandwidth of 89 MBytes/second is achieved during back-to-back Dword writes cycles. Each of the following write cycles is posted to the LBXs in three clocks.

In this example, the PCMC is parked on PCI and therefore, does not need to arbitrate for the bus. When parked, the PCMC drives the SCPA command on the PIG[3:0] lines and asserts DRVPCI, causing the host address lines to be driven on the PCI AD[31:0] lines. After the write is posted, the PCMC drives the DCPWA command on the PIG[3:0] lines to drive the previously posted address onto the AD[31:0] lines. The PCMC then drives DCPWD onto the PIG[3:0] lines, to drive the previously posted write data onto the AD[31:0] lines. As this is occurring on PCI, the second write cycle is being posted on the host bus. In this case, the second write is to a sequential and incrementing address. Thus, the PCMC leaves FRAME# asserted, converting the write cycle into a PCI burst cycle. The PCMC continues to drive the DCPWD command on the PIG[3:0] lines. The LBXs advance the posted write buffer pointer to point to the next posted Dword when DCPWD is sampled on PIG[3:0] and TRDY# is sampled asserted. Therefore, if the target inserts a wait-state by negating TRDY#, the LBXs continue to drive the data for the current transfer. The remaining writes are posted on the host bus, while the PCMC and LBXs complete the writes on PCI.

CPU I/O write cycles to PCI differ from the memory write cycle described here in that I/O writes are never posted. BRDY# is asserted to terminate the cycle only after TRDY# is sampled asserted, completing the cycle on PCI.

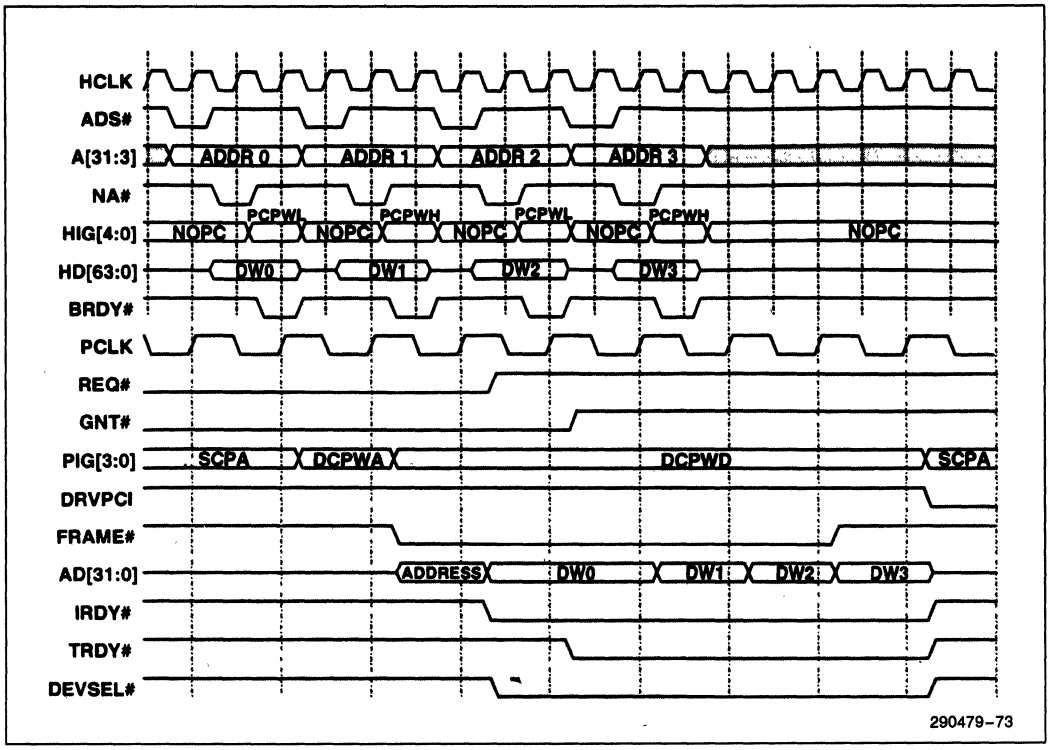


Figure 64. CPU Memory Writes to PCI

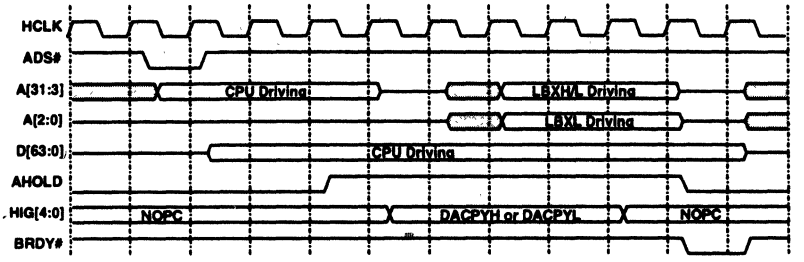
7.3 Register Access Cycles

The PCMC contains two registers which are mapped into I/O space, the Configuration Space Enable Register (I/O port CF8h) and the Turbo-Reset Control Register (I/O port CF9h). All other internal PCMC configuration registers are mapped into PCI configuration space. Configuration space must be enabled by writing a non-zero value to the Key field in the CSE Register before accesses to these registers can occur. These registers are mapped to locations C000h through C0FFh in PCI configuration space. If the Key field is programmed with 0h, CPU I/O cycles to locations C000h through CFFFh are forwarded to PCI as ordinary I/O cycles. Externally, accesses to the I/O mapped registers and the configuration space mapped registers use the same bus transfer protocol. Only the PCMC internal decode of the cycle differs. NA# is never asserted during PCMC configuration register or PCI configuration register access cycles. See Section 3.2, PCI Configuration Space Mapped Registers for details on the PCMC configuration space mapping mechanism.

7.3.1 CPU WRITE CYCLE TO PCMC INTERNAL REGISTER

A write to an internal PCMC register (either CSE Register, TRC Register or a configuration space-mapped register) is shown in Figure 65. The cycle begins with the address, byte enables and status signals (W/R#, D/C# and M/IO#) being driven to a valid state indicating an I/O write to either CF8h to access the CSE register, CF9h to access the TRC Register or C0XXh when configuration space is enabled to access a PCMC internal configuration register. The PCMC decodes the cycle and asserts AHOLD to tri-state the CPU address lines. The PCMC signals the LBXs to copy either the upper Dword or the lower Dword of the data bus onto the address lines. The PCMC makes the decision on which Dword to copy based on the BE[7:0]# lines. The HIG[4:0] lines are driven to DACPYH or DACPYL depending on whether the lower Dword of the data bus or the upper Dword of the data bus needs to be copied onto the address bus. The LBXs sample the HIG[4:0] command, and drive the data onto the address lines. The PCMC samples the A[31:0] lines on the second rising edge of HCLK after the LBXs begin driving the data. Finally, the PCMC negates AHOLD and asserts BRDY#, terminating the cycle.

If the write is to the CSE Register and the Key field is programmed to 0000b then configuration space is disabled. If the Key field is programmed to a non-zero value then configuration space is enabled.



290479-75

Figure 65. CPU Write to a PCMC Configuration Register

7.3.2 CPU READ FROM PCMC INTERNAL REGISTER

A read from an internal PCMC register (either CSE Register, TRC Register or a configuration space-mapped register) is shown in Figure 66. The I/O read cycle is from either CF8h to access the CSE register, CF9h to access the TRC Register or C0XXh when configuration space is enabled to access a configuration space-mapped register. The PCMC decodes the cycle and asserts AHOLD to tri-state the CPU address lines. The PCMC then drives the contents of the addressed register onto the A[31:0] lines. One byte is enabled on each rising HCLK edge for four consecutive clocks. The PCMC signals the LBXs that the current cycle is a read from an internal PCMC register by issuing the ADCPY command to the LBXs over the HIG[4:0] lines. The LBXs sample the HIG[4:0] command and copy the address lines onto the data lines. Finally, the PCMC negates AHOLD, and asserts BRDY# terminating the cycle.

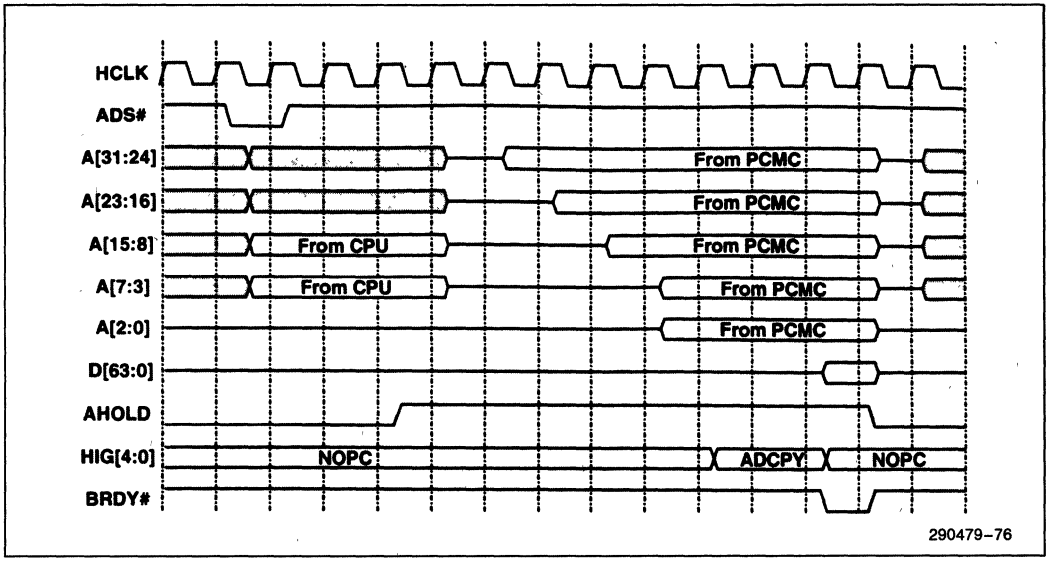


Figure 66. CPU Read from PCMC Configuration Register

7.3.3 CPU WRITE TO PCI DEVICE CONFIGURATION REGISTER

In order to write to or read from a PCI device configuration register the Key field in the CSE register must be programmed to a non-zero value, enabling configuration space. When configuration space is enabled, PCI device configuration registers are accessed by CPU I/O accesses within the range of CnXXh where each PCI device has a unique non-zero value of n. This allows a separate configuration space for each of 15 devices on PCI. Recall that when configuration space is enabled, the PCMC configuration registers are mapped into I/O ports C000h through C0FFh.

A write to a PCI device configuration register is shown in Figure 67. The PCMC internally latches the host address lines and byte enables. The PCMC asserts AHOLD to tri-state the CPU address bus and drives the address lines with the translated address for the PCI configuration cycle. The translation is described in Section 3.2, PCI Configuration Space Mapped Registers. On the HIG[4:0] lines, the PCMC signals the LBXs to latch either the upper Dword of the host data bus or the lower Dword of the host data bus to be driven onto PCI during the data phase of the PCI cycle. On the PIG[3:0] lines, the PCMC signals the LBXs to drive the latched host address lines on the PCI AD[31:0] lines. The upper two bytes of the address lines are used during configuration as IDSEL signals for the PCI devices. The IDSEL pin on each PCI device is connected to one of the AD[31:17] lines.

The PCMC drives the command for a configuration write (1011) onto the C/BE[3:0]# lines and asserts FRAME# for one PCI clock. The PCMC drives the PIG[3:0] lines signaling the LBXs to drive the contents of the PCI write buffer onto the PCI AD[31:0] lines. This command is driven for only one PCI clock before returning to the SCPA command on the PIG[3:0] lines. The LBXs continue to drive the AD[31:0] lines with the valid write data as long as DRVPCI is asserted. The PCMC then asserts IRDY# and waits until sampling the TRDY# signal active. When TRDY# is sampled asserted, the PCMC negates DRVPCI tri-stating the LBX AD[31:0] lines. BRDY# is asserted for one clock to terminate the CPU cycle.

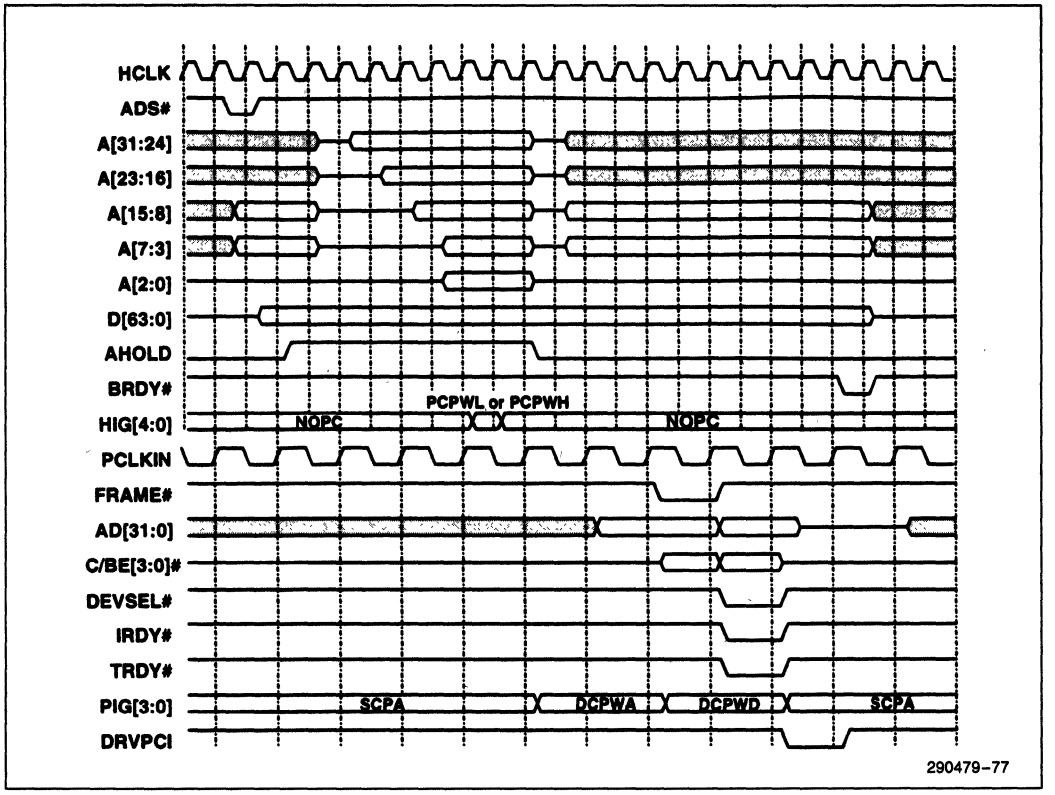


Figure 67. CPU Write to PCI Device Configuration Register

7.3.4 CPU READ FROM PCI DEVICE CONFIGURATION REGISTER

In order to write to or read from a PCI device configuration register the Key field in the CSE register must be programmed to a non-zero value, enabling configuration space. When configuration space is enabled, PCI device configuration registers are accessed by CPU I/O accesses within the range of CnXXh where each PCI device has a unique non-zero value of n. This allows a separate configuration space for each of 15 devices on PCI. Recall that when configuration space is enabled, the PCMC configuration registers occupy I/O addresses C0XXH.

A CPU read from a PCI device configuration register is shown in Figure 68. The PCMC internally latches the host address lines and byte enables. The PCMC asserts AHOLD to tri-state the CPU address bus. The PCMC drives the address lines with the translated address for the PCI configuration cycle. The translation is described in Section 3.2, PCI Configuration Space Mapped Registers. On the PIG[3:0] lines, the PCMC signals the LBXs to drive the latched host address lines on the PCI AD[31:0] lines. The upper two bytes of the address lines are used during configuration as IDSEL signals for the PCI devices. The IDSEL pin on each PCI device is connected to one of the AD[31:17] lines.

The PCMC drives the command for a configuration read (1010) onto the C/BE[3:0]# lines and asserts FRAME# for one PCI clock. The PCMC drives the PIG[3:0] lines signaling the LBXs to latch the data on the PCI AD[31:0] lines into the CPU-to-PCI first read prefetch buffer. The PCMC then drives the HIG[4:0] lines signaling the LBXs to drive the data from the buffer onto the host data lines. The PCMC asserts IRDY# and waits until sampling TRDY# active. After TRDY# is sampled active, BRDY# is asserted for one clock to terminate the CPU cycle.

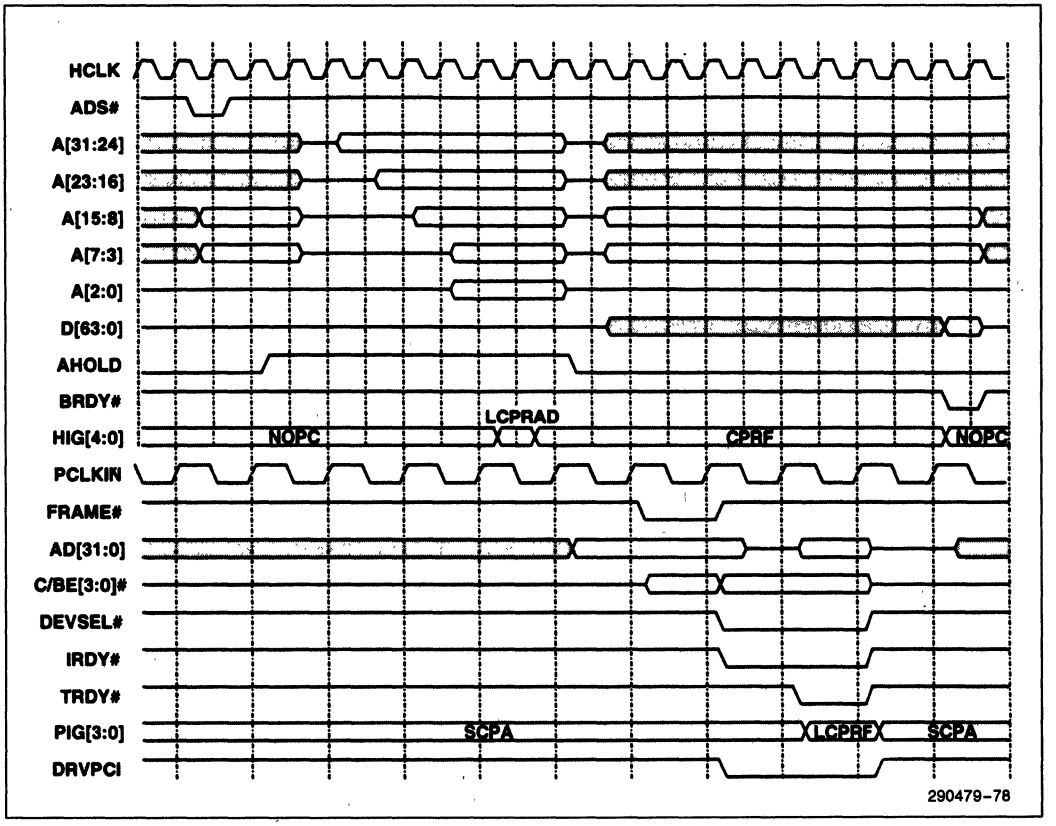


Figure 68. CPU Read from PCI Device Configuration Register

During system initialization, the CPU typically attempts to read from the configuration space of all 15 possible PCI devices to detect the presence of the devices. If no device is present, DEVSEL# is not be asserted and the cycle is terminated, returning FF . . . FFh to the CPU. Figure 69 depicts an attempted read from a configuration register of a non-existent device. If no device responds then the PCMC aborts the cycle and sends the DRVFF command over the HIG[4:0] lines causing the LBXs to drive FF . . . FFh onto the host data lines.

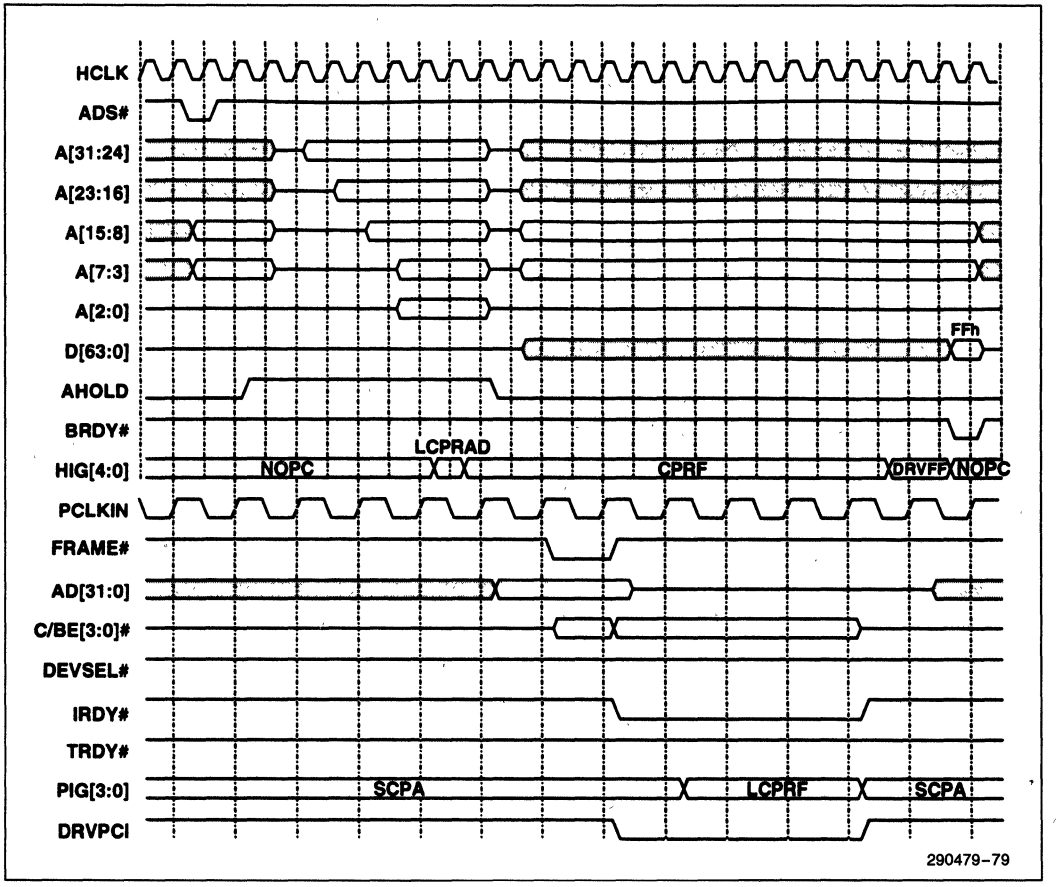


Figure 69. CPU Attempted Configuration Read from Non-Existent PCI Device

7.4 PCI-to-Main Memory Cycles

7.4.1 PCI MASTER WRITE TO MAIN MEMORY

Figure 70 depicts a PCI master burst write to main memory. The PCI master begins by driving the address on the AD[31:0] lines and asserting FRAME#. Upon sampling FRAME# active, the PCMC drives the LCPA command on the PIG[3:0] lines causing the LBXs to retain the address that was latched on the previous PCLK rising edge. The PCMC then samples MEMCS# active, indicating that the cycle is directed to main memory. The PCMC drives the PPMWA command on the PIG[3:0] lines to move the latched PCI address into the write buffer address register. The PCMC then drives the DPWA command on the HIG[4:0] lines enabling the LBXs to drive the PCI master write address onto the host address bus. The PCMC asserts EADS# to initiate a first level cache snoop cycle and simultaneously begins an internal second level cache snoop cycle. Since the snoop is a result of a PCI master write, INV is asserted with EADS#. HITM# remains negated and the snoop either hits an unmodified line or misses in the second level cache, thus no write-back cycles are required. If the snoop hit an unmodified line in either the first or second level cache, the line is invalidated. The cycle is immediately forwarded to the DRAM interface. The four posted Dwords are written to main memory as two Qwords with two CAS[7:0]# cycles. In this example, the DRAM interface is configured for X-3-3-3 write timing, thus each CAS[7:0]# low pulse is two HCLKs in length.

The PCMC disconnects the cycle by asserting STOP# when one of the two four-Dword-deep PCI-to-Memory Posted Write Buffers is full. If the master terminates the cycle before sampling STOP# asserted, then IRDY#, STOP# and DEVSEL# are negated when FRAME# is sampled negated. If the master intended to continue bursting, then the master negates FRAME# when it samples STOP# asserted. IRDY#, STOP# and DEVSEL# are then negated one clock later.

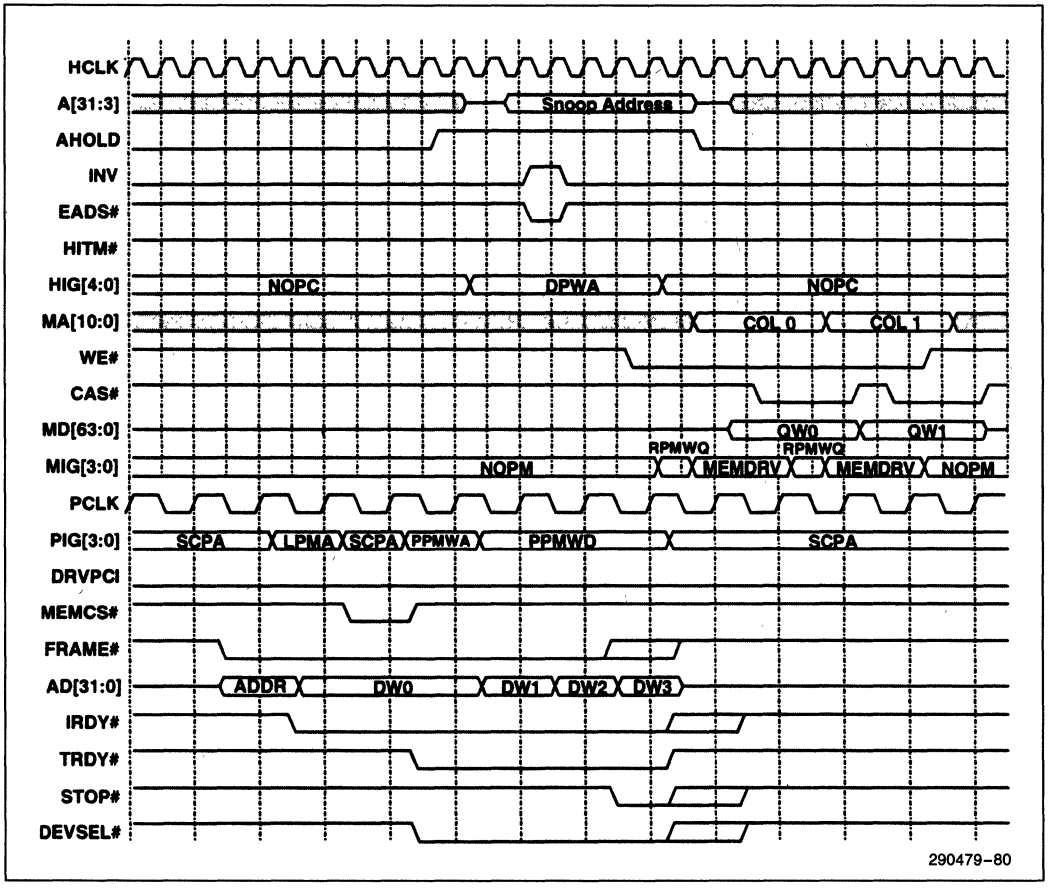


Figure 70. PCI Master Write to Main Memory-Page Hit

7.4.2 PCI MASTER READ FROM MAIN MEMORY

Figure 71 depicts a PCI master read from main memory. The PCI master initiates the cycle by driving the read address on the AD[31:0] lines and asserting FRAME#. The PCMC drives the LPMA command on the PIG[3:0] lines causing the LBXs to retain the address latched on the previous PCLK rising edge. The PCMC drives the DPRA command on the HIG[4:0] lines enabling the LBXs to drive the read address onto the host address lines. The snoop cycle misses in the second level cache and either hits an unmodified line or misses in the first level cache. The cycle is then forwarded to the DRAM interface. A read of four Qwords is performed. Each Qword is posted in the PCI-Memory Read Prefetch Buffer. The data is then driven onto PCI in an eight Dword burst cycle. If the master terminates the cycle before sampling STOP#, then IRDY#, STOP# and DEVSEL# are all negated after FRAME# is sampled inactive. If the master intended to continue bursting, then the master negates FRAME# when it samples STOP# asserted and IRDY#, STOP# and DEVSEL# are negated one clock later.

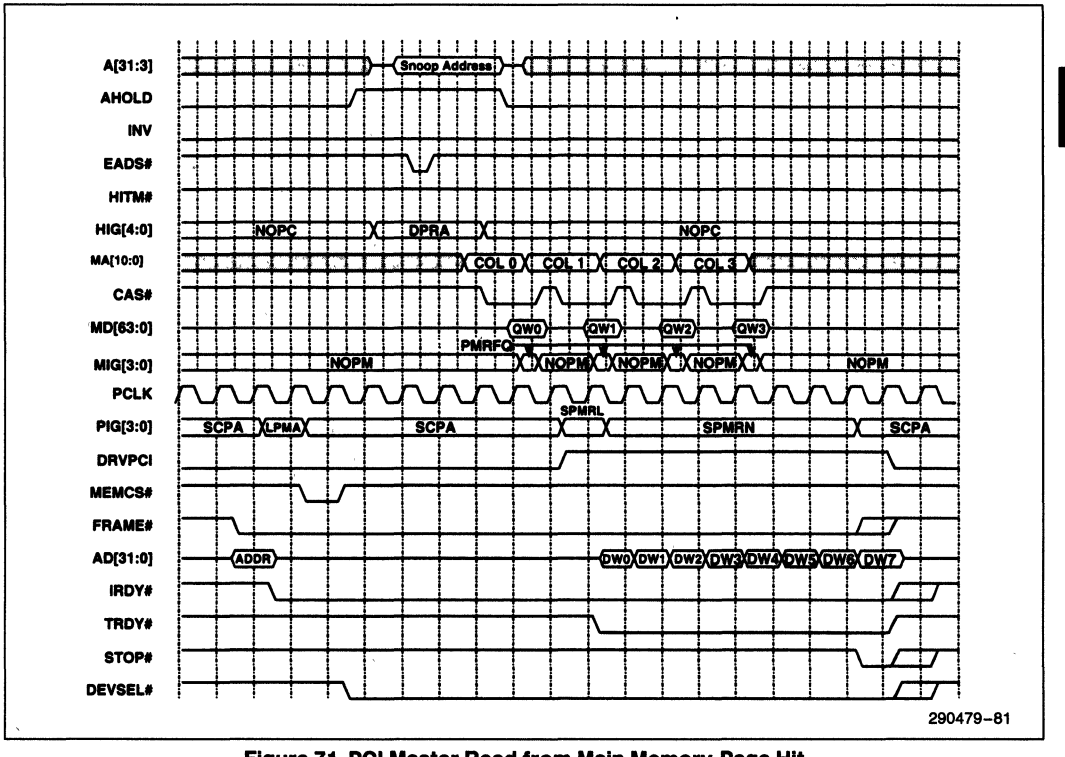


Figure 71. PCI Master Read from Main Memory-Page Hit

8.0 SYSTEM CLOCKING AND RESET

8.1 Clock Domains

The 82434LX and 82434NX PCMCs and 82433LX and 82433NX LBXs operate based on two clocks, HCLK and PCLK. The CPU, second level cache, and the DRAM interfaces operate based on HCLK. The PCI interface timing is based on PCLK.

8.2 Clock Generation and Distribution

Figure 72 shows an example of the 82434LX and 82434NX PCMC host clock distribution in the CPU, cache and memory subsystem. HCLK is distributed to the CPU, PCMC, LBXs and the second level cache SRAMs (in the case of a burst SRAM second level cache).

The host clock originates from an oscillator which is connected to the HCLKOSC input on the PCMC. The PCMC generates six low skew copies of HCLK, HCLKA–HCLKF. Figure 72 shows an example of a host clock distribution scheme for a uni-processor system. In this figure, clock loading is balanced with each HCLK output driving two loads in the system. Each clock output should drive a trace of length *k* with stubs at the end of the trace of length *l* connecting to the two loads. The *l* and *k* parameters should be matched for each of the six clock outputs to minimize overall system clock skew. One of the HCLK outputs is used to clock the PCMC and the Pentium processor. Because the clock driven to the PCMC HCLKIN input and the Pentium processor CLK input originates with the same HCLK output, clock skew between the PCMC and the CPU can be kept lower than between the PCMC and other system components. Another copy of HCLK is used to clock the LBXs. A 256 KByte burst SRAM second level cache can be implemented with eight 32 KByte x 9 synchronous SRAMs. The four remaining copies of HCLK are used to clock the SRAMs. Each HCLK output drives two SRAMs. A 512 KByte second level cache is implemented with four 64 KByte x 18 synchronous SRAMs. Two of the four extra copies are used to clock the SRAMs while the other two are unused. Any one of the HCLK outputs can be used to clock the PCMC and Pentium processor, the two LBXs or any pair of SRAMs. All six copies are identical in drive strength.

Figure 73 depicts the PCI clock distribution.

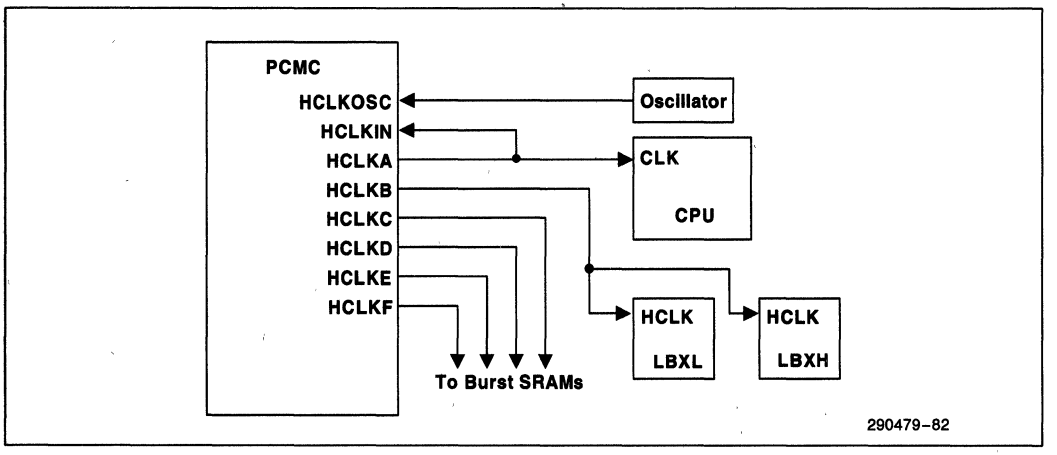


Figure 72. HCLK Distribution Example

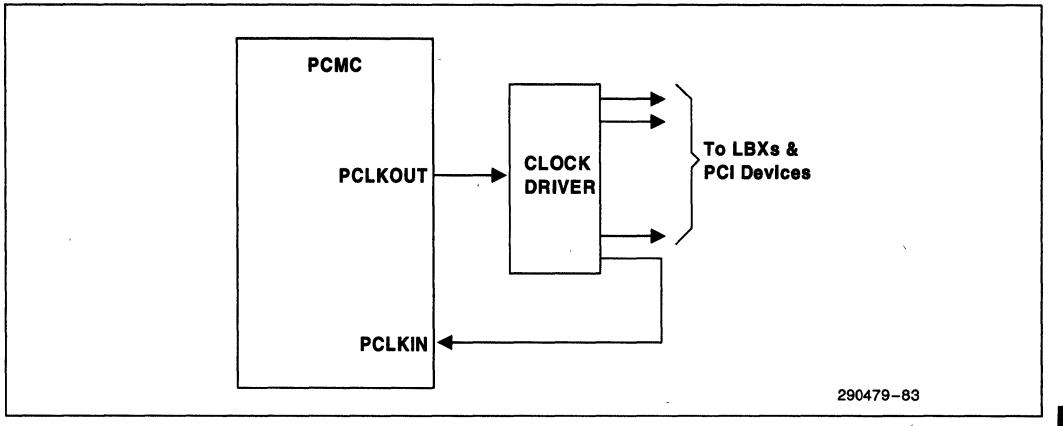


Figure 73. PCI Clock Distribution

The PCMC generates PCLKOUT with an internal Phase Locked Loop (PLL). The PCLKOUT signal is buffered using a single component to produce several low skew copies of PCLK to drive the LBXs and other devices on PCI. One of the outputs of the clock driver is directed back to the PCLKIN input on the PCMC. The PLL locks the rising edges of PCLKIN in phase with the rising edges of HCLKIN. The PLL effectively compensates for the delay of the external clock driver. The resulting PCI clock is one half the frequency of HCLK. Timing for all of the PCI interface signals is based on PCLKIN. All PCI interface inputs are sampled on PCLKIN rising edges and all outputs transition as valid delays from PCLKIN rising edges. Clock skew between the PCLKIN pin on the PCMC and the PCLK pins on the LBXs must be kept within 1.25 ns to guarantee proper operation of the LBXs.

8.3 Phase Locked Loop Circuitry

The 82434LX and 82434NX PCMCs each contain two internal Phase Locked Loops (PLLs). Loop filters and power supply decoupling circuitry must be provided externally. Figure 74 shows the PCMC connections to the external PLL circuitry.

One of the PCMC internal Phase Locked Loops (PLL) locks onto the HCLKIN input. The PLL is used by the PCMC in generating and sampling timing critical signals. An external loop filter is required. The PLLARC1 and PLLARC2 pins connect to the external HCLK loop filter. Two resistors and a capacitor form the loop filter. The loop filter circuitry should be placed as close as possible to the PCMC loop filter pins. The PLL also has dedicated power and ground pins, PLLAVDD, PLLAVSS and PLLAGND. These power pins require a low noise supply. PLLAVDD, PLLAVSS and PLLAGND must be connected to the RC network shown in Figure 74.

The second PCMC internal Phase Locked Loop (PLL) locks the PCLKIN input in phase with the HCLKIN input. The PLL is used by the PCMC to keep the PCI clock in phase with the host clock. An external loop filter is required. The PLLBRC1 and PLLBRC2 pins connect to the external PCLK loop filter. Two resistors and a capacitor form the loop filter. The loop filter circuitry should be placed as close as possible to the PCMC loop filter pins. The PLL also has dedicated power and ground pins, PLLBVDD, PLLBVSS and PLLBGND. These power pins require a low noise supply. PLLBVDD, PLLBVSS and PLLBGND must be connected to the RC network shown in Figure 74.

The resistance and capacitance values for the external PLL circuitry are listed below.

R1 = 10 KΩ ± 5%

R2 = 150Ω ± 5%

R3 = 33Ω ± 5%

C1 = 0.01 μF ± 10%

C2 = 0.47 μF ± 10%

An additional 0.01 μF capacitor in parallel with C2 will help to improve noise immunity.

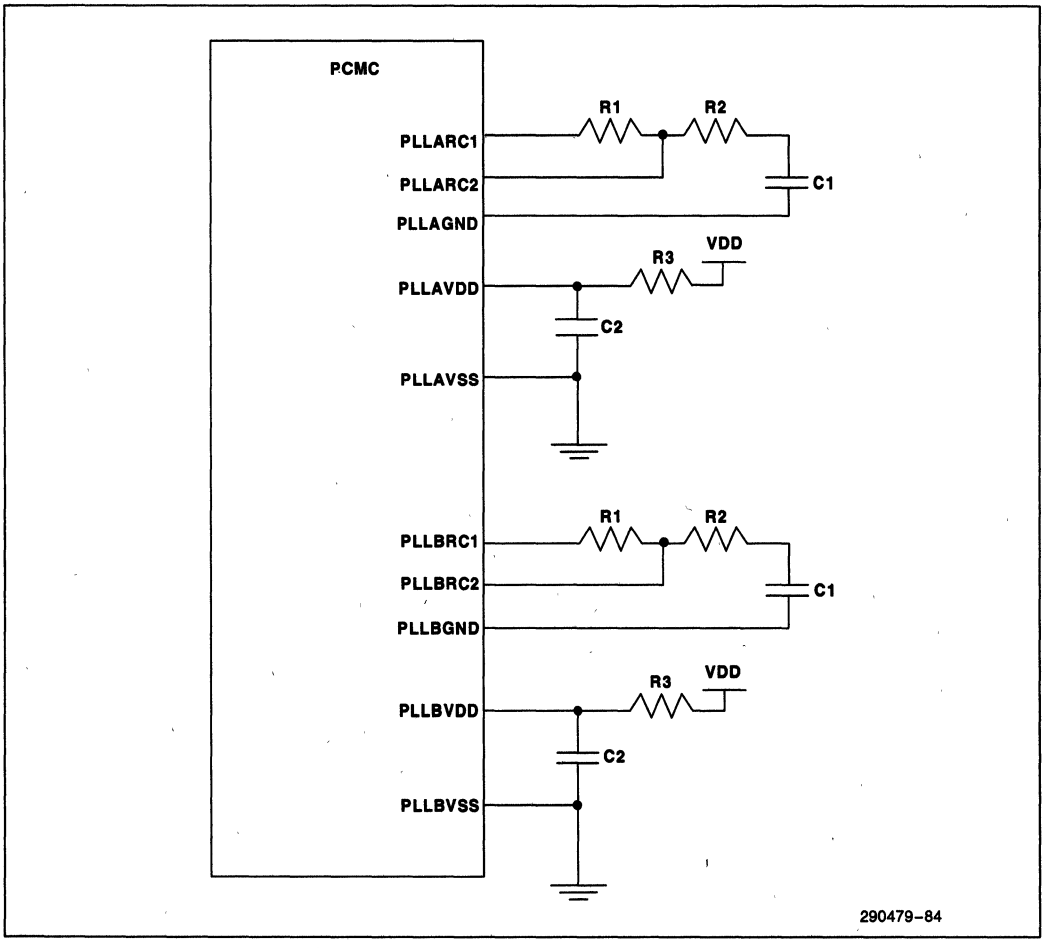


Figure 74. PCMC PLL Circuitry Connections

8.4 System Reset

Figure 75 shows the 82434LX and 82434NX PCMC system reset connections. The 82434LX and 82434NX PCMC reset logic monitors PWROK and generates CPURST, PCIRST# and INIT.

When asserted, PWROK is an indicator to the PCMC that VDD and HCLK have stabilized long enough for proper system operation. CPURST is asserted to initiate hard reset. INIT is asserted to initiate soft reset. PCIRST# is asserted to reset devices on PCI.

Hard reset is initiated by the PCMC in response to one of two conditions. First, hard reset is initiated when power is first applied to the system. PWROK must be driven inactive and must not be asserted until 1 ms after VDD and HCLK have stabilized at their AC and DC specifications. While PWROK is negated, the 82434LX asserts CPURST and PCIRST#. PWROK can be asserted asynchronously. When PWROK is asserted, the 82434LX first ensures that it has been completely initialized before negating CPURST and PCIRST#. CPURST is negated synchronously to the rising edge of HCLK. PCIRST# is negated asynchronously.

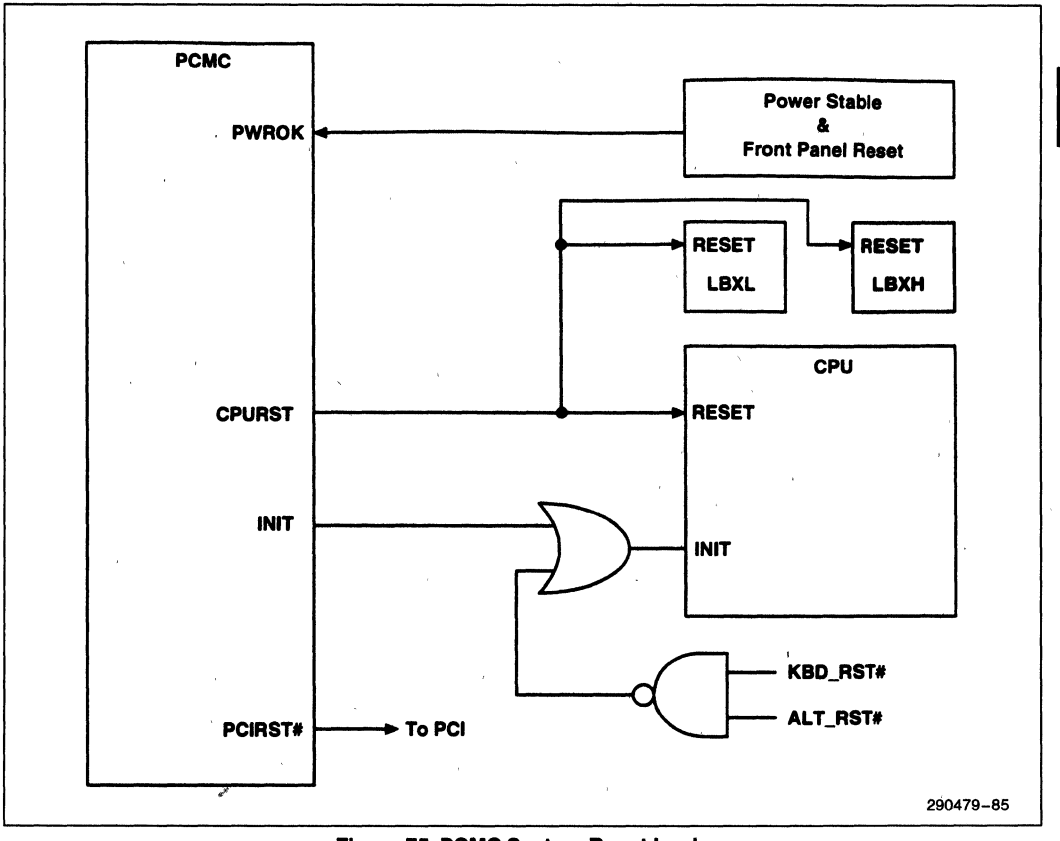


Figure 75. PCMC System Reset Logic

When PWROK is negated, the PCMC asserts AHOLD causing the CPU to tri-state the host address lines. Address lines A[31:29] are sampled by the PCMC 1 ms after the rising edge of PWROK. The values sampled on A[31:30] are inverted inside the PCMC and then stored in Configuration Register 52h bits 7 and 6. The A[31:30] strapping options are depicted in Table 18.

Table 18. A[31:30] Strapping Options

A[31:30]	Configuration Register 52h, Bits[7:6]	Secondary Cache Size
11	00	Not Populated
10	01	Reserved
01	10	256 KByte Cache
00	11	512 KByte Cache

The value sampled on A29 is inverted inside the PCMC and stored in the SRAM Type Bit (bit 5) in the SCC Register. A28 is required to be pulled high for compatibility with future versions of the PCMC.

The PCMC also initiates hard reset when the System Hard Reset Enable bit in the Turbo-Reset Control Register (I/O address CF9h) is set to 1 and the Reset CPU bit toggles from 0 to 1. The PCMC drives CPURST and PCIRST# active for a minimum of 1 ms.

Table 19 shows the state of all 82434LX PCMC output and bi-directional signals during hard reset. During hard reset both CPURST and PCIRST# are asserted. When the hard reset is due to PWROK negation, AHOLD is asserted. The PCMC samples the strapping options on the A[31:29] lines 1 ms after the rising edge of PWROK. When hard reset is initiated via a write to the Turbo-Reset Control Register (I/O port CF9h) AHOLD remains negated throughout the hard reset. Table 19 also applies to the 82434NX, with the exception of the signals listed in Section 8.5, 82434NX Reset Sequencing.

Table 19. 82434LX Output and I/O Signal States During Hard Reset

Signal	State	Signal	State
A[31:0]	Input	IRDY#	Input
AHOLD	High/Low	KEN#	Undefined
BOFF#	High	MA[10:0]	Undefined
BRDY#	High	MDLE	High
CAA[6:3]	Undefined	MEMACK#	High-Z
CAB[6:3]	Undefined	MIG[2:0]	Low
CADS[1:0]#	High	NA#	High
CADV[1:0]#	High	PAR	Input
CALE	High	PEN#	High
CAS[7:0]#	High	PERR#	Input
COE[1:0]#	High	PLOCK#	Input
CWE[7:0]#	High	PIG3	Low
C/BE[3:0]#	Input	PIG[2:0]	High
DEVSEL#	Input	RAS[5:0]#	High
DRVPCI	Low	REQ#	High-Z
EADS#	High	SERR#	Input
FRAME#	Input	STOP#	Input
HIG[4:0]	Low	TRDY#	Input
INIT	Low	WE#	High
INV	Low		

Soft reset is initiated by the PCMC in response to one of two conditions. First, when the System Hard Reset Enable bit in the TRC Register is reset to 0, and the Reset CPU bit toggles from 0 to 1, the PCMC initiates soft reset by asserting INIT for a minimum of 2 HCLKs. Second, the PCMC initiates a soft reset upon detecting a shutdown cycle from the CPU. In this case, the PCMC first broadcasts a shutdown special cycle on PCI and then asserts INIT for a minimum of 2 HCLKs.

8.5 82434NX Reset Sequencing

When PWROK is negated, the 82434NX PCMC drives the following signals low—BRDY#, NA#, AHOLD, EADS#, INV, BOFF#, KEN#, PEN#, CPURST, INIT, CALE, CADS[1:0]#, CADV[1:0]#, CAA[6:3], CAB[6:3], COE[1:0]#, CWE[7:0]#. HCLK[A:F] are driven as soon as the 3.3V supply is active. Note that CWE[7:0]# low prevents the second level cache data RAMs from driving the data bus, even though COE[1:0]# are also driven low. Also, note that BOFF# driven low causes the CPU to tri-state all outputs to the 82434NX PCMC and 82433NX LBX, except HITM#, SMIACT#, and PCHK#. This minimizes the number of signals that the CPU may drive to the PCMC when the 3.3V supply is active and the 5V supply is not active.

Figure 76 shows how the 82434NX sequences CPURST and PCIRST# in response to PWROK assertion.

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

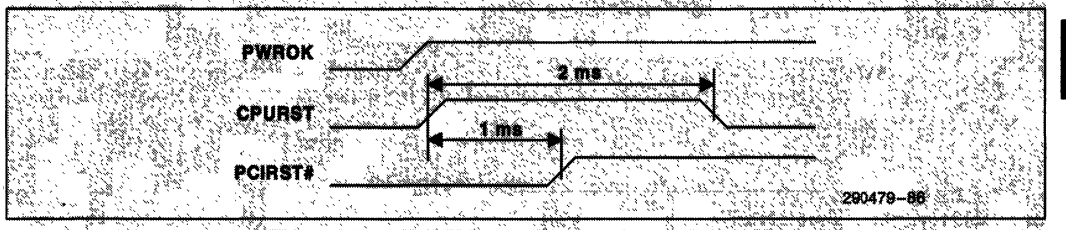


Figure 76. 82434NX Reset Sequencing at Power-Up

2

9.0 ELECTRICAL CHARACTERISTICS

NOTICE: This data sheet contains information on products in the sampling and initial production phases of development. The specifications are subject to change without notice. Verify with your local Intel Sales office that you have the latest data sheet before finalizing a design.

9.1 Absolute Maximum Ratings

Case Temperature under Bias .....0°C to +85°C

Storage Temperature .....−55°C to +150°C

Voltage on Any Pin
with Respect to Ground.....−0.3 to V_{CC} + 0.3V

Supply Voltage
with Respect to V_{SS} .....−0.3 to +6.5V

Maximum Total Power Dissipation 2.0W

Maximum Power Dissipation, $V_{CC}3$ .....470 mW

The Maximum total power dissipation in the 82434NX on the V_{CC} and $V_{CC}3$ pins is 2.0W. The $V_{CC}3$ pins may draw as much as 470 mW, however, total power will not exceed 2.0W.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

9.2 Thermal Characteristics

The 82434LX and 82434NX PCMCs are designed for operation at case temperatures between 0°C and 85°C. The thermal resistances of the package are given in Table 20.

Table 20. PCMC Package Thermal Resistance

Parameter	Air Flow Meters/Second (Linear Feet per Minute)				
	0 (0)	0.5 (98.4)	1.0 (196.9)	2.0 (393.7)	5.0 (984.3)
θ_{JA} (°C/Watt)	31	27	24.5	23	19
θ_{JC} (°C/Watt)	8.6				

9.3 82434LX DC Characteristics

Functional Operating Range (V_{CC} = 5V ±5%; T_{CASE} = 0°C to +85°C)

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{IL1}	Input Low Voltage	−0.3	0.8	V	Note 1, V_{CC} = 4.75V
V_{IH1}	Input High Voltage	2.2	V_{CC} + 0.3	V	Note 1, V_{CC} = 5.25V
V_{IL2}	Input Low Voltage	−0.3	1.35	V	Note 2, V_{CC} = 4.75V
V_{IH2}	Input High Voltage	3.85	V_{CC} + 0.3	V	Note 2, V_{CC} = 5.25V
V_{T1}	Schmitt Trigger Threshold Voltage, Falling Edge	0.7	1.35	V	Note 3, V_{CC} = 5.0V
V_{T1+}	Schmitt Trigger Threshold Voltage, Falling Edge	1.4	2.2	V	Note 3, V_{CC} = 5.0V
V_{H1}	Hysteresis Voltage	0.3	1.2	V	Note 3, V_{CC} = 5.0V
V_{T2-}	Schmitt Trigger Threshold Voltage, Falling Edge	1.25	2.3	V	Note 3, V_{CC} = 5.0V
V_{T2+}	Schmitt Trigger Threshold Voltage, Rising Edge	2.3	3.7	V	Note 3, V_{CC} = 5.0V
V_{H2}	Hysterersis Voltage	0.3	1.2	V	Note 3, V_{CC} = 5.0V

Functional Operating Range (V_{CC} = 5V ±5%; T_{CASE} = 0°C to +85°C) (Continued)

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{OL1}	Output Low Voltage		0.5	V	Note 4
V_{OH1}	Output High Voltage	V_{CC} − 0.5		V	Note 4
V_{OL2}	Output Low Voltage		0.4	V	Note 5
V_{OH2}	Output High Voltage	2.4		V	Note 5
I_{OL1}	Output Low Current		1	mA	Note 6
I_{OH1}	Output High Current	−1		mA	Note 6
I_{OL2}	Output Low Current		3	mA	Note 7
I_{OH2}	Output High Current	−2		mA	Note 7
I_{OL3}	Output Low Current		6	mA	Note 8
I_{OH3}	Output High Current	−2		mA	Note 8
I_{OL4}	Output Low Current		3	mA	Note 9
I_{OH4}	Output High Current	−1		mA	Note 9
I_{IH}	Input Leakage Current		+10	uA	
I_{IL}	Input Leakage Current		−10	uA	
C_{IN}	Input Capacitance		12	pF	F_C = 1 MHz
C_{OUT}	Output Capacitance		12	pF	F_C = 1 MHz
$C_{I/O}$	I/O Capacitance		12	pF	F_C = 1 MHz

NOTES:
1. V_{IL1} and V_{IH1} apply to the following signals: A[31:0], BE[7:0]#, D/C#, W/R#, M/IO#, HLOCK#, ADS#, PCHK#, HITM#, CACHE#, SMIACT#, PCLKIN, HCLKIN, HCLKOSC, FLSHBUF#, MEMCS#, SERR#, PERR#, MEMREQ#, GNT#, PLOCK#, STOP#, IRDY#, TRDY#, FRAME#, C/BE[3:0]#.
2. V_{IL2} and V_{IH2} apply to the following signals: PPOUT[1:0], EOL.
3. V_{T1-}, V_{T1+} and V_{H1} apply to PWROK. V_{T2-}, V_{T2+} and V_{H2} apply to TESTEN.
4. V_{OL1} and V_{OH1} apply to the following signals: HIG[4:0], MIG[2:0], PIG[3:0], DRVPCI, MDLE, PCIRST#.
5. V_{OL2} and V_{OH2} apply to the following signals: REQ#, MEMACK#, FRAME#, C/BE[3:0]#, TRDY#, IRDY#, STOP#, PLOCK#, DEVSEL#, PAR, PERR#, SERR#, BOFF#, AHOLD, BRDY#, NA#, EADS#, KEN#, INV, A[31:0], PCLKOUT, HCLKA–HCLKF, CALE, COE[1:0]#, CWE[7:0]#, CADV[1:0]#, CADS[1:0]#, CAA[6:3], CAB[6:3], RAS[5:0]#, CAS[7:0]#, MA[10:0], WE#.
6. I_{OL1} and I_{OH1} apply to the following signals: HIG[4:0], MIG[2:0], PIG[3:0], DRVPCI, MDLE, PCIRST#.
7. I_{OL2} and I_{OH2} apply to the following signals: C/BE[3:0]#, REQ#, MEMACK#, MA[10:0], WE#.
8. I_{OL3} and I_{OH3} apply to the following signals: FRAME#, TRDY#, IRDY#, STOP#, PLOCK#, DEVSEL#, PAR, PERR#, SERR#.
9. I_{OL4} and I_{OH4} apply to the following signals: BOFF#, AHOLD, BRDY#, NA#, EADS#, KEN#, INV, CPURST, INIT, A[31:0], PCLKOUT, CALE, COE[1:0]#, CADS[1:0]#, CADV[1:0]#, CWE[7:0]#, CAA[6:3], CAB[6:3], RAS[5:0]#, CAS[7:0]#.

9.4 82434NX DC Characteristics

Functional Operating Range (V_{CC} = 5V ±5%; $V_{CC}3$ = 3.135 to 3.465 V; T_{CASE} = 0°C to +85°C)

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{IL1}	Input Low Voltage	−0.3	0.8	V	Note 1, V_{CC} = 4.75V
V_{IH1}	Input High Voltage	2.2	V_{CC} + 0.3	V	Note 1, V_{CC} = 5.25V
V_{IL2}	Input Low Voltage	−0.3	1.35	V	Note 2, V_{CC} = 4.75V
V_{IH2}	Input High Voltage	3.85	V_{CC} + 0.3	V	Note 2, V_{CC} = 5.25V
V_{IL3}	Input Low Voltage	−0.3	0.8	V	Note 3, V_{ccq} = 3.135V
V_{IH3}	Input High Voltage	2.2	V_{CC} + 0.3	V	Note 3, V_{ccq} = 3.465V
V_{T1-}	Schmitt Trigger Threshold Voltage, Falling Edge	0.7	1.35	V	Note 4, V_{CC} = 5.0V
V_{T1+}	Schmitt Trigger Threshold Voltage, Rising Edge	1.4	2.2	V	Note 4, V_{CC} = 5.0V
V_{H1}	Hysteresis Voltage	0.3	1.2	V	Note 4, V_{CC} = 5.0V
V_{T2-}	Schmitt Trigger Threshold Voltage, Falling Edge	1.25	2.3	V	Note 4, V_{CC} = 5.0V
V_{T2+}	Schmitt Trigger Threshold Voltage, Rising Edge	2.3	3.7	V	Note 4, V_{CC} = 5.0V
V_{H2}	Hysterersis Voltage	0.3	1.2	V	Note 4, V_{CC} = 5.0V
V_{OL1}	Output Low Voltage		0.5	V	Note 5
V_{OH1}	Output High Voltage	V_{CC} − 0.5		V	Note 5
V_{OL2}	Output Low Voltage		0.4	V	Note 6
V_{OH2}	Output High Voltage	2.4		V	Note 6
I_{OL1}	Output Low Current		1	mA	Note 7
I_{OH1}	Output High Current	−1		mA	Note 7
I_{OL2}	Output Low Current		3	mA	Note 8

Functional Operating Range (V_{CC} = 5V ±5%; $V_{CC}3$ = 3.135 to 3.465 V; T_{CASE} = 0°C to +85°C) (Continued)

Symbol	Parameter	Min	Max	Unit	Test Conditions
I_{OH2}	Output High Current	−2		mA	Note 8
I_{OL3}	Output Low Current		6	mA	Note 9
I_{OH3}	Output High Current	−2		mA	Note 9
I_{OL4}	Output Low Current		3	mA	Note 10
I_{OH4}	Output High Current	−1		mA	Note 10
I_{IH}	Input Leakage Current		+10	uA	
I_{IL}	Input Leakage Current		−10	uA	
C_{IN}	Input Capacitance		12	pF	F_C = 1 MHz
C_{OUT}	Output Capacitance		12	pF	F_C = 1 MHz
$C_{I/O}$	I/O Capacitance		12	pF	F_C = 1 MHz

NOTES:

1. V_{IL1} and V_{IH1} apply to the following signals: BE[7:0]#, D/C#, W/R#, M/IO#, HLOCK#, ADS#, PCHK#, HITM#, CACHE#, SMIACT#, PCLKIN, HCLKOSC, FLSHBUF#, MEMCS#, SERR#, PERR#, MEMREQ#, GNT#, PLOCK#, STOP#, IRDY#, TRDY#, FRAME#, C/BE[3:0]#.
2. V_{IL2} and V_{IH2} apply to the following signals: PPOUT[1:0], EOL.
3. VIL3 and VIH3 apply to the following signals: A[31:0], HCLKIN.
4. V_{T1-}, V_{T1+} and V_{H1} apply to PWRGK. V_{T2-}, V_{T2+} and V_{H2} apply to TESTEN.
5. V_{OL1} and V_{OH1} apply to the following signals: HIG[4:0], MIG[2:0], PIG[3:0], DRVPCI, MDLE, PCIRST#.
6. V_{OL2} and V_{OH2} apply to the following signals: REQ#, MEMACK#, FRAME#, C/BE[3:0]#, TRDY#, IRDY#, STOP#, PLOCK#, DEVSEL#, PAR, PERR#, SERR#, BOFF#, AHOLD, BRDY#, NA#, EADS#, KEN#, INV, A[31:0], PCLKOUT, HCLKA–HCLKF, CALE, COE[1:0]#, CWE[7:0]#, CADV[1:0]#, CADS[1:0]#, CAA[6:3], CAB[6:3], RAS[7:0]#, CAS[7:0]#, MA[11:0], WE#.
7. I_{OL1} and I_{OH1} apply to the following signals: HIG[4:0], MIG[2:0], PIG[3:0], DRVPCI, MDLE, A[31:8], A[2:0], PCIRST#.
8. I_{OL2} and I_{OH2} apply to the following signals: C/BE[3:0]#, REQ#, MEMACK#, MA[11:0], WE#.
9. I_{OL3} and I_{OH3} apply to the following signals: FRAME#, TRDY#, IRDY#, STOP#, PLOCK#, DEVSEL#, PAR, PERR#, SERR#.
10. I_{OL4} and I_{OH4} apply to the following signals: BOFF#, AHOLD, BRDY#, NA#, EADS#, KEN#, INV, CPURST, INIT, A[7:3], PCLKOUT, CALE, COE[1:0]#, CADS[1:0]#, CADV[1:0]#, CWE[7:0]#, CAA[6:3], CAB[6:3], RAS[7:0]#, CAS[7:0]#.
11. The output buffers for BRDY#, NA#, AHOLD, EADS#, INV, BOFF#, KEN#, PEN#, CPURST, INIT, CALE, CADS[1:0], CADV[1:0]#, CAA[6:3], CAB[6:3], COE[1:0]#, CWE[7:0]#, A[31:3] AND HCLK[A:F] are powered with $V_{CC}3$ and therefore drive 3.3V signal levels.

9.5 82434LX AC Characteristics

The AC characteristics given in this section consist of propagation delays, valid delays, input setup requirements, input hold requirements, output float delays, output enable delays, output-to-output delays, pulse widths, clock high and low times and clock period specifications. Figure 77 through Figure 85 define these specifications. Section 9.5 lists the 82434LX AC Characteristics. Output test loads are listed in the right column.

In Figure 77 through Figure 85, VT = 1.5V for the following signals:

A[31:0], BE[7:0]#, PEN#, D/C#, W/R#, M/IO#, HLOCK#, ADS#, PCHK#, HITM#, EADS#, BRDY#, BOFF#, AHOLD, NA#, KEN#, INV, CACHE#, SMIACT#, INIT, CPURST, CALE, CADV[1:0]#, COE[1:0]#, CWE[7:0]#, CADS[1:0]#, CAA[6:3], CAB[6:3], WE#, RAS[5:0]#, CAS[7:0]#, MA[10:0], C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, PLOCK#, GNT#, DEVSEL#, MEMREQ#, PAR, PERR#, SERR#, REQ#, MEMCS#, FLSHBUF#,MEMACK#, PWROK, HCLKIN, HCLKA–HCLKF, PCLKIN, PCLKOUT.

VT = 2.5V for the following signals:

PPOUT[1:0], EOL, HIG[4:0], PIG[3:0], MIG[2:0], DRVPCI, MDLE, PCIRST#.

9.5.1 HOST CLOCK TIMING, 66 MHz (82434LX)

Functional Operating Range (V_{CC} = 4.9V to 5.25V; T_{CASE} = 0°C to +70°C)

Symbol	Parameter	Min	Max	Figure	Notes
t1a	HCLKOSC High Time	6.0		82	
t1b	HCLKOSC Low Time	5.0		82	
t2a	HCLKIN Period	15	20	82	
t2b	HCLKIN Period Stability		±100		ps(1)
t2c	HCLKIN High Time	4		82	
t2d	HCLKIN Low Time	4		82	
t2e	HCLKIN Rise Time		1.5	83	
t2f	HCLKIN Fall Time		1.5	83	
t3a	HCLKA–HCLKF Output-to-Output Skew		0.5	85	0 pF
t3b	HCLKA–HCLKF High Time	5.0		85	0 pF
t3c	HCLKA–HCLKF Low Time	5.0		85	0 pF

NOTE:
1. Measured on rising edge of adjacent clocks at 1.5V.

9.5.2 CPU INTERFACE TIMING, 66 MHz (82434LX)

Functional Operating Range (V_{CC} = 4.9V to 5.25V; T_{CASE} = 0°C to +70°C)

Symbol	Parameter	Min	Max	Fig	Notes
t10a	ADS#, HITM#, W/R#, M/IO#, D/C#, HLOCK#, CACHE#, BE[7:0]#, SMIACT# Setup Time to HCLKIN Rising	4.6		79	
t10b	ADS#, HITM#, W/R#, M/IO#, D/C#, HLOCK#, CACHE#, BE[7:0]#, SMIACT# Hold Time from HCLKIN Rising	0.8		79	
t11a	PCHK# Setup Time to HCLKIN Rising	4.3		79	
t11b	PCHK# Hold Time from HCLKIN Rising	1.1		79	
t12a	A[18:3] Rising Edge Setup Time to HCLKIN Rising	4.5		79	Setup to HCLKIN rising when ADS# is sampled active by PCMC.
t12aa	A[18:3] Falling Edge Setup Time to HCLKIN Rising	3.2		79	Setup to HCLKIN Rising when ADS# is Sampled Active by PCMC.
t12ab	A[18:3] Rising Edge Setup Time to HCLKIN Rising	4.7			Setup to HCLKIN Rising when ADS# is Sampled Active by PCMC.
t12ac	A[18:3] Falling Edge Setup Time to HCLKIN Rising	4.1			Setup to HCLKIN Rising when ADS# is Sampled Active by PCMC.
t12b	A[31:0] Hold Time from HCLKIN Rising	0.5		79	Hold from HCLKIN rising two clocks after ADS# is sampled active by PCMC.
t12c	A[31:0] Setup Time to HCLKIN Rising	6.5		79	Setup to HCLKIN rising when EADS# is sampled active by the CPU.
t12d	A[31:0] Hold Time from HCLKIN Rising	1.5		79	Hold from HCLKIN rising when EADS# is sampled active by the CPU.

Functional Operating Range (V_{CC} = 4.9V to 5.25V; T_{CASE} = 0°C to +70°C) (Continued)

Symbol	Parameter	Min	Max	Fig	Notes
t12e	A[31:0] Output Enable from HCLKIN Rising	0	13	81	
t12f	A[31:0] Valid Delay from HCLKIN Rising	1.3	13	78	0 pF
t12g	A[31:0] Float Delay from HCLKIN Rising	0	13	80	
t12h	A[2:0] Propagation Delay from BE[7:0]#	1	16	77	0 pF
t13a	BRDY# Rising Edge Valid Delay from HCLKIN Rising	1.7	7.8	78	0 pF
t13b	BRDY# Falling Edge Valid Delay from HCLKIN Rising	1.7	7.6	78	0 pF
t14	NA# Valid Delay from HCLKIN Rising	1.3	7.8	78	0 pF
t15a	AHOLD Valid Delay from HCLKIN Rising	1.3	7.1	78	0 pF
t15b	BOFF# Valid Delay from HCLKIN Rising	1.8	7.1	78	
t16a	EADS#, INV, PEN# Valid Delay from HCLKIN Rising	1.3	7.4	78	0 pF
t16b	CPURST Rising Edge Valid Delay from HCLKIN Rising	0.9	7.5	78	
t16c	CPURST Falling Edge Valid Delay from HCLKIN Rising	0.9	7.0	78	
t16d	KEN# Valid delay from HCLKIN Rising	1.3	7.6	78	
t17	INIT High Pulse Width	2 HCLKs		84	Soft reset via TRC register or CPU shutdown special cycle
t18	CPURST High Pulse Width	1 ms		84	Hard reset via TRC register, 0 pF

9.5.3 SECOND LEVEL CACHE STANDARD SRAM TIMING, 66 MHz (82434LX)

Functional Operating Range (V_{CC} = 4.9V to 5.25V; T_{CASE} = 0°C to +70°C)

Symbol	Parameter	Min	Max	Fig	Notes
t20a	CAA[6:3]/CAB[6:3] Propagation Delay from A[6:3]	0	8.5	77	0 pF
t20b	CAA[6:3]/CAB[6:3] Valid Delay from HCLKIN Rising	0	7.2	78	0 pF
t21a	COE[1:0]# Falling Edge Valid Delay from HCLKIN Rising	0	9	78	0 pF
t21b	COE[1:0]# Rising Edge Valid Delay from HCLKIN Rising	0	5.5	78	0 pF
t22a	CWE[7:0]#/CBS[7:0]# Falling Edge Valid Delay from HCLKIN Rising	2	14	78	CPU burst or single write to second level cache, 0 pF
t22b	CWE[7:0]#/CBS[7:0]# Rising Edge Valid Delay from HCLKIN Rising	3	14	78	CPU burst or single write to second level cache, 0 pF
t22c	CWE[7:0]#/CBS[7:0]# Valid Delay from HCLKIN Rising	1.4	7.7	78	Cache line Fill, 0 pF
t22d	CWE[7:0]#/CBS[7:0]# Low Pulse Width	1 HCLK		84	0 pF
t22e	CWE[7:0]#/CBS[7:0]# Driven High before CALE Driven High	-1		85	Last write to second level cache during cache line fill, 0 pF
t22f	CAA[4:3]/CAB[4:3] Valid before CWE[7:0]# Falling	1.5		85	CPU burst write to second level cache, 0 pF
t23	CALE Valid Delay from HCLKIN Rising	0	7.5	78	0 pF
t24	CR/W[1:0]# Valid Delay from HCLKIN Rising	1.5	7.6	78	0 pF
t25	CBS[1:0]# Valid Delay from HCLKIN Rising; Reads from Cache SRAMs	1.0	12.0	78	0 pF

9.5.4 SECOND LEVEL CACHE BURST SRAM TIMING, 66 MHz (82434LX)

Functional Operating Range (V_{CC} = 4.9V to 5.25V; T_{CASE} = 0°C to +70°C)

Symbol	Parameter	Min	Max	Fig	Notes
t30a	CAA[6:3]/CAB[6:3] Propagation Delay from A[6:3]	0	8.5	77	0 pF
t30b	CAA[6:3]/CAB[6:3] Valid Delay from HCLKIN Rising	0	7.0	78	0 pF
t31	CADS[1:0]# Valid Delay from HCLKIN Rising	1.5	7.7	78	0 pF
t32	CADV[1:0]# Valid Delay from HCLKIN Rising	1.5	7.1	78	0 pF
t33	CWE[7:0]# Valid Delay from HCLKIN Rising	1.0	9.0	78	0 pF
t34a	COE[1:0]# Falling Edge Valid Delay from HCLKIN Rising	0	9.0	78	0 pF
t34b	COE[1:0]# Rising Edge Valid Delay from HCLKIN Rising	0	5.5	78	0 pF
t35	CALE Valid Delay from HCLKIN Rising	0	7.5	78	0 pF

9.5.5 DRAM INTERFACE TIMING, 66 MHz (82434LX)

Functional Operating Range (V_{CC} = 4.9V to 5.25V; T_{CASE} = 0°C to +70°C)

Symbol	Parameter	Min	Max	Fig	Notes
t40a	RAS[5:0]# Valid Delay from HCLKIN Rising	0	7.5	78	50 pF
t40b	RAS[5:0]# Pulse Width High	4 HCLKs − 5		84	RAS# precharge at beginning of page miss cycle, 50 pF
t41a	CAS[7:0]# Valid Delay from HCLKIN Rising	0	7.5	78	50 pF
t41b	CAS[7:0]# Pulse Width High	1 HCLKIN − 5		84	CAS# precharge during burst cycles, 50 pF
t42	WE# Valid Delay from HCLKIN Rising	0	21	78	50 pF
t43a	MA[10:0] Propagation Delay from A[23:3]	0	23	77	50 pF
t43b	MA[10:0] Valid Delay from HCLKIN Rising	0	10.1	78	50 pF

9.5.6 PCI CLOCK TIMING, 66 MHz (82434LX)

Functional Operating Range (V_{CC} = 4.9V to 5.25V; T_{CASE} = 0°C to +70°C)

Symbol	Parameter	Min	Max	Fig	Notes
t50a	PCLKOUT High Time	13		82	20 pF
t50b	PCLKOUT Low Time	13		82	20 pF
t51a	PCLKIN High Time	12		82	
t51b	PCLKIN Low Time	12		82	
t51c	PCLKIN Rise Time		3	83	
t51d	PCLKIN Fall Time		3	83	

9.5.7 PCI INTERFACE TIMING, 66 MHz (82434LX)

Functional Operating Range (V_{CC} = 4.9V to 5.25V; T_{CASE} = 0°C to +70°C)

Symbol	Parameter	Min	Max	Fig	Notes
t60a	C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, PLOCK#, PAR, PERR#, SERR#, DEVSEL# Valid Delay from PCLKIN Rising	2	11	78	Min: 0 pF Max: 50 pF
t60b	C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, PLOCK#, PAR, PERR#, SERR#, DEVSEL# Output Enable Delay from PCLKIN Rising	2		81	
t60c	C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, PLOCK#, PAR, PERR#, SERR#, DEVSEL# Float Delay from PCLKIN Rising	2	28	80	
t60d	C/BE[3:0]#, FRAME#, PLOCK#, PAR, PERR#, SERR#, Setup Time to PCLKIN Rising	7		79	
t60da	TRDY#, IRDY# Setup Time to PCLKIN Rising	8.1		77	
t60db	STOP#, DEVSEL# Setup Time to PCLKIN Rising	8.5		77	
t60e	C/BE[3:0]#, FRAME#, PLOCK#, PAR, PERR#, SERR# Hold Time from PCLKIN Rising	0		77	
t61a	REQ#, MEMACK# Valid Delay from PCLKIN Rising	2	12	78	Min: 0 pF Max: 50 pF
t61b	REQ#, MEMACK# Output Enable Delay from PCLKIN Rising	2		81	
t61c	REQ#, MEMACK# Float Delay from PCLKIN Rising	2	28	80	
t62a	FLSHREQ#, MEMREQ# Setup Time to PCLKIN Rising	12		79	
t62b	FLSHREQ#, MEMREQ# Hold Time from PCLKIN Rising	0		79	
t63a	GNT# Setup Time to PCLKIN Rising	10		79	
t63b	GNT# Hold Time from PCLKIN Rising	0		79	
t64a	MEMCS# Setup Time to PCLKIN Rising	7		79	
t64b	MEMCS# Hold Time from PCLKIN Rising	0		79	
t65	PCIRST# Low Pulse Width	1 ms		84	Hard Reset via TRC Register, 0 pF

9.5.8 LBX INTERFACE TIMING, 66 MHz (82434LX)

Functional Operating Range (V_{CC} = 4.9V to 5.25V; T_{CASE} = 0°C to +70°C)

Symbol	Parameter	Min	Max	Fig	Notes
t70	HIG[4:0] Valid Delay from HCLKIN Rising	0.8	6.5	78	0 pF
t71	MIG[2:0] Valid Delay from HCLKIN Rising	0.9	6.5	78	0 pF
t72	PIG[3:0] Valid Delay from PCLKIN Rising	0.7	10.9	78	0 pF
t73	PCIDRV Valid Delay from PCLKIN Rising	1	13.5	78	0 pF
t74a	MDLE Falling Edge Valid Delay from HCLKIN Rising	0.6	5.6	78	0 pF
t74b	MDLE Rising Edge Valid Delay from HCLKIN Rising	0.6	6.8	85	0 pF
t75a	EOL, PPOUT[1:0] Setup Time to PCLKIN Rising	7.7		79	
t75b	EOL, PPOUT[1:0] Hold Time from PCLKIN Rising	1.0		79	

9.5.9 HOST CLOCK TIMING, 60 MHz (82434LX)

Functional Operating Range (V_{CC} = 4.75V to 5.25V; T_{CASE} = 0°C to +85°C)

Symbol	Parameter	Min	Max	Fig	Notes
t1a	HCLKOSC High Time	6.0		82	
t1b	HCLKOSC Low Time	5.0		82	
t2a	HCLKIN Period	16.66	20	82	
t2b	HCLKIN Period Stability		±100		ps(1)
t2c	HCLKIN High Time	4		82	
t2d	HCLKIN Low Time	4		82	
t2e	HCLKIN Rise Time		1.5	83	
t2f	HCLKIN Fall Time		1.5	83	
t3a	HCLKA–HCLKF Output-to-Output Skew		0.5	85	0 pF
t3b	HCLKA–HCLKF High Time	5.0		82	0 pF
t3c	HCLKA–HCLKF Low Time	5.0		82	0 pF

NOTE:
1. Measured on rising edge of adjacent clocks at 1.5V.

9.5.10 CPU INTERFACE TIMING, 60 MHz (82434LX)

Functional Operating Range (V_{CC} = 4.75V to 5.25V; T_{CASE} = 0°C to +85°C)

Symbol	Parameter	Min	Max	Fig	Notes
t10a	ADS#, HITM#, W/R#, M/IO#, D/C#, HLOCK#, CACHE#, BE[7:0]#, SMIACT# Setup Time to HCLKIN Rising	4.6		79	
t10b	ADS#, HITM#, W/R#, M/IO#, D/C#, HLOCK#, CACHE#, BE[7:0]#, SMIACT# Hold Time from HCLKIN Rising	1.1		79	
t11a	PCHK# Setup Time to HCLKIN Rising	4.3		79	
t11b	PCHK# Hold Time from HCLKIN Rising	1.1		79	
t12a	A[18:3] Rising Edge Setup Time to HCLKIN Rising	4.5		79	Setup to HCLKIN rising when ADS# is sampled active by PCMC.
t12aa	A[18:3] Falling Edge Setup Time to HCLKIN Rising	3.2		79	Setup to HCLKIN Rising when ADS# is Sampled Active by PCMC.
t12ab	A[18:3] Rising Edge Setup Time to HCLKIN Rising	4.7		79	Setup to HCLKIN Rising when ADS# is Sampled Active by PCMC.
t12ac	A[18:3] Falling Edge Setup Time to HCLKIN Rising	4.1		79	Setup to HCLKIN Rising when ADS# is Sampled Active by PCMC.
t12b	A[31:0] Hold Time from HCLKIN Rising	0.5		79	Hold from HCLKIN rising two clocks after ADS# is sampled active by PCMC.
t12c	A[31:0] Setup Time to HCLKIN Rising	6.5		79	Setup to HCLKIN rising when EADS# is sampled active by the CPU.
t12d	A[31:0] Hold Time from HCLKIN Rising	1.5		79	Hold from HCLKIN rising when EADS# is sampled active by the CPU.
t12e	A[31:0] Output Enable from HCLKIN Rising	0	13	81	
t12f	A[31:0] Valid Delay from HCLKIN Rising	1.3	13	78	0 pF
t12g	A[31:0] Float Delay from HCLKIN Rising	0	13	80	

Functional Operating Range (V_{CC} = 4.75V to 5.25V; T_{CASE} = 0°C to +85°C) (Continued)

Symbol	Parameter	Min	Max	Fig	Notes
t12h	A[2:0] Propagation Delay from BE[7:0]#	1	16	77	0 pF
t13a	BRDY# Rising Edge Valid Delay from HCLKIN Rising	2.1	7.9	78	0 pF
t13b	BRDY# Falling Edge Valid Delay from HCLKIN Rising	2.1	7.9	78	0 pF
t14	NA# Valid Delay from HCLKIN Rising	1.4	8.4	78	0 pF
t15a	AHOLD Valid Delay from HCLKIN Rising	2.0	7.6	78	0 pF
t15b	BOFF# Valid Delay from HCLKIN Rising	2.0	7.6	78	
t16a	EADS#, INV, PEN# Valid Delay from HCLKIN Rising	2.0	8.0	78	0 pF
t16b	CPURST Rising Edge Valid Delay from HCLKIN Rising	1.2	7.5	78	
t16c	CPURST Falling Edge Valid Delay from HCLKIN Rising	1.2	7.5	78	
t16d	KEN# Valid delay from HCLKIN Rising	1.7	8.2	78	
t17	INIT High Pulse Width	2 HCLKs		84	Soft reset via TRC register or CPU shutdown special cycle
t18	CPURST High Pulse Width	1 ms		84	Hard reset via TRC register, 0 pF

9.5.11 SECOND LEVEL CACHE STANDARD SRAM TIMING, 60 MHz (82434LX)

Functional Operating Range (V_{CC} = 4.75V to 5.25V; T_{CASE} = 0°C to +85°C)

Symbol	Parameter	Min	Max	Fig	Notes
t20a	CAA[6:3]/CAB[6:3] Propagation Delay from A[6:3]	0	8.5	77	0 pF
t20b	CAA[6:3]/CAB[6:3] Valid Delay from HCLKIN Rising	0	7.2	78	0 pF
t21a	COE[1:0]# Falling Edge Valid Delay from HCLKIN Rising	0	9	78	0 pF
t21b	COE[1:0]# Rising Edge Valid Delay from HCLKIN Rising	0	5.5	78	0 pF
t22a	CWE[7:0]#/CBS[7:0]# Falling Edge Valid Delay from HCLKIN Rising	2	14	78	CPU burst or single write to second level cache, 0 pF
t22b	CWE[7:0]#/CBS[7:0]# Rising Edge Valid Delay from HCLKIN Rising	3	15	78	CPU burst or single write to second level cache, 0 pF
t22c	CWE[7:0]#/CBS[7:0]# Valid Delay from HCLKIN Rising	1.4	7.7	78	Cache line Fill, 0 pF
t22d	CWE[7:0]#/CBS[7:0]# Low Pulse Width	1 HCLK		84	0 pF
t22e	CWE[7:0]#/CBS[7:0]# Driven High before CALE Driven High	−1		85	Last write to second level cache during cache line fill, 0 pF
t22f	CAA[4:3]/CAB[4:3] Valid before CWE[7:0]# Falling	1.5		85	CPU burst write to second level cache, 0 pF
t23	CALE Valid Delay from HCLKIN Rising	0	8	78	0 pF
t24	CR/W[1:0]# Valid Delay from HCLKIN Rising	1.5	8.2	78	0 pF
t25	CBS[1:0]# Valid Delay from HCLKIN Rising; Reads from Cache SRAMs	1.0	12.0	78	0 pF

9.5.12 SECOND LEVEL CACHE BURST SRAM TIMING, 60 MHz (82434LX)

Functional Operating Range (V_{CC} = 4.75V to 5.25V; T_{CASE} = 0°C to +85°C)

Symbol	Parameter	Min	Max	Fig	Notes
t30a	CAA[6:3]/CAB[6:3] Propagation Delay from A[6:3]	0	8.5	77	0 pF
t30b	CAA[6:3]/CAB[6:3] Valid Delay from HCLKIN Rising	0	8.2	78	0 pF
t31	CADS[1:0]# Valid Delay from HCLKIN Rising	1.5	8.2	78	0 pF
t32	CADV[1:0]# Valid Delay from HCLKIN Rising	1.5	8.2	78	0 pF
t33	CWE[7:0]# Valid Delay from HCLKIN Rising	1.0	10.5	78	0 pF
t34a	COE[1:0]# Falling Edge Valid Delay from HCLKIN Rising	0	9.5	78	0 pF
t34b	COE[1:0]# Rising Edge Valid Delay from HCLKIN Rising	0	6.0	78	0 pF
t35	CALE Valid Delay from HCLKIN Rising	0	8.5	78	0 pF

9.5.13 DRAM INTERFACE TIMING, 60 MHz (82434LX)

Functional Operating Range (V_{CC} = 4.75V to 5.25V; T_{CASE} = 0°C to +85°C)

Symbol	Parameter	Min	Max	Fig	Notes
t40a	RAS[5:0]# Valid Delay from HCLKIN Rising	0	8.0	78	50 pF
t40b	RAS[5:0]# Pulse Width High	4 HCLKs−5		84	RAS# precharge at beginning of page miss cycle, 50 pF
t41a	CAS[7:0]# Valid Delay from HCLKIN Rising	0	8.0	78	50 pF
t41b	CAS[7:0]# Pulse Width High	1 HCLK−5		84	CAS# precharge during burst cycles, 50 pF
t42	WE# Valid Delay from HCLKIN Rising	0	21	78	50 pF
t43a	MA[10:0] Propagation Delay from A[23:3]	0	23	77	50 pF
t43b	MA[10:0] Valid Delay from HCLKIN Rising	0	10.7	78	50 pF

9.5.14 PCI CLOCK TIMING, 60 MHz (82434LX)

Functional Operating Range (V_{CC} = 4.75V to 5.25V; T_{CASE} = 0°C to +85°C)

Symbol	Parameter	Min	Max	Fig	Notes
t50a	PCLKOUT High Time	13		82	20 pF
t50b	PCLKOUT Low Time	13		82	20 pF
t51a	PCLKIN High Time	12		82	
t51b	PCLKIN Low Time	12		82	
t51c	PCLKIN Rise Time		3	83	
t51d	PCLKIN Fall Time		3	83	

9.5.15 PCI INTERFACE TIMING, 60 MHz (82434LX)

Functional Operating Range (V_{CC} = 4.75V to 5.25V; T_{CASE} = 0°C to +85°C)

Symbol	Parameter	Min	Max	Fig	Notes
t60a	C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, PLOCK#, PAR, PERR#, SERR#, DEVSEL# Valid Delay from PCLKIN Rising	2	11	78	Min: 0 pF Max: 50 pF
t60b	C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, PLOCK#, PAR, PERR#, SERR#, DEVSEL# Output Enable Delay from PCLKIN Rising	2		81	
t60c	C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, PLOCK#, PAR, PERR#, SERR#, DEVSEL# Float Delay from PCLKIN Rising	2	28	80	
t60d	C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, PLOCK#, PAR, PERR#, SERR#, DEVSEL# Setup Time to PCLKIN Rising	9		79	
t60e	C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, PLOCK#, PAR, PERR#, SERR#, DEVSEL# Hold Time from PCLKIN Rising	0		79	
t61a	REQ#, MEMACK# Valid Delay from PCLKIN Rising	2	12	78	Min: 0 pF Max: 50 pF
t61b	REQ#, MEMACK# Output Enable Delay from PCLKIN Rising	2		81	
t61c	REQ#, MEMACK# Float Delay from PCLKIN Rising	2	28	80	
t62a	FLSHREQ#, MEMREQ# Setup Time to PCLKIN Rising	12		79	

Functional Operating Range (V_{CC} = 4.75V to 5.25V; T_{CASE} = 0°C to +85°C) (Continued)

Symbol	Parameter	Min	Max	Fig	Notes
t62b	FLSHREQ#, MEMREQ# Hold Time from PCLKIN Rising	0		79	
t63a	GNT# Setup Time to PCLKIN Rising	10		79	
t63b	GNT# Hold Time from PCLKIN Rising	0		79	
t64a	MEMCS# Setup Time to PCLKIN Rising	7		79	
t64b	MEMCS# Hold Time from PCLKIN Rising	0		79	
t65	PCIRST# Low Pulse Width	1 ms		84	Hard Reset via TRC Register, 0 pF

9.5.16 LBX INTERFACE TIMING, 60 MHz (82434LX)

Functional Operating Range (V_{CC} = 4.75V to 5.25V; T_{CASE} = 0°C to +85°C)

Symbol	Parameter	Min	Max	Fig	Notes
t70	HIG[4:0] Valid Delay from HCLKIN Rising	0.8	6.7	78	0 pF
t71	MIG[2:0] Valid Delay from HCLKIN Rising	0.9	6.5	78	0 pF
t72	PIG[3:0] Valid Delay from PCLKIN Rising	1.5	12	78	0 pF
t73	PCIDRV Valid Delay from PCLKIN Rising	1	13	78	0 pF
t74a	MDLE Falling Edge Valid Delay from HCLKIN Rising	0.6	6.8	78	0 pF
t74b	MDLE Rising Edge Valid Delay from HCLKIN Rising	0.6	6.8	85	0 pF
t75a	EOL, PPOUT[1:0] Setup Time to PCLKIN Rising	7.7		79	
t75b	EOL, PPOUT[1:0] Hold Time from PCLKIN Rising	1.0		79	

9.6 82434NX AC Characteristics

The AC characteristics given in this section consist of propagation delays, valid delays, input setup requirements, input hold requirements, output float delays, output enable delays, output-to-output delays, pulse widths, clock high and low times and clock period specifications. Figure 77 through Figure 85 define these specifications. Output test loads are listed in the right column.

In Figure 77 through Figure 85, VT = 1.5V for the following signals:

A[31:0], BE[7:0]#, PEN#, D/C#, W/R#, M/IO#, HLOCK#, ADS#, PCHK#, HITM#, EADS#, BRDY#, BOFF#, AHOLD, NA#, KEN#, INV, CACHE#, SMIACT#, INIT, CPURST, CALE, CADV[1:0]#, COE[1:0]#, CWE[7:0]#, CADS[1:0]#, CAA[6:3], CAB[6:3], WE#, RAS[5:0]#, CAS[7:0]#, MA[10:0], C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, PLOCK#, GNT#, DEVSEL#, MEMREQ#, PAR, PERR#, SERR#, REQ#, MEMCS#, FLSHBUF#, MEMACK#, PWROK, HCLKIN, HCLKA–HCLKF, PCLKIN, PCLKOUT,

VT = 2.5V for the following signals:

PPOUT[1:0], EOL, HIG[4:0], PIG[3:0], MIG[2:0], DRVPCI, MDLE, PCIRST#

2

9.6.1 HOST CLOCK TIMING, 66 MHz (82434NX), PRELIMINARY

Functional Operating Range (V_{CC} = 4.75V to 5.25V; $V_{CC}3$ = 3.135V to 3.465V; T_{CASE} = 0°C to +85°C)

Symbol	Parameter	Min	Max	Fig	Notes
t1a	HCLKOSC High Time	6.0		82	
t1b	HCLKOSC Low Time	5.0		82	
t2a	HCLKIN Period	15	20	82	
t2b	HCLKIN Period Stability		±100		ps(1)
t2c	HCLKIN High Time	4		82	
t2d	HCLKIN Low Time	4		82	
t2e	HCLKIN Rise Time		1.5	83	
t2f	HCLKIN Fall Time		1.5	83	
t3a	HCLKA–HCLKF Output-to-Output Skew		0.5	85	0 pF
t3b	HCLKA–HCLKF High Time	5.0		82	0 pF
t3c	HCLKA–HCLKF Low Time	5.0		82	0 pF

NOTES:
1. Measured on rising edge of adjacent clocks at 1.5V.

9.6.2 CPU INTERFACE TIMING, 66 MHz (82434NX), PRELIMINARY

Functional Operating Range (V_{CC} = 4.75V to 5.25V; $V_{CC}3$ = 3.135V to 3.465V; T_{CASE} = 0°C to +85°C)

Symbol	Parameter	Min	Max	Fig	Notes
t10a	ADS#, W/R#, Setup Time to HCLKIN Rising	4.6		79	
t10b	BE[7:0]# Setup Time to HCLKIN Rising	4.6		79	
t10c	HITM# Setup Time to HCLKIN Rising	6.4		79	
t10d	CACHE#, M/IO# Setup Time to HCLKIN Rising	4.6		79	
t10e	D/C# Setup Time to HCLKIN Rising	4.0		79	
t10f	HLOCK#, SMIACT#, Setup Time to HCLKIN Rising	4.0		79	
t10g	HITM#, M/IO#, D/C#, Hold Time from HCLKIN Rising	0.7		79	
t10h	W/R#, HLOCK#, Hold Time from HCLKIN Rising	0.8		79	
t10i	ADS#, BE[7:0]# Hold Time from HCLKIN Rising	1.1		79	
t10j	CACHE#, SMIACT# Hold Time from HCLKIN Rising	1.1		79	
t11a	PCHK# Setup Time to HCLKIN Rising	4.3		79	
t11b	PCHK# Hold Time from HCLKIN Rising	1.1		79	
t12a	A[31:0] Setup Time to HCLKIN Rising	2.7		79	Setup to HCLKIN rising when ADS# is sampled active by PCMC.
t12b	A[31:0] Hold Time from HCLKIN Rising	0.5			HOLD from HCLKIN Rising two clocks after ADS# is sampled active by PCMC.
t12c	A[31:0] Setup Time to HCLKIN Rising	6.0		79	Setup to HCLKIN rising when EADS# is sampled active by the CPU.

Functional Operating Range (V_{CC} = 4.75V to 5.25V; $V_{CC}3$ = 3.135V to 3.465V; T_{CASE} = 0°C to +85°C) (Continued)

Symbol	Parameter	Min	Max	Fig	Notes
t12d	A[31:0] Hold Time from HCLKIN Rising	1.5		79	Hold from HCLKIN rising when EADS# is sampled active by the CPU.
t12e	A[31:0] Output Enable from HCLKIN Rising	0	13	81	
t12f	A[31:0] Valid Delay from HCLKIN Rising	1.3	13	78	0 pF
t12g	A[31:0] Float Delay from HCLKIN Rising	0	13	80	
t12h	A[2:0] Propagation Delay from BE[7:0]#	1.0	16	77	0 pF
t13a	BRDY# Rising Edge Valid Delay from HCLKIN Rising	1.6	7.5	78	0 pF
t13b	BRDY# Falling Edge Valid Delay from HCLKIN Rising	1.6	7.5	78	0 pF
t14	NA# Valid Delay from HCLKIN Rising	.9	7.6	78	0 pF
t15a	AHOLD Valid Delay from HCLKIN Rising	1.5	7.0	78	0 pF
t15b	BOFF# Valid Delay from HCLKIN Rising	1.5	7.0	78	0 pF
t16a	EADS#, INV, PEN# Valid Delay from HCLKIN Rising	1.5	7.5	78	0 pF
t16b	CPURST Rising Edge Valid Delay from HCLKIN Rising	1.2	7.0	78	0 pF
t16c	CPURST Falling Edge Valid Delay from HCLKIN Rising	1.2	7.0	78	0 pF
t16d	KEN# Valid delay from HCLKIN Rising	1.5	7.5	78	0 pF
t17	INIT High Pulse Width	2 HCLKs		84	0 pF
t18	CPURST High Pulse Width	1 ms		84	0 pF; Hard reset via TRC register

9.6.3 SECOND LEVEL CACHE STANDARD SRAM TIMING, 66 MHz (82434NX), PRELIMINARY

Functional Operating Range (V_{CC} = 4.75V to 5.25V; $V_{CC}3$ = 3.135V to 3.465V; T_{CASE} = 0°C to +85°C)

Symbol	Parameter	Min	Max	Fig	Notes
t20a	CAA[6:3]/CAB[6:3] Propagation Delay from A[6:3]	0	8.5	77	0 pF
t20b	CAA[6:3]/CAB[6:3] Valid Delay from HCLKIN Rising	0	7.2	78	0 pF
t21a	COE[1:0]# Falling Edge Valid Delay from HCLKIN Rising	0	9	78	0 pF
t21b	COE[1:0]# Rising Edge Valid Delay from HCLKIN Rising	0	5.5	78	0 pF
t22a	CWE[7:0]#/CBS[7:0]# Falling Edge Valid Delay from HCLKIN Rising	2	14	78	CPU burst or single write to second level cache, 0 pF
t22b	CWE[7:0]#/CBS[7:0]# Rising Edge Valid Delay from HCLKIN Rising	3	14	78	CPU burst or single write to second level cache, 0 pF
t22c	CWE[7:0]#/CBS[7:0]# Valid Delay from HCLKIN Rising	1.0	7.7	78	Cache line Fill, 0 pF
t22d	CWE[7:0]#/CBS[7:0]# Low Pulse Width	1 HCLK		84	0 pF
t22e	CWE[7:0]#/CBS[7:0]# Driven High before CALE Driven High	−1		85	Last write to second level cache during cache line fill, 0 pF
t22f	CAA[4:3]/CAB[4:3] Valid before CWE[7:0]# Falling	1.5		85	CPU burst write to second level cache, 0 pF
t23	CALE Valid Delay from HCLKIN Rising	0	8.0	78	0 pF
t24	CR/W[1:0]# Valid Delay from HCLKIN Rising	1.5	8.2	78	0 pF
t25	CBS[1:0]# Valid Delay from HCLKIN Rising; Reads from Cache SRAMs	1.0	12.0	78	0 pF
t26a	CCS[1:0]# Propagation Delay from ADS# Falling		7.0	77	0 pF; First access after powerdown
t26b	CCS[1:0]# Valid Delay from HCLKIN Rising	1.5	8.2	78	0 pF; Entering powerdown

9.6.4 SECOND LEVEL CACHE BURST SRAM TIMING, 66 MHz (82434NX), PRELIMINARY

Functional Operating Range (V_{CC} = 4.75V to 5.25V; $V_{CC}3$ = 3.135V to 3.465V; T_{CASE} = 0°C to +85°C)

Symbol	Parameter	Min	Max	Fig	Notes
t30a	CAA[6:3]/CAB[6:3] Propagation Delay from A[6:3]	0	8.5	77	0 pF
t30b	CAA[6:3]/CAB[6:3] Valid Delay from HCLKIN Rising	0	8.2	78	0 pF
t31	CADS[1:0]# Valid Delay from HCLKIN Rising	1.5	8.0	78	0 pF
t32	CADV[1:0]# Valid Delay from HCLKIN Rising	1.5	8.0	78	0 pF
t33	CWE[7:0]# Valid Delay from HCLKIN Rising	1.5	9.0	78	0 pF
t34a	COE[1:0]# Falling Edge Valid Delay from HCLKIN Rising	0.5	9.0	78	0 pF
t34b	COE[1:0]# Rising Edge Valid Delay from HCLKIN Rising	0.5	6.0	78	0 pF
t35	CALE Valid Delay from HCLKIN Rising	0	8.0	78	0 pF

2

9.6.5 DRAM INTERFACE TIMING, 66 MHz (82434NX), PRELIMINARY

Functional Operating Range (V_{CC} = 4.75V to 5.25V; $V_{CC}3$ = 3.135V to 3.465V; T_{CASE} = 0°C to +85°C)

Symbol	Parameter	Min	Max	Fig	Notes
t40a	RAS[7:0]# Valid Delay from HCLKIN Rising	0	8.0	78	50 pF
t40b	RAS[7:0]# Pulse Width High	4 HCLKs−5		84	RAS# precharge at beginning of page miss cycle, 50 pF
t41a	CAS[7:0]# Valid Delay from HCLKIN Rising	0	8.0	78	50 pF
t41b	CAS[7:0]# Pulse Width High	1 HCLKIN−5		84	CAS# precharge during burst cycles, 50 pF
t42	WE# Valid Delay from HCLKIN Rising	0	21	78	50 pF
t43a	MA[10:0] Propagation Delay from A[23:3]	0	23	77	50 pF
t43b	MA[10:0] Valid Delay from HCLKIN Rising	0	10.7	78	50 pF
t43c	MA11 Propagation Delay from A[25:24]	0	28.0	77	50 pF
t43d	MA11 Valid Delay from HCLKIN Rising	0	12	78	50 pF

9.6.6 PCI CLOCK TIMING, 66 MHz (82434NX), PRELIMINARY

Functional Operating Range (V_{CC} = 4.75V to 5.25V; $V_{CC}3$ = 3.135V to 3.465V; T_{CASE} = 0°C to +85°C)

Symbol	Parameter	Min	Max	Fig	Notes
t50a	PCLKOUT High Time	13		82	20 pF
t50b	PCLKOUT Low Time	13		82	20 pF
t51a	PCLKIN High Time	12		82	
t51b	PCLKIN Low Time	12		82	
t51c	PCLKIN Rise Time		3	83	
t51d	PCLKIN Fall Time		3	83	

9.6.7 PCI INTERFACE TIMING, 66 MHz (82434NX), PRELIMINARY

Functional Operating Range (V_{CC} = 4.75V to 5.25V; $V_{CC}3$ = 3.135V to 3.465V; T_{CASE} = 0°C to +85°C)

Symbol	Parameter	Min	Max	Fig	Notes
t60a	C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, PLOCK#, PAR, PERR#, SERR#, DEVSEL# Valid Delay from PCLKIN Rising	2	11	78	Min: 0 pF Max: 50 pF
t60b	C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, PLOCK#, PAR, PERR#, SERR#, DEVSEL# Output Enable Delay from PCLKIN Rising	2		81	
t60c	C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, PLOCK#, PAR, PERR#, SERR#, DEVSEL# Float Delay from PCLKIN Rising	2	28	80	
t60d	C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, PLOCK#, PAR, PERR#, SERR#, DEVSEL# Setup Time to PCLKIN Rising	7		79	
t60e	C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, PLOCK#, PAR, PERR#, SERR#, DEVSEL# Hold Time from PCLKIN Rising	0		79	
t61a	REQ#, MEMACK# Valid Delay from PCLKIN Rising	2	12	78	Min: 0 pF Max: 50 pF
t61b	REQ#, MEMACK# Output Enable Delay from PCLKIN Rising	2		81	
t61c	REQ#, MEMACK# Float Delay from PCLKIN Rising	2	28	80	
t62a	FLSHREQ#, MEMREQ# Setup Time to PCLKIN Rising	12		79	
t62b	FLSHREQ#, MEMREQ# Hold Time from PCLKIN Rising	0		79	

Functional Operating Range (V_{CC} = 4.75V to 5.25V; $V_{CC}3$ = 3.135V to 3.465V; T_{CASE} = 0°C to +85°C) (Continued)

Symbol	Parameter	Min	Max	Fig	Notes
t63a	GNT# Setup Time to PCLKIN Rising	10		79	
t63b	GNT# Hold Time from PCLKIN Rising	0		79	
t64a	MEMCS# Setup Time to PCLKIN Rising	7		79	
t64b	MEMCS# Hold Time from PCLKIN Rising	0		79	
t65	PCIRST# Low Pulse Width	1 ms		84	Hard Reset via TRC Register,0 pF

9.6.8 LBX INTERFACE TIMING, 66 MHz (82434NX), PRELIMINARY

Functional Operating Range (V_{CC} = 4.75V to 5.25V; $V_{CC}3$ = 3.135V to 3.465V; T_{CASE} = 0°C to +85°C)

Symbol	Parameter	Min	Max	Fig	Notes
t70	HIG[4:0] Valid Delay from HCLKIN Rising	0.8	6.5	78	0 pF
t71	MIG[2:0] Valid Delay from HCLKIN Rising	0.9	6.5	78	0 pF
t72	PIG[3:0] Valid Delay from PCLKIN Rising	1.5	12	78	0 pF
t73	PCIDRV Valid Delay from PCLKIN Rising	1	13	78	0 pF
t74a	MDLE Falling Edge Valid Delay from HCLKIN Rising	0.6	6.0	78	0 pF
t74b	MDLE Rising Edge Valid from HCLKIN Rising	0.6	6.0	85	0 pF
t75a	EOL, PPOUT[1:0] Setup Time to PCLKIN Rising	7.7		79	
t75b	EOL, PPOUT[1:0] Hold Time from PCLKIN Rising	1.0		79	

9.6.9 HOST CLOCK TIMING, 50 and 60 MHz (82434NX)

Functional Operating Range (V_{CC} = 4.75V to 5.25V; $V_{CC}3$ = 3.135V to 3.465V; T_{CASE} = 0°C to +85°C)

Symbol	Parameter	Min	Max	Fig	Notes
t1a	HCLKOSC High Time	6.0		82	
t1b	HCLKOSC Low Time	5.0		82	
t2a	HCLKIN Period	16.66	20	82	
t2b	HCLKIN Period Stability		±100		ps(1)
t2c	HCLKIN High Time	4		82	
t2d	HCLKIN Low Time	4		82	
t2e	HCLKIN Rise Time		1.5	83	
t2f	HCLKIN Fall Time		1.5	83	
t3a	HCLKA–HCLKF Output-to-Output Skew		0.5	85	0 pF
t3b	HCLKA–HCLKF High Time	5.0		82	0 pF
t3c	HCLKA–HCLKF Low Time	5.0		82	0 pF

NOTES:
1. Measured on rising edge of adjacent clocks at 1.5V.

9.6.10 CPU INTERFACE TIMING, 50 AND 60 MHz (82434NX)

Functional Operating Range (V_{CC} = 4.75V to 5.25V; $V_{CC}3$ = 3.135V to 3.465V; T_{CASE} = 0°C to +85°C)

Symbol	Parameter	Min	Max	Fig	Notes
t10a	ADS#, W/R#, Setup Time to HCLKIN Rising	4.6		79	
t10b	BE[7:0]# Setup Time to HCLKIN Rising	4.6		79	
t10c	HITM# Setup Time to HCLKIN Rising	6.8		79	
t10d	CACHE#, M/IO# Setup Time to HCLKIN Rising	4.6		79	
t10e	D/C# Setup Time to HCLKIN Rising	4.6		79	
t10f	HLOCK#, SMIACT#, Setup Time to HCLKIN Rising	4.6		79	
t10g	HITM#, M/IO#, D/C#, Hold Time from HCLKIN Rising	0.7		79	
t10h	W/R#, HLOCK# Hold from HCLKIN Rising	0.8		79	
t10i	ADS#, BE[7:0]# Hold Time from HCLKIN Rising	0.9		79	
t10j	CACHE#, SMIACT# Hold Time from HCLKIN Rising	1.1		79	
t11a	PCHK# Setup Time to HCLKIN Rising	4.3		79	
t11b	PCHK# Hold Time from HCLKIN Rising	1.1		79	
t12a	A[31:0] Setup Time to HCLKIN Rising	3.0		79	Setup to HCLKIN rising when ADS# is sampled active by PCMC.
t12b	A[31:0] Hold Time from HCLKIN Rising	0.5		79	HOLD from HCLKIN Rising two clocks after ADS# is sampled active by PCMC
t12c	A[31:0] Setup Time to HCLKIN Rising	6.5		79	Setup to HCLKIN rising when EADS# is sampled active by the CPU.
t12d	A[31:0] Hold Time from HCLKIN Rising	1.5		79	Hold from HCLKIN rising when EADS# is sampled active by the CPU.

Functional Operating Range (V_{CC} = 4.75V to 5.25V; $V_{CC}3$ = 3.135V to 3.465V; T_{CASE} = 0°C to +85°C)
(Continued)

Symbol	Parameter	Min	Max	Fig	Notes
t12e	A[31:0] Output Enable from HCLKIN Rising	0	13	81	
t12f	A[31:0] Valid Delay from HCLKIN Rising	1.3	13	78	0 pF
t12g	A[31:0] Float Delay from HCLKIN Rising	0	13	80	
t12h	A[2:0] Propagation Delay from BE[7:0]#	1.0	16	77	0 pF
t13a	BRDY# Rising Edge Valid Delay from HCLKIN Rising	2.1	7.9	78	0 pF
t13b	BRDY# Falling Edge Valid Delay from HCLKIN Rising	2.1	7.9	78	0 pF
t14	NA# Valid Delay from HCLKIN Rising	1.4	8.4	78	0 pF
t15a	AHOLD Valid Delay from HCLKIN Rising	2.0	7.6	78	0 pF
t15b	BOFF# Valid Delay from HCLKIN Rising	2.0	7.6	78	0 pF
t16a	EADS#, INV, PEN# Valid Delay from HCLKIN Rising	2.0	8.0	78	0 pF
t16b	CPURST Rising Edge Valid Delay from HCLKIN Rising	1.2	7.5	78	0 pF
t16c	CPURST Falling Edge Valid Delay from HCLKIN Rising	1.2	7.5	78	0 pF
t16d	KEN# Valid delay from HCLKIN Rising	1.7	8.2	78	0 pF
t17	INIT High Pulse Width	2 HCLKs		84	0 pF
t18	CPURST High Pulse Width	1 ms		84	0 pF; Hard reset via TRC register

9.6.11 SECOND LEVEL CACHE STANDARD SRAM TIMING, 50 AND 60 MHz (82434NX)

Functional Operating Range (V_{CC} = 4.75V to 5.25V; $V_{CC}3$ = 3.135V to 3.465V; T_{CASE} = 0°C to +85°C)

Symbol	Parameter	Min	Max	Fig	Notes
t20a	CAA[6:3]/CAB[6:3] Propagation Delay from A[6:3]	0	8.5	77	0 pF
t20b	CAA[6:3]/CAB[6:3] Valid Delay from HCLKIN Rising	0	7.2	78	0 pF
t21a	COE[1:0]# Falling Edge Valid Delay from HCLKIN Rising	0	9.0	78	0 pF
t21b	COE[1:0]# Rising Edge Valid Delay from HCLKIN Rising	0	5.5	78	0 pF
t22a	CWE[7:0]#/CBS[7:0]# Falling Edge Valid Delay from HCLKIN Rising	2	14	78	CPU burst or single write to second level cache, 0 pF
t22b	CWE[7:0]#/CBS[7:0]# Rising Edge Valid Delay from HCLKIN Rising	3	15	78	CPU burst or single write to second level cache, 0 pF
t22c	CWE[7:0]#/CBS[7:0]# Valid Delay from HCLKIN Rising	1.4	7.7	78	Cache line Fill, 0 pF
t22d	CWE[7:0]#/CBS[7:0]# Low Pulse Width	14		84	0 pF
t22e	CWE[7:0]#/CBS[7:0]# Driven High before CALE Driven High	-1		85	Last write to second level cache during cache line fill, 0 pF
t22f	CAA[4:3]/CAB[4:3] Valid before CWE[7:0]# Falling	1.5		85	CPU burst write to second level cache, 0 pF
t23	CALE Valid Delay from HCLKIN Rising	0	8	78	0 pF
t24	CR/W[1:0]# Valid Delay from HCLKIN Rising	1.5	8.2	78	0 pF
t25	CBS[1:0]# Valid Delay from HCLKIN Rising; Reads from Cache SRAMs	1.0	12.0	78	0 pF
t26a	CCS[1:0]# Propagation Delay from ADS# Falling		7.0	77	0 pF; First access after powerdown
t26b	CCS[1:0]# Valid Delay from HCLKIN Rising	1.5	8.2	78	0 pF; Entering powerdown

9.6.12 SECOND LEVEL CACHE BURST SRAM TIMING, 50 AND 60 MHz (82434NX)

Functional Operating Range (V_{CC} = 4.75V to 5.25V; $V_{CC}3$ = 3.135V to 3.465V; T_{CASE} = 0°C to +85°C)

Symbol	Parameter	Min	Max	Fig	Notes
t30a	CAA[6:3]/CAB[6:3] Propagation Delay from A[6:3]	0	8.5	77	0 pF
t30b	CAA[6:3]/CAB[6:3] Valid Delay from HCLKIN Rising	0	8.2	78	0 pF
t31	CADS[1:0]# Valid Delay from HCLKIN Rising	1.5	8.2	78	0 pF
t32	CADV[1:0]# Valid Delay from HCLKIN Rising	1.5	8.2	78	0 pF
t33	CWE[7:0]# Valid Delay from HCLKIN Rising	1.0	10.5	78	0 pF
t34a	COE[1:0]# Falling Edge Valid Delay from HCLKIN Rising	0	9.5	78	0 pF
t34b	COE[1:0]# Rising Edge Valid Delay from HCLKIN Rising	0	6.0	78	0 pF
t35	CALE Valid Delay from HCLKIN Rising	0	8.5	78	0 pF

2

9.6.13 DRAM INTERFACE TIMING, 50 AND 60 MHz (82434NX)

Functional Operating Range (V_{CC} = 4.75V to 5.25V; $V_{CC}3$ = 3.135V to 3.465V; T_{CASE} = 0°C to +85°C)

Symbol	Parameter	Min	Max	Fig	Notes
t40a	RAS[7:0]# Valid Delay from HCLKIN Rising	0	8.0	78	50 pF
t40b	RAS[7:0]# Pulse Width High	4 HCLKs–5		84	RAS# precharge at beginning of page miss cycle, 50 pF
t41a	CAS[7:0]# Valid Delay from HCLKIN Rising	0	8.0	78	50 pF
t41b	CAS[7:0]# Pulse Width High	1 HCLK–5		84	CAS# precharge during burst cycles, 50 pF
t42	WE# Valid Delay from HCLKIN Rising	0	21	78	50 pF
t43a	MA[10:0] Propagation Delay from A[23:9]	0	23	77	50 pF
t43b	MA[10:0] Valid Delay from HCLKIN Rising	0	10.7	78	50 pF
t43c	MA11 Propagation Delay from A[25:24]	0	24.3	77	50 pF
t43d	MA11 Valid Delay from HCLKIN Rising	0	12	78	50 pF

9.6.14 PCI CLOCK TIMING, 50 AND 60 MHz (82434NX)

Functional Operating Range (V_{CC} = 4.75V to 5.25V; $V_{CC}3$ = 3.135V to 3.465V; T_{CASE} = 0°C to +85°C)

Symbol	Parameter	Min	Max	Fig	Notes
t50a	PCLKOUT High Time	13		82	20 pF
t50b	PCLKOUT Low Time	13		82	20 pF
t51a	PCLKIN High Time	12		82	
t51b	PCLKIN Low Time	12		82	
t51c	PCLKIN Rise Time		3	83	
t51d	PCLKIN Fall Time		3	83	

9.6.15 PCI INTERFACE TIMING, 50 AND 60 MHz (82434NX)

Functional Operating Range (V_{CC} = 4.75V to 5.25V; $V_{CC}3$ = 3.135V to 3.465V; T_{CASE} = 0°C to +85°C)

Symbol	Parameter	Min	Max	Fig	Notes
t60a	C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, PLOCK#, PAR, PERR#, SERR#, DEVSEL# Valid Delay from PCLKIN Rising	2	11	78	Min: 0 pF Max: 50 pF
t60b	C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, PLOCK#, PAR, PERR#, SERR#, DEVSEL# Output Enable Delay from PCLKIN Rising	2		81	
t60c	C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, PLOCK#, PAR, PERR#, SERR#, DEVSEL# Float Delay from PCLKIN Rising	2	28	80	
t60d	C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, PLOCK#, PAR, PERR#, SERR#, DEVSEL# Setup Time to PCLKIN Rising	9		79	
t60e	C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, PLOCK#, PAR, PERR#, SERR#, DEVSEL# Hold Time from PCLKIN Rising	0		79	
t61a	REQ#, MEMACK# Valid Delay from PCLKIN Rising	2	12	78	Min: 0 pF Max: 50 pF
t61b	REQ#, MEMACK# Output Enable Delay from PCLKIN Rising	2		81	
t61c	REQ#, MEMACK# Float Delay from PCLKIN Rising	2	28	80	
t62a	FLSHREQ#, MEMREQ# Setup Time to PCLKIN Rising	12		79	
t62b	FLSHREQ#, MEMREQ# Hold Time from PCLKIN Rising	0		79	
t63a	GNT# Setup Time to PCLKIN Rising	10		79	
t63b	GNT# Hold Time from PCLKIN Rising	0		79	
t64a	MEMCS# Setup Time to PCLKIN Rising	7		79	
t64b	MEMCS# Hold Time from PCLKIN Rising	0		79	
t65	PCIRST# Low Pulse Width	1 ms		84	Hard Reset via TRC Register, 0 pF

9.6.16 LBX INTERFACE TIMING, 50 AND 60 MHz (82434NX)

Functional Operating Range (V_{CC} = 4.75V to 5.25V; $V_{CC}3$ = 3.135V to 3.465V; T_{CASE} = 0°C to +85°C)

Symbol	Parameter	Min	Max	Fig	Notes
t70	HIG[4:0] Valid Delay from HCLKIN Rising	0.8	6.7	78	0 pF
t71	MIG[2:0] Valid Delay from HCLKIN Rising	0.9	6.5	78	0 pF
t72	PIG[3:0] Valid Delay from PCLKIN Rising	1.5	12	78	0 pF
t73	PCIDRV Valid Delay from PCLKIN Rising	1	13	78	0 pF
t74a	MDLE Falling Edge Valid Delay from HCLKIN Rising	0.6	6.8	78	0 pF
t74b	MDLE Rising Edge Valid Delay from HCLKIN Rising	0.6	6.8	85	0 pF
t75a	EOL, PPOUT[1:0] Setup Time to PCLKIN Rising	7.7		79	
t75b	EOL, PPOUT[1:0] Hold Time from PCLKIN Rising	1.0		79	

2

9.6.17 TIMING DIAGRAMS

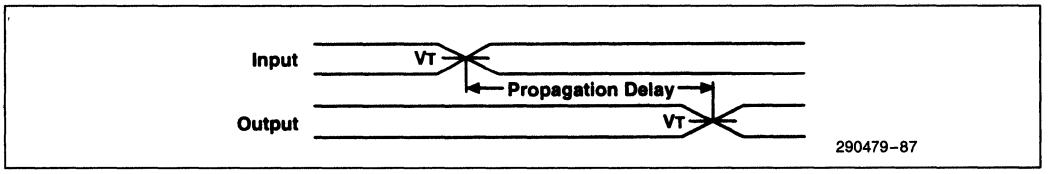


Figure 77. Propagation Delay

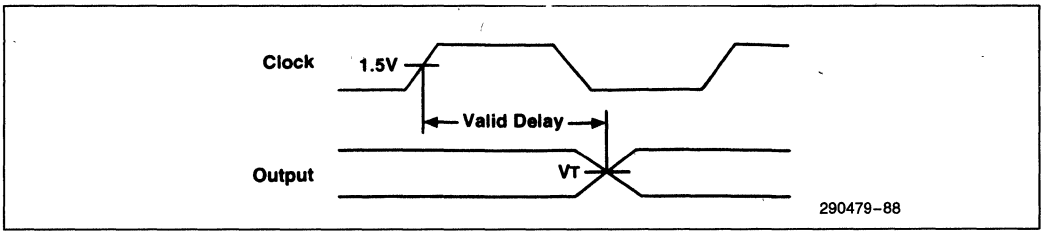


Figure 78. Valid Delay from Rising Clock Edge

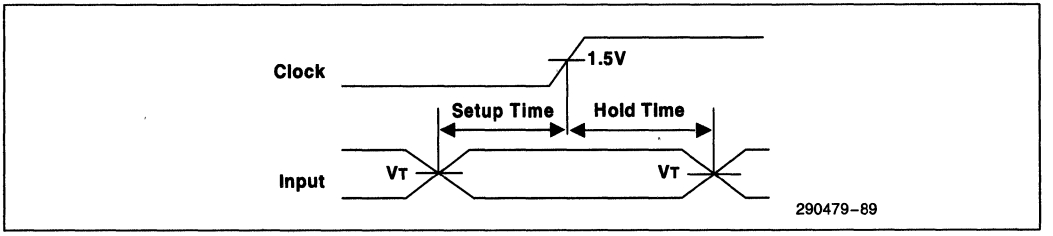


Figure 79. Setup and Hold Times

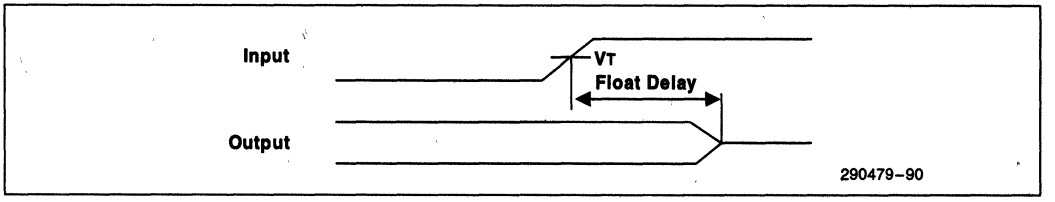


Figure 80. Float Delay

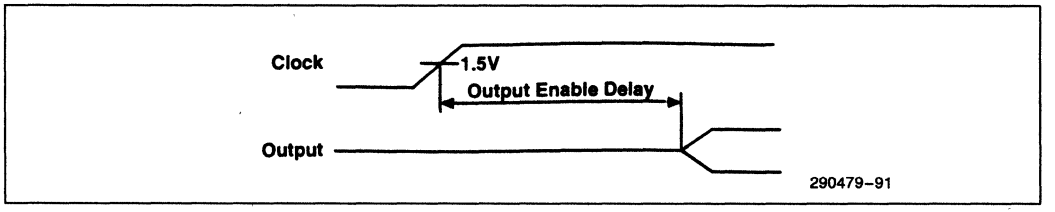


Figure 81. Output Enable Delay

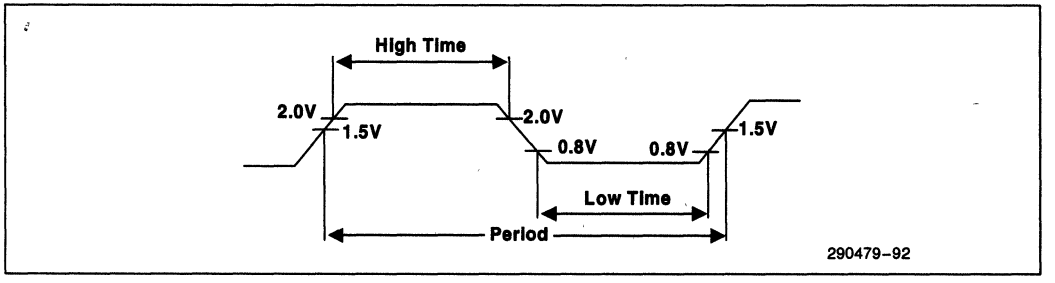


Figure 82. Clock High and Low Times and Period

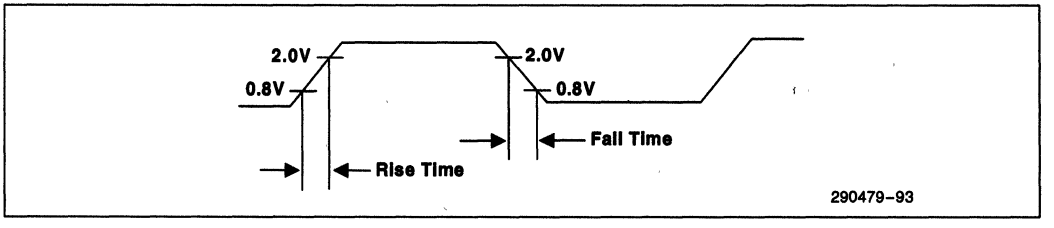


Figure 83. Clock Rise and Fall Times

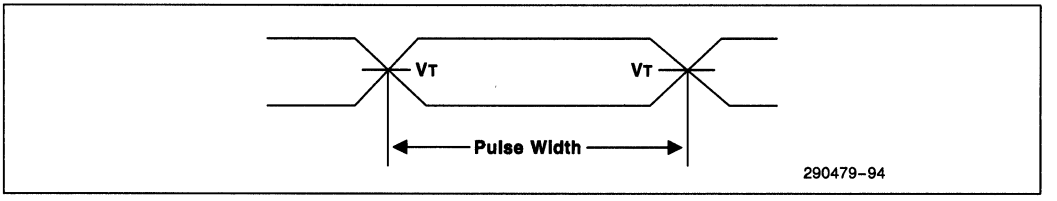


Figure 84. Pulse Width

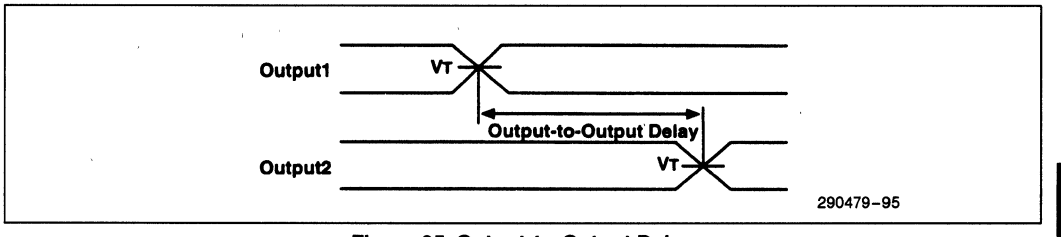
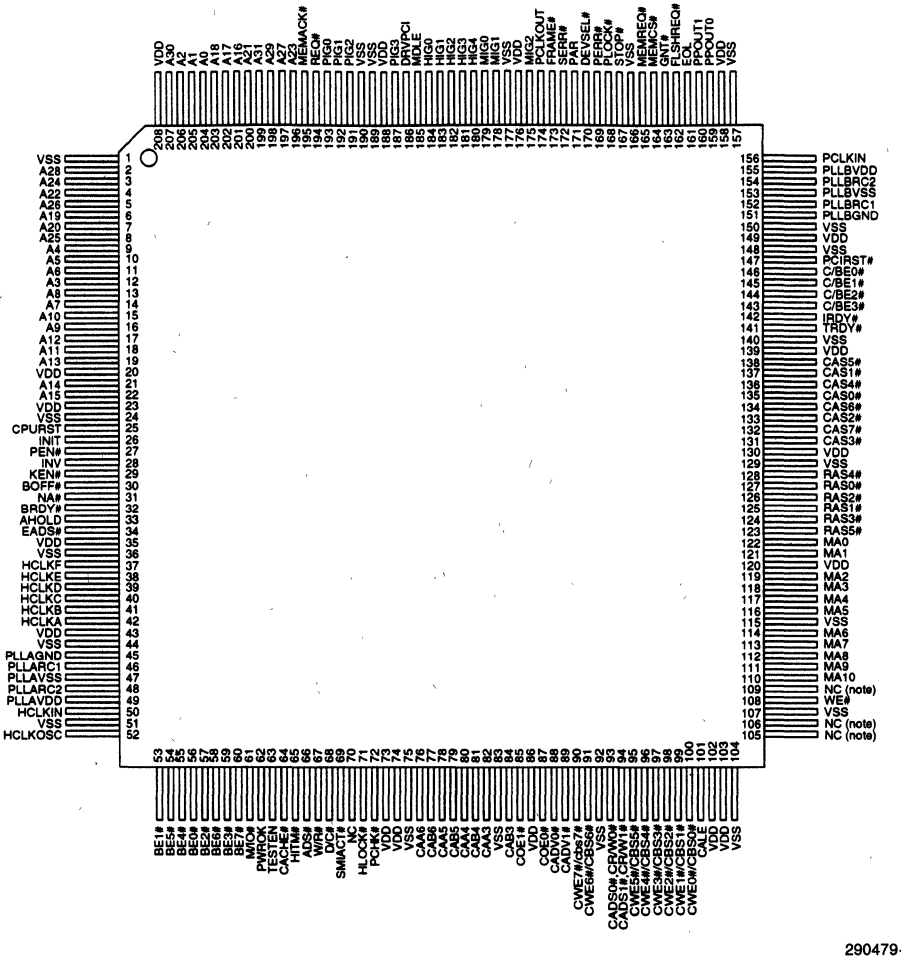


Figure 85. Output-to-Output Delay

10.0 PINOUT AND PACKAGE INFORMATION

10.1 Pin Assignment

Except for the pins listed in Figure 86 notes, the pin assignment for the 82434LX and 82434NX are the same.



290479-96

NOTES:

1. For the 82434NX, pin 105 = RAS6#, 106 = RAS7#, and 109 = MA11. These pins are no connects for the 82434LX and are signal connections for the 82434NX.
2. For the 82434NX, pins 23, 35, 43, 74, 86, and 102 are 3.3V VDD pins (i.e., VDD3). These pins are VDD pins for the 82434LX.

Figure 86. PCMC Pin Assignment

Table 21. 82434LX Alphabetical Pin Assignment

Pin Name	Pin #	Type
A0	204	t/s
A1	205	t/s
A2	206	t/s
A3	12	t/s
A4	9	t/s
A5	10	t/s
A6	11	t/s
A7	14	t/s
A8	13	t/s
A9	16	t/s
A10	15	t/s
A11	18	t/s
A12	17	t/s
A13	19	t/s
A14	21	t/s
A15	22	t/s
A16	201	t/s
A17	202	t/s
A18	203	t/s
A19	6	t/s
A20	7	t/s
A21	200	t/s
A22	4	t/s
A23	196	t/s
A24	3	t/s
A25	8	t/s
A26	5	t/s
A27	197	t/s
A28	2	t/s
A29	198	t/s
A30	207	t/s
A31	199	t/s
ADS#	66	in

Pin Name	Pin #	Type
AHOLD	33	out
BE0#	56	in
BE1#	53	in
BE2#	57	in
BE3#	59	in
BE4#	55	in
BE5#	54	in
BE6#	58	in
BE7#	60	in
BOFF#	30	out
BRDY#	32	out
CAA3	82	out
CAA4	80	out
CAA5	78	out
CAA6	76	out
CAB3	84	out
CAB4	81	out
CAB5	79	out
CAB6	77	out
CACHE#	64	in
CADS0#,CR/W0#	93	out
CADS1#,CR/W1#	94	out
CADV0# (82434LX) CADV0#/CCS0# (82434NX)	88	out
CADV1# (82434LX) CADV1#/CCS1# (82434NX)	89	out
CALE	101	out
CAS0#	135	out
CAS1#	137	out
CAS2#	133	out
CAS3#	131	out
CAS4#	136	out

Pin Name	Pin #	Type
CAS5#	138	out
CAS6#	134	out
CAS7#	132	out
CBE0#	146	t/s
CBE1#	145	t/s
CBE2#	144	t/s
CBE3#	143	t/s
COE0#	87	out
COE1#	85	out
CPURST	25	out
CWE0#/CBS0#	100	out
CWE1#/CBS1#	99	out
CWE2#/CBS2#	98	out
CWE3#/CBS3#	97	out
CWE4#/CBS4#	96	out
CWE5#/CBS5#	95	out
CWE6#/CBS6#	91	out
CWE7#/CBS7#	90	out
D/C#	68	in
DEVSEL#	170	s/t/s
DRVPCI	186	out
EADS#	34	out
EOL	161	in
FLSHREQ#	162	in
FRAME#	173	s/t/s
GNT#	163	in
HCLKA	42	out
HCLKB	41	out
HCLKC	40	out
HCLKD	39	out
HCLKE	38	out
HCLKF	37	out
HCLKIN	50	in

Table 21. 82434LX Alphabetical Pin Assignment (Continued)

Pin Name	Pin #	Type
HCLKOSC	52	in
HIG0	184	out
HIG1	183	out
HIG2	182	out
HIG3	181	out
HIG4	180	out
HITM#	65	in
HLOCK#	71	in
INIT	26	out
INV	28	out
IRDY#	142	s/t/s
KEN#	29	out
M/IO#	61	in
MA0	122	out
MA1	121	out
MA2	119	out
MA3	118	out
MA4	117	out
MA5	116	out
MA6	114	out
MA7	113	out
MA8	112	out
MA9	111	out
MA10	110	out

Pin Name	Pin #	Type
MA11 (82434NX only)	109	out
MDLE	185	out
MEMACK#	195	out
MEMCS#	164	in
MEMREQ#	165	in
MIG0	179	out
MIG1	178	out
MIG2	175	out
NA#	31	out
NC	70	NC
NC (82434LX only)	105	NC
NC (82434LX only)	106	NC
NC (82434LX only)	109	NC
PAR	171	t/s
PCHK#	72	in
PCIRST#	147	out
PCLKIN	156	in
PCLKOUT	174	out
PEN#	27	out
PERR#	169	s/o/d
PIG0	193	out
PIG1	192	out
PIG2	191	out
PIG3	187	out

Pin Name	Pin #	Type
PLLAGND	45	V
PLLARC1	46	in
PLLARC2	48	in
PLLAVDD	49	V
PLLAVSS	47	V
PLLBGND	151	V
PLLBRC1	152	in
PLLBRC2	154	in
PLLBVDD	155	V
PLLBVSS	153	V
PLOCK#	168	s/t/s
PPOUT0	159	in
PPOUT1	160	in
PWROK	62	in
RAS0#	127	out
RAS1#	125	out
RAS2#	126	out
RAS3#	124	out
RAS4#	128	out
RAS5#	123	out
RAS6# (82434NX only)	105	out
RAS7# (82434NX only)	106	out

Table 21. 82434LX Alphabetical Pin Assignment (Continued)

Pin Name	Pin #	Type
REQ#	194	out
SERR#	172	s/o/d
SMIACT#	69	in
STOP#	167	s/t/s
TESTEN	63	in
TRDY#	141	s/t/s
V_{DD}	20	V
V_{DD} (82434LX) V_{DD3} (82434NX)	23	V
V_{DD} (82434LX) V_{DD3} (82434NX)	35	V
V_{DD} (82434LX) V_{DD3} (82434NX)	43	V
V_{DD}	73	V
V_{DD} (82434LX) V_{DD3} (82434NX)	74	V
V_{DD} (82434LX) V_{DD3} (82434NX)	86	V
V_{DD} (82434LX) V_{DD3} (82434NX)	102	V

Pin Name	Pin #	Type
V_{DD}	103	V
V_{DD}	120	V
V_{DD}	130	V
V_{DD}	139	V
V_{DD}	149	V
V_{DD}	158	V
V_{DD}	176	V
V_{DD}	188	V
V_{DD}	208	V
V_{SS}	1	V
V_{SS}	24	V
V_{SS}	36	V
V_{SS}	44	V
V_{SS}	51	V
V_{SS}	75	V
V_{SS}	83	V

Pin Name	Pin #	Type
V_{SS}	92	V
V_{SS}	104	V
V_{SS}	107	V
V_{SS}	115	V
V_{SS}	129	V
V_{SS}	140	V
V_{SS}	148	V
V_{SS}	150	V
V_{SS}	157	V
V_{SS}	166	V
V_{SS}	177	V
V_{SS}	189	V
V_{SS}	190	V
W/R#	67	in
WE#	108	out

Table 22. Numerical Pin Assignment

Pin #	Pin Name	Type
1	V_{SS}	V
2	A28	t/s
3	A24	t/s
4	A22	t/s
5	A26	t/s
6	A19	t/s
7	A20	t/s
8	A25	t/s
9	A4	t/s
10	A5	t/s
11	A6	t/s
12	A3	t/s
13	A8	t/s
14	A7	t/s
15	A10	t/s
16	A9	t/s
17	A12	t/s
18	A11	t/s
19	A13	t/s
20	V_{DD}	V
21	A14	t/s
22	A15	t/s
23	V_{DD} (82434LX) V_{DD3} (82434NX)	V
24	V_{SS}	V
25	CPURST	out
26	INIT	out
27	PEN#	out
28	INV	out
29	KEN#	out
30	BOFF#	out
31	NA#	out

Pin #	Pin Name	Type
32	BRDY#	out
33	AHOLD	out
34	EADS#	out
35	V_{DD} (82434LX) V_{DD3} (82434NX)	V
36	V_{SS}	V
37	HCLKF	out
38	HCLKE	out
39	HCLKD	out
40	HCLKC	out
41	HCLKB	out
42	HCLKA	out
43	V_{DD} (82434LX) V_{DD3} (82434NX)	V
44	V_{SS}	V
45	PLLAGND	V
46	PLLARC1	in
47	PLLAVSS	V
48	PLLARC2	in
49	PLLAVDD	V
50	HCLKIN	in
51	V_{SS}	V
52	HCLKOSC	in
53	BE1#	in
54	BE5#	in
55	BE4#	in
56	BE0#	in
57	BE2#	in
58	BE6#	in
59	BE3#	in
60	BE7#	in
61	M/IO#	in

Pin #	Pin Name	Type
62	PWROK	in
63	TESTEN	in
64	CACHE#	in
65	HITM#	in
66	ADS#	in
67	W/R#	in
68	D/C#	in
69	SMIACT#	in
70	NC	NC
71	HLOCK#	in
72	PCHK#	in
73	V_{DD}	V
74	V_{DD} (82434LX) V_{DD3} (82434NX)	V
75	V_{SS}	V
76	CAA6	out
77	CAB6	out
78	CAA5	out
79	CAB5	out
80	CAA4	out
81	CAB4	out
82	CAA3	out
83	V_{SS}	V
84	CAB3	out
85	COE1#	out
86	V_{DD} (82434LX) V_{DD3} (82434NX)	V
87	COE0#	out
88	CADV0# (82434LX) CADV0#/CCS0# (82434NX)	out
89	CADV1# (82434LX) CADV1#/CCS1# (82434NX)	out

Table 22. Numerical Pin Assignment (Continued)

Pin #	Pin Name	Type
90	CWE7#/CBS7#	out
91	CWE6#/CBS6#	out
92	V_{SS}	V
93	CADS0#,CR/W0#	out
94	CADS1#,CR/W1#	out
95	CWE5#/CBS5#	out
96	CWE4#/CBS4#	out
97	CWE3#/CBS3#	out
98	CWE2#/CBS2#	out
99	CWE1#/CBS1#	out
100	CWE0#/CBS0#	out
101	CALE	out
102	V_{DD} (82434LX) V_{DD3} (82434NX)	V
103	V_{DD}	V
104	V_{SS}	V
105	NC (82434LX) RAS6# (82434NX)	NC out
106	NC (82434LX) RAS7# (82434NX)	NC out
107	V_{SS}	V
108	WE#	out
109	NC (82434LX) MA11 (82434NX)	NC out
110	MA10	out
111	MA9	out
112	MA8	out
113	MA7	out
114	MA6	out
115	V_{SS}	V
116	MA5	out
117	MA4	out
118	MA3	out

Pin #	Pin Name	Type
119	MA2	out
120	V_{DD}	V
121	MA1	out
122	MA0	out
123	RAS5#	out
124	RAS3#	out
125	RAS1#	out
126	RAS2#	out
127	RAS0#	out
128	RAS4#	out
129	V_{SS}	V
130	V_{DD}	V
131	CAS3#	out
132	CAS7#	out
133	CAS2#	out
134	CAS6#	out
135	CAS0#	out
136	CAS4#	out
137	CAS1#	out
138	CAS5#	out
139	V_{DD}	V
140	V_{SS}	V
141	TRDY#	s/t/s
142	IRDY#	s/t/s
143	CBE3#	t/s
144	CBE2#	t/s
145	CBE1#	t/s
146	CBE0#	t/s
147	PCIRST#	out
148	V_{SS}	V
149	V_{DD}	V
150	V_{SS}	V

Pin #	Pin Name	Type
151	PLLBGND	V
152	PLLBRC1	in
153	PLLBVSS	V
154	PLLBRC2	in
155	PLLBVDD	V
156	PCLKIN	in
157	V_{SS}	V
158	V_{DD}	V
159	PPOUT0	in
160	PPOUT1	in
161	EOL	in
162	FLSHREQ#	in
163	GNT#	in
164	MEMCS#	in
165	MEMREQ#	in
166	V_{SS}	V
167	STOP#	s/t/s
168	PLOCK#	s/t/s
169	PERR#	s/o/d
170	DEVSEL#	s/t/s
171	PAR	t/s
172	SERR#	s/o/d
173	FRAME#	s/t/s
174	PCLKOUT	out
175	MIG2	out
176	V_{DD}	V
177	V_{SS}	V
178	MIG1	out
179	MIG0	out
180	HIG4	out
181	HIG3	out
182	HIG2	out

Table 22. Numerical Pin Assignment (Continued)

Pin #	Pin Name	Type
183	HIG1	out
184	HIG0	out
185	MDLE	out
186	DRVPCI	out
187	PIG3	out
188	V_{DD}	V
189	V_{SS}	V
190	V_{SS}	V
191	PIG2	out

Pin #	Pin Name	Type
192	PIG1	out
193	PIG0	out
194	REQ#	out
195	MEMACK#	out
196	A23	t/s
197	A27	t/s
198	A29	t/s
199	A31	t/s
200	A21	t/s

Pin #	Pin Name	Type
201	A16	t/s
202	A17	t/s
203	A18	t/s
204	A0	t/s
205	A1	t/s
206	A2	t/s
207	A30	t/s
208	V_{DD}	V

10.2 Package Characteristics

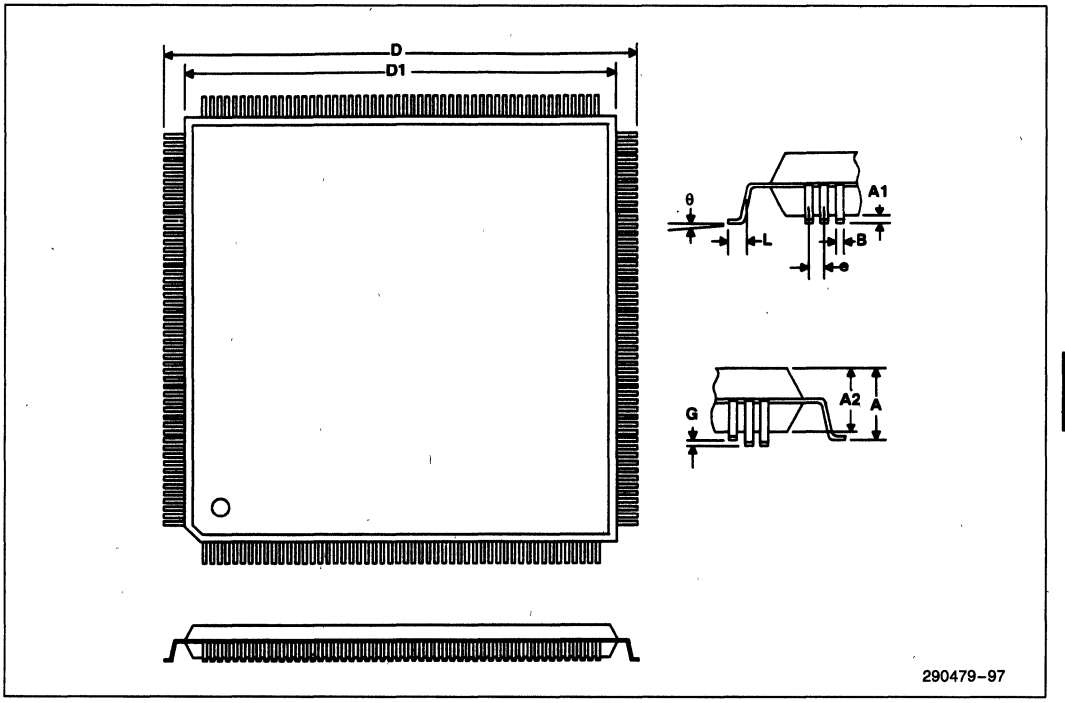


Figure 87. 208-Pin Quad Flatpack (QFP) Dimensions

Table 23. 82434LX Package Dimensions

Symbol	Description	Value (mm)
A	Seating Height	3.5 (max)
A1	Stand-Off Height	0.20–0.50
A2	Package Height	3.0 (nominal)
B	Lead Width	0.18 +0.1/−0.05
D	Package Length and Width, Including Pins	30.6 ± 0.3
D1	Package Length and Width, Excluding Pins	28 ± 0.1
e	Linear Lead Pitch	0.5 ± 0.1
G	Lead Coplanarity	0.1 (max)
L	Lead Length	0.5 ± 0.2
θ	Lead Angle	0°–10°

Table 24. 82434NX Package Dimensions

Symbol	Description	Value (mm)
A	Seating Height	3.7 (max)
A1	Stand-Off Height	0.05–0.50
A2	Package Height	3.45 (max)
B	Lead Width	0.13–0.27
D	Package Length and Width, Including Pins	30.6 ± 0.3
D1	Package Length and Width, Excluding Pins	28 ± 0.1
e	Linear Lead Pitch	0.5 (nominal)
G	Lead Coplanarity	0.1 (max)
L	Lead Length	0.5 ± 0.2
θ	Lead Angle	0°–10°

11.0 TESTABILITY

A NAND tree is provided in the 82434LX and 82434NX PCMCs for Automated Test Equipment (ATE) board level testing. The NAND tree allows the tester to test the connectivity of a subset of the PCMC signal pins.

For the 82434LX, the output of the NAND tree is driven on pin 109. The NAND tree is enabled when A24 = 1, A25 = 0, A26 = 1, and TESTEN = 1 at the rising edge of PWROK. PLL Bypass mode is enabled when A24 = 1, and TESTEN = 1 at the rising edge of PWROK. In PLL Bypass mode, the 82434LX and 82434NX PCMC AC specifications are affected as follows:

1. Output valid delays increase by 20 ns.
2. All hold times are 20 ns.
3. Setup times and propagation delays are unaffected.
4. Input clock high and low times are 100 ns.

In both the NAND tree test mode and PLL Bypass mode, TESTEN must remain asserted throughout the testing. A[28:24] should be set up at least 1 HCLK before the rising edge of PWROK and held at least 3 HCLKs after PWROK. Table 11 shows the order of the NAND tree inside the PCMC.

When not in NAND Tree test mode, the 82434LX drives the output of the host clock PLL onto pin 109.

82434NX Test Modes

The state of A[28:24], TESTEN, CPURST, and PWROK can place the 82434NX PCMC into two test modes. When PWROK is low, A[27:24] and TESTEN directly control the mode of operation of the PCMC. When PWROK is high, the state of A[27:24] and TESTEN are latched and the PCMC remains in the indicated mode until PWROK is again negated. The high order LBX samples the state of A27 on the falling edge of CPURST.

When PWROK is low and both TESTEN and A27 are low, the 82434NX drives MA11 onto pin 109. If both TESTEN and A27 are low when PWROK transitions from low to high, the PCMC continues to drive MA11 onto pin 109. If the high order LBX samples A27 low on the falling edge of CPURST, it will tri-state pin 123.

When PWROK is low, TESTEN is low, and A27 is high the PCMC drives the output of the host clock PLL onto pin 109. Observing pin 109 when in this mode indicates if the host clock PLL has locked onto the correct frequency. If TESTEN is low and A27 is high when PWROK transitions from low to high the PCMC continues to drive the output of the host clock PLL onto pin 109, regardless of the values of TESTEN and A27. If the high order LBX samples A27 high on the falling edge of CPURST, it drives the output of its host clock PLL onto pin 123. No phase delay information can be inferred from these outputs.

When PWROK is low, TESTEN is high, A26 is high, A25 is low, A28 is high and A24 is high, the PCMC will drive the output of the NAND tree onto pin 109. If TESTEN is high, A26 is high, and A25 is low when PWROK transitions from low to high, the PCMC continues to drive the output of the NAND tree onto pin 109.

A27 must be pulled low via a pulldown resistor to ground for normal operation.

Table 25. NAND Tree Order

Order	Pin #	Signal	Order	Pin #	Signal	Order	Pin #	Signal
1	141	TRDY#	25	199	A31	49	17	A12
2	142	IRDY#	26	200	A21	50	18	A11
3	143	CBE3#	27	201	A16	51	19	A13
4	144	CBE2#	28	202	A17	52	21	A14
5	145	CBE1#	29	203	A18	53	22	A15
6	146	CBE0#	30	204	A0	54	53	BE1#
7	159	PPOUT0	31	205	A1	55	54	BE5#
8	160	PPOUT1	32	206	A2	56	55	BE4#
9	161	EOL	33	207	A30	57	56	BE0#
10	162	FLSHBUF#	34	2	A28	58	57	BE2#
11	163	GNT#	35	3	A24	59	58	BE6#
12	164	MEMCS#	36	4	A22	60	59	BE3#
13	165	MEMREQ#	37	5	A26	61	60	BE7#
14	167	STOP#	38	6	A19	62	61	M/IO#
15	168	PLOCK#	39	7	A20	63	64	CACHE#
16	169	PERR#	40	8	A25	64	65	HITM#
17	170	DEVSEL#	41	9	A4	65	66	ADS#
18	171	PAR	42	10	A5	66	67	W/R#
19	172	SERR#	43	11	A6	67	68	D/C#
20	173	FRAME#	44	12	A3	68	69	SMIACT#
21	194	REQ#	45	13	A8	69	71	HLOCK#
22	196	A23	46	14	A7	70	72	PCHK#
23	197	A27	47	15	A10	71	63	TESTEN
24	198	A29	48	16	A9			

ADDITIONAL TESTING NOTES:
1. HCLKOUT[6:1] can be toggled via HCLKIN.
2. CAx[6:3] are flow through outputs via A[6:3] after PWROK transitions high.
3. MA[10:0] are flow through outputs via A[13:3] after PWROK transitions high.
4. CAS[7:0]# outputs can be tested by performing a DRAM read cycle.
5. PCLKOUT can be tested in PLL bypass mode, frequency is HCLK/2.
6. PCIRST is the NAND Tree output of Tree Cell 6.
7. INIT is the NAND Tree output of Tree Cell 53.