1 AMD-K5™ Processor Features

- Four-issue superscalar core with six parallel execution units arranged in a five-stage pipeline
- 16-Kbyte, dual-tagged, four-way, set-associative instruction cache
- 8-Kbyte, dual-tagged, dual-ported with four banks, four-way set-associative, writeback data cache
- Full, out-of-order speculative execution and completion
- Dynamic cache line-oriented branch prediction with 1-Kbyte branch predictions and low 3-cycle branch mispredict penalty
- Integrated, high-performance floating-point unit (FPU) with low-latency add/multiply and single-cycle issue
- Static clock control with Phase Lock Loop (PLL) circuitry
- 3.3-V operation and System Management Mode (SMM) for lower power consumption
- 64-bit Pentium-compatible bus and system interface in a 296-pin SPGA package
- Compatible with existing Pentium (P54C) support infrastructure and system designs
- Fully compatible with the Microsoft® Windows® operating systems and the large installed library of x86 software

1.1 Redefining the Next Generation

AMD continues to bring superior, high-performance processor solutions to the personal computer market. The AMD-K5 processor offers superior price/performance value over other 5th-generation processors—making it an ideal solution for mainstream desktop computers. Compatible with the entire installed library of x86 software, the AMD-K5 processor is a superior engine for the Microsoft Windows operating systems.

The AMD-K5 processor uses an independently developed "superscalar RISC-based design" manufactured in AMD's 0.35-micron complementary metal-oxide semiconductor (CMOS) process. The design stems from a rich history of experience in RISC and x86 technology, providing a solid foundation for the development of our proprietary 4.3-million-transistor AMD-K5 processor.

1.2 High-Performance Design

The superscalar RISC design techniques provide next-generation performance levels and the power to run complex 32-bit operating systems and applications. The AMD-K5 processor features a four-issue superscalar core that incorporates dynamic branch prediction and out-of-order speculative execution. While other 5th-generation processors feature a two-issue core, the AMD-K5 processor's RISC core is four-issue.

1.3 Compatibility

The AMD-K5 processor's compatibility is established using a rigorous testing procedure that begins with software simulation before the design is first committed to silicon. Throughout the design and manufacturing process, industry-standard tools and systems are used for compatibility testing.

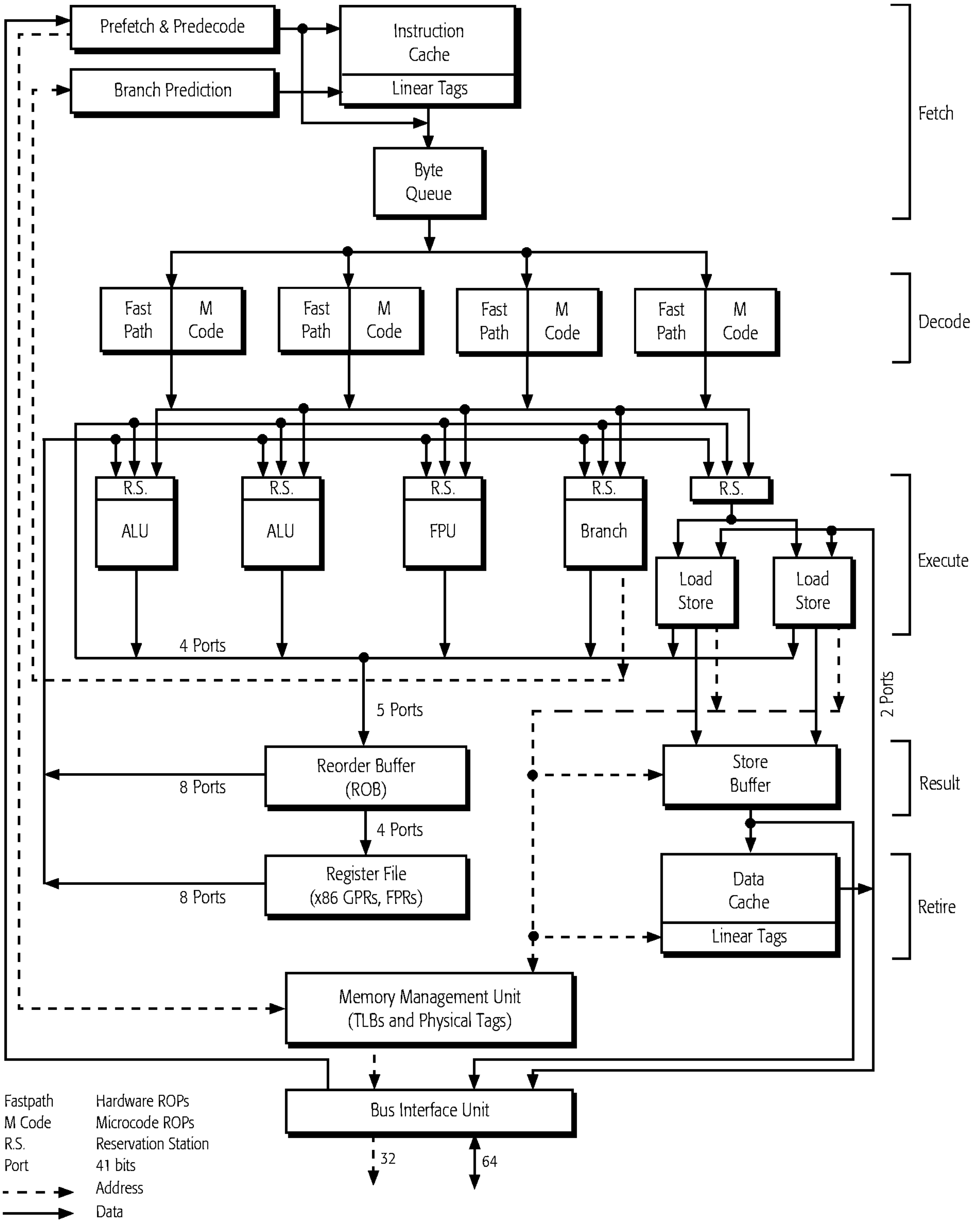
Extended compatibility and qualification testing are provided by industry-leading personal computer and chip set manufacturers. Testing culminates with certification from XXCAL, Inc., an independent third-party testing lab. This combination of differentiating features is responsible for the AMD-K5 processor's overall design and performance advantages.

Compatibility with the Microsoft Windows operating system and the immense library of x86 software furthers these advantages, and is the foundation of the AMD-K5 processor's leading-edge solution.

Revision History

Date	Revision	Description
Jan. 1997	F	The PR166 OPN added to Ordering Information in Section 3 on page 5.
		The valid combinations are updated in the Ordering Information in Section 3 on page 5.
		Model 2 added to the CPU Identification in Section 5 on page 12.
		P-Rating information added to the CPU Identification in Section 5 on page 12.
		Manufacturer in JTAG ID code changed to bits 11-1 in CPU Identification in Section 5 on page 12.
		1.75 multiplier added to the BF1-BF0 pin description in Signal Descriptions in Section 7 on page 14.
		New data cache write allocate information added beginning on page 34
		Pipelining information added beginning on page 47.
		V_{CC} changes in Operating Ranges on page 57.
		I_{CC} updated in Table 14 on page 58.
		The package thermal specifications on page 82 are updated for new models and I_{CC} specs.
		All references to model 1 are changed to models 1 and 2.

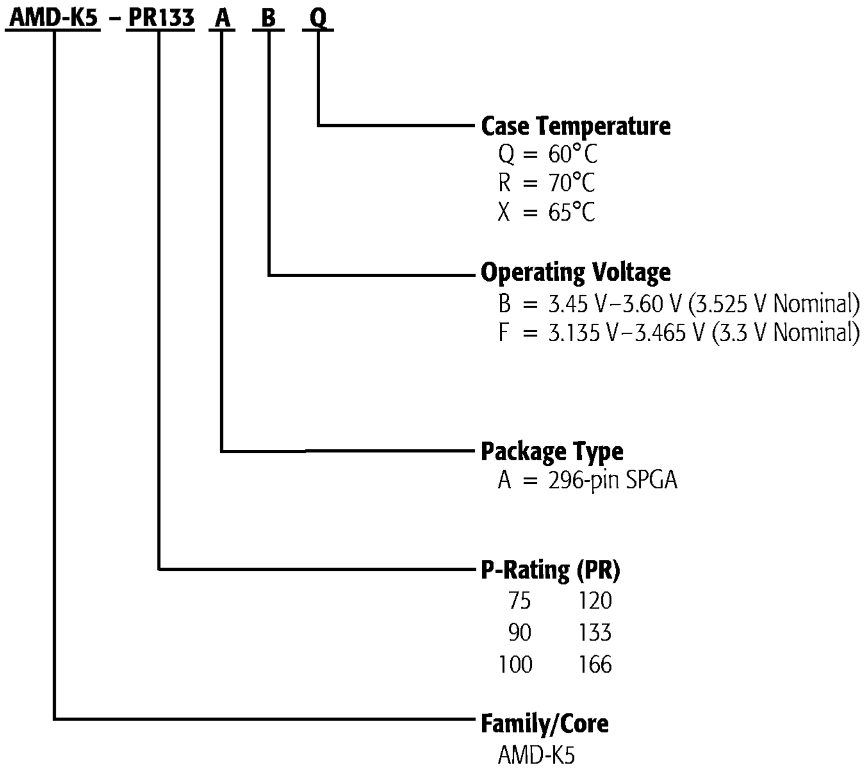
2 Block Diagram



3 Ordering Information

Standard Products

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



AMD-K5 - PR133 A B Q

Case Temperature
Q = 60°C
R = 70°C
X = 65°C

Operating Voltage
B = 3.45 V–3.60 V (3.525 V Nominal)
F = 3.135 V–3.465 V (3.3 V Nominal)

Package Type
A = 296-pin SPGA

P-Rating (PR)

75	120
90	133
100	166

Family/Core
AMD-K5

Valid Combinations

OPN	Package Type	Operating Voltage	Case Temperature
AMD-K5-PR166ABX	296-pin SPGA	3.45 V–3.60 V	65°C
AMD-K5-PR133ABR	296-pin SPGA	3.45 V–3.60 V	70°C
AMD-K5-PR133ABQ	296-pin SPGA	3.45 V–3.60 V	60°C
AMD-K5-PR120ABR	296-pin SPGA	3.45 V–3.60 V	70°C
AMD-K5-PR100ABQ	296-pin SPGA	3.45 V–3.60 V	60°C
AMD-K5-PR90ABQ	296-pin SPGA	3.45 V–3.60 V	60°C
AMD-K5-PR75ABR	296-pin SPGA	3.45 V–3.60 V	70°C

Notes:

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

4 Architectural Introduction

The x86 architecture is the dominant standard for the personal computer marketplace. However, maintaining backwards compatibility with previous generations of x86 processors carries several inherent limitations associated with the x86 architecture: variable-length instruction set, fewer general-purpose registers, and complex addressing modes. The AMD-K5 processor overcomes these burdens by providing superscalar architecture that incorporates innovative technology: instruction predecoding, improved cache architecture, branch prediction with speculative execution, a superscalar RISC core, out-of-order execution, and register renaming.

4.1 Superscalar RISC Core

The AMD-K5 processor's superscalar RISC core consists of six execution units: two arithmetic logic units (ALU), two load/ store units, one branch unit, and one floating-point unit (FPU). This superscalar core is fully decoupled from the x86 bus through the conversion of variable-length x86 instructions into simple, fixed-length RISC operations (ROPs) that are easier to handle and execute faster. Once the x86 instruction has been converted, a dispatcher issues four ROPs at a time to the superscalar core. The processor's superscalar core can execute at a peak rate of six ROPs per cycle. The superscalar core supports data forwarding and data bypassing to immediately forward the results of an execution to successive instructions. This eliminates the delay of writing the results to output registers or memory and reading them back to the instruction needing the results.

4.2 Out-of-Order Execution

The AMD-K5 processor implements out-of-order execution to eliminate delays due to pipeline dependencies. Each execution unit has two reservation stations that hold ROPs prior to execution (except the FPU, which has one reservation station). ROPs can be issued out of order from the reservation stations and executed out of order. Some execution units will empty their reservation stations before others. Since each execution unit can operate independently, other units can continue execution when one or more units are stalled. A 16-entry reorder buffer keeps track of the original instruction sequence and ensures that the results are retired in program order.

4.3 Register Renaming

The x86 architecture has only eight general-purpose registers. This significantly increases register reuse (loads and stores) and register dependencies. The register reuse is addressed with multiple load/store execution units and a dual-ported data cache. The AMD-K5 processor uses register renaming to overcome register dependencies. Multiple logical registers for each physical register allow execution units to use the same physical name registers simultaneously.

4.4 64-Bit Data Bus Interface Unit

The AMD-K5 processor uses a 64-bit data bus that provides higher throughput and support for 64-bit data paths, and a cache/burst-oriented line refill for loading the processor's internal separate instruction and data caches. As code and data enter the bus interface unit, the internal cache refills continually as fast as five clock cycles per cache line. The enhanced bandwidth of the processor's data bus and the continuous cache refill process reduces processing delays and supports superior processor and overall system performance.

4.5 Innovative x86 Instruction Predecoding

While processing variable-length instructions is manageable in single-issue 4th-generation and dual-issue 5th-generation CPUs, only the AMD-K5 processor employs the necessary innovative techniques to issue as many as four x86 instructions per clock cycle.

Every byte of code that enters the AMD-K5 processor is tagged with associated predecode information that identifies the x86 instruction boundaries and enables multiple x86 instructions (varying in length from 8 to 120 bits) to be aligned. Once aligned, the instructions are assigned issue positions for the most efficient instruction processing—contributing to the processor's high performance. In addition to indicating where the x86 instruction begins and ends, the predecode information identifies the position of the opcode and the number of simple RISC-like operations (ROPs) the individual x86 instruction requires for later translation.

After the x86 instructions are predecoded, they are loaded into the instruction cache. When accessed from the instruction cache, the *speculative* instructions (x86 instructions from a predicted branch stream) are pushed into the byte queue and await further decoding. The byte queue not only contains the x86 instructions but also the associated predecode tags that mark each instruction's position and operation type.

4.6 Cache Architecture

Much of the AMD-K5 processor's performance advantage can be credited to the processor's instruction cache architecture and its ability to feed the processor core. Using separate instruction and data caches eliminates the internal conflicts over simultaneous instruction cache access and x86 loads and stores. The processor's 16-Kbyte instruction cache is dual-tagged, avoiding the linear-to-physical address translation required to access every entry and allowing faster cache access. In addition, the processor maintains a separate set of physical instruction tags for snooping and aliasing, and through a special protocol, prevents flushing the cache even during Translation Lookaside Buffer (TLB) flushes or context switches.

The processor's instruction cache implements a four-way set-associative structure for maximum cache performance in a given size and maintains branch prediction information with every cache line.

The 8-Kbyte data cache allows two cache lines of data to be accessed simultaneously in a single clock cycle, as long as separate banks within the data cache are accessed. Supporting two accesses per clock enables the data cache to overcome the load/store bottlenecks inherent in the x86 architecture.

The AMD-K5 processor's data cache uses a modified, exclusive, shared, invalid (MESI) protocol to maintain data coherency with other caches in the system and to ensure that a read from a given memory location returns valid data. Each cache line is assigned one of the four protocol states to identify the status of the information stored in the cache. The writeback cache design updates memory only when necessary. This keeps the system bus free for use by other devices and improves the overall system performance.

4.7 Branch Prediction

A branch occurs on average once every seven x86 instructions. When a branch is encountered, the processor predicts which direction the instruction flow will follow. The AMD-K5 processor adds branch prediction information to each instruction cache line in the form of a predicted address tag that indicates the target address of the first branch that is predicted to be taken in the cache line. The processor's dynamic branch prediction mechanism allows for 1024 branch targets and a 75% branch prediction accuracy. Combined with a minimal 3-cycle mispredict penalty, the branch prediction mechanism optimizes the processor's speculative execution of x86 software, such as the Microsoft Windows operating system and associated applications.

The dynamic branch prediction of the processor enables instructions to be fetched and fed into the processor's execution core, eliminating many pipeline bubbles and contributing to the superior performance of the AMD-K5 processor.

4.8 Unique x86 Instruction Conversion and Decoding

The logical instruction flow within the AMD-K5 processor continues as up to 32-bytes of predecoded x86 instructions are fetched from the byte queue of the instruction cache and forwarded in order to the decoder.

The processor's decoder converts complex x86 instructions into relatively simple, fast-executing ROPs that are of fixed length and easy to process. Simultaneously, the operands needed to perform the ROPs' operations are fetched from the register file or from the reorder buffer.

At the beginning of the decode process, the decoder scans the x86 instructions and allocates the instructions to the appropriate decode position. This allocation depends on the 5-bit tag given to each x86 instruction during predecode. When the predecoded instruction passes through the AMD-K5 processor's decoder, the number of ROPs needed to equate to the x86 instruction is already known from predecoding, saving valuable processing time.

During allocation, the instruction's pathways are identified. If an x86 instruction requires less than four ROPs for conversion, it is sent immediately to any of the four decode positions (Fastpath). Complex x86 instructions requiring four or more ROPs (or ROP sequences) are transferred to the Microcode ROM (MROM) for conversion.

Once through the decode position, the ROPs are dispatched in parallel to reservation stations that reside in each of the processor's six execution units. A reservation station precedes the input to individual execution units. Each execution unit has a pair of reservation stations.

The processor sends ROPs to the reservation stations in order, but when the ROPs are passed on to the execution units they can be executed out of order because the reservation stations can empty at different times. Out-of-order execution eliminates the need for compiler-specific optimization and reduces dependencies. The ROPs wait in the reservation stations for the execution unit processing to complete and for the needed operands, which come from the register file, the data cache, or are forwarded from other execution units. As an execution unit finishes processing one instruction, it receives another instruction from the reservation station. Using reservation stations in this manner, the processor minimizes instruction stalls due to dependencies on execution resources and allows a higher issue rate to be maintained.

4.9 Reorder Buffer

The AMD-K5 processor uses a central reorder buffer—a key to supporting speculative out-of-order execution (issue and completion). The central reorder buffer is used to rename registers, provide subsequent forwarding of requested intermediate results, recover from mispredictions and exceptions, and hold the relative speculative state.

The processor's 16-entry reorder buffer stores results from x86 instructions that have been speculatively executed at the time a branch was predicted. When ROPs are dispatched to one of the processor's six independent execution units, an entry at the top of the reorder buffer is allocated for each ROP. Up to four entries are allocated simultaneously. The reorder buffer keeps track of the original instruction sequence and ensures that results are retired in program order, writing the results of the executed instruction to the register file. If a branch is mispredicted, the results of the instructions along the mispredicted path are invalidated in the reorder buffer before there is any effect on the x86 registers or memory system.

4.10 Register File

A problem with the x86 architecture has been its limited number of general-purpose registers. Fewer registers means frequent reuse of registers, which potentially leads to a reduction in performance. The AMD-K5 CPU utilizes register renaming and avoids this performance reduction.

Because the movement of values between registers and memory locations is unavoidable with the x86 instruction set, a key advantage of the AMD-K5 CPU is its single-cycle load from the data cache. This, in combination with the multiported register file and renaming in the reorder buffer, gives near optimal speculative performance within the constraints of the x86 instruction set.

4.11 The Right Combination—Compatibility and Performance

While each feature has a significant function, it is the combination of all features that is responsible for the AMD-K5 processor's overall design and performance advantages. Compatibility with the Microsoft Windows operating system and the immense library of x86 software furthers these advantages, and is the foundation of the AMD-K5 processor's leading-edge solution.

5 CPU Identification

Upon completion of RESET, the DX register contains a component identification. The upper byte of DX (DH) will contain 05h. The lower byte of DX (DL) will contain a CPU model (0h–2h)/stepping identifier (xh).

CPU ID

Family ID (DH)	Model ID (DL, top 4 bits)	CPU Frequency (MHz)	Bus Speed	Processor P-Rating	BF Pin	BF1–BF0 Pins
5	0	75	50	AMD-K5-PR75	1	N/A
		90	60	AMD-K5-PR90	1	N/A
		100	66	AMD-K5-PR100	1	N/A
	1	90	60	AMD-K5-PR120	N/A	10
		100	66	AMD-K5-PR133	N/A	10
	2	116.7	66	AMD-K5-PR166	N/A	00

Notes:

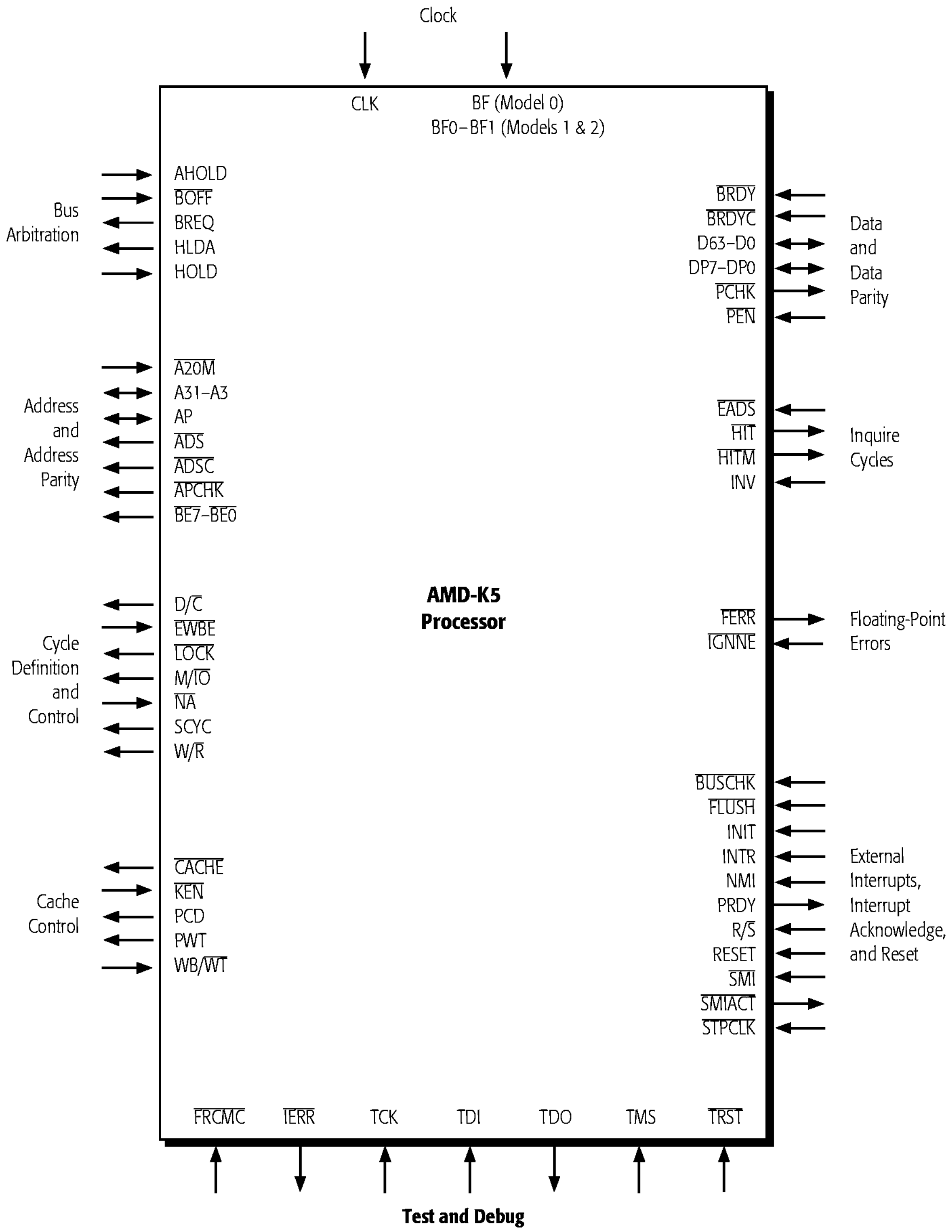
This table does not constitute product announcements. Instead, the information in the table represents possible product offerings. AMD will announce actual products based on availability and market demand

The boundary scan test access port (TAP) returns the following information in the device identification register (DIR).

JTAG ID Code

Version (Bits 31–28)	Bond Option (Bit 27)	Unused (Bits 26–24)	Part Number (Bits 23–12)	Manufacturer (Bits 11–1)	LSB (Bit 0)
xh	xb	000b	50xh (Model 0) 51xh (Model 1) 52xh (Model 2)	00000000001b	1b

6 Logic Symbol Diagram



7 Signal Descriptions

A31–A5/A4–A3

Address Lines ***Input/Output***

A31–A3 are used with $\overline{\text{BE7}}$–$\overline{\text{BE0}}$ to form the address bus. These signals are outputs to address memory space, I/O space, and system management memory. A31–A5 are used as inputs for inquire cycles. A4–A3 are not used during the inquire cycle, but must be driven to valid levels. During bus hold, address hold, or back-off, A31–A3 are floated. (See Switching Characteristics t_{14} and t_{15}.)

$\overline{\text{A20M}}$

Address Bit 20 Mask ***Input***

Asserting $\overline{\text{A20M}}$ will mask address bit 20 internally for internal cache accesses or driving memory cycles on the external bus. $\overline{\text{A20M}}$ should be asserted only in Real mode. Its effect is not defined in Protected mode. The state of $\overline{\text{A20M}}$ is ignored during transfers to and from SMM memory. $\overline{\text{A20M}}$ is sampled on every rising clock edge. (See Switching Characteristics t_{26} and t_{27}.)

$\overline{\text{ADS}}$

Address Status ***Output***

$\overline{\text{ADS}}$ indicates the beginning of a new bus cycle. Valid addresses and cycle information are available on the address bus simultaneously with the assertion of $\overline{\text{ADS}}$. $\overline{\text{ADS}}$ is floated during bus hold or back-off.

$\overline{\text{ADSC}}$

Address Status Copy ***Output***

$\overline{\text{ADSC}}$ performs the same function as $\overline{\text{ADS}}$. It permits greater fanout. $\overline{\text{ADSC}}$ is normally used to directly drive the cache to achieve greater speed.

AHOLD

Address Hold ***Input***

A31–A3 and AP are floated on the clock after AHOLD is recognized as asserted. Other signals remain active. This allows another bus master to access the processor's address bus for a cache inquire cycle. AHOLD has a small internal pulldown resistor. (See Switching Characteristics t_{22} and t_{23}.)

AP — ***Address Parity*** — ***Input/Output***

The AP signal provides even parity for the address bus. This signal is driven simultaneously with the address bus. Inquire cycles that do not provide even parity in the same clock cycle as $\overline{\mathrm{EADS}}$ will result in the assertion of $\overline{\mathrm{APCHK}}$. (See $\overline{\mathrm{APCHK}}$.)

$\overline{\mathbf{APCHK}}$ — ***Address Parity Check*** — ***Output***

If the processor detects an address parity error on the address bus for inquire cycles, $\overline{\mathrm{APCHK}}$ is asserted on the second clock after $\overline{\mathrm{EADS}}$ is sampled . It remains active for one clock.

$\overline{\mathbf{BE7}}$–$\overline{\mathbf{BE0}}$ — ***Byte Enables*** — ***Output***

The $\overline{\mathrm{BE7}}$–$\overline{\mathrm{BE0}}$ signals indicate active bytes during read and write cycles. The eight byte-enable signals correspond to the eight bytes of the data bus as follows:

- $\overline{\mathrm{BE7}}$: D63–D56
- $\overline{\mathrm{BE6}}$: D55–D48
- $\overline{\mathrm{BE5}}$: D47–D40
- $\overline{\mathrm{BE4}}$: D39–D32
- $\overline{\mathrm{BE3}}$: D31–D24
- $\overline{\mathrm{BE2}}$: D23–D16
- $\overline{\mathrm{BE1}}$: D15–D8
- $\overline{\mathrm{BE0}}$: D7–D0

These signals are driven at the same time as the address bus. The byte-enable signals are also used to decode special cycles as defined in Table 6.

BF (Model 0) — ***Bus Frequency*** — ***Input***

For the AMD-K5 Model 0 processor, the BF signal determines the internal operating speed of the processor. The frequency of the CLK signal is multiplied internally by a ratio determined by the state of the BF signal during RESET. If BF is sampled High at RESET, the clock frequency is 1.5x the bus frequency. If BF is sampled Low at RESET, the clock frequency is 2x the bus frequency.

BF Pin	Internal Clock Multiplier
0	2
1	1.5

BF1–BF0 (Model 1 and Model 2)

Bus Frequency ***Input***

For the AMD-K5 model 1 and model 2 processors, the BF1 and BF0 signals determine the internal operating speed of the processor. The frequency of the CLK signal is multiplied internally by a ratio determined by the states of the BF1 and BF0 signals during RESET. The processor speed multiplier is determined as shown below:

BF1 Pin	BF0 Pin	Internal Clock Multiplier
0	0	1.75
0	1	Reserved
1	0	1.5
1	1	1.5

$\overline{\text{BOFF}}$

Backoff ***Input***

The processor will transition to a bus hold state and float the associated signals on the clock that $\overline{\text{BOFF}}$ is sampled as asserted. An alternate master may drive the bus signals on the clock after $\overline{\text{BOFF}}$ is sampled asserted. When $\overline{\text{BOFF}}$ is negated, the processor will restart any bus cycle from the beginning. Burst cycles interrupted by $\overline{\text{BOFF}}$ will restart from the beginning of the burst cycle. $\overline{\text{BOFF}}$ takes priority over $\overline{\text{BRDY}}$. If $\overline{\text{BRDY}}$ is sampled asserted in the same cycle as $\overline{\text{BOFF}}$, the cycle will be restarted. (See Switching Characteristics t_{22} and t_{23}.)

$\overline{\text{BRDY}}$

Burst Ready ***Input***

$\overline{\text{BRDY}}$ is sampled on the second and following clocks of a bus cycle to indicate completion of a data transfer cycle. $\overline{\text{BRDY}}$ is ignored at the end of the first clock of a bus cycle and when the bus is in an idle state. The data bus is sampled when $\overline{\text{BRDY}}$ is asserted. Up to four assertions of $\overline{\text{BRDY}}$ are needed to complete the bus cycle. (See Switching Characteristics t_{20} and t_{21}.)

$\overline{\text{BRDYC}}$

Burst Ready Copy ***Input***

$\overline{\text{BRDYC}}$ is functionally identical to $\overline{\text{BRDY}}$. These signals are connected internally by an OR gate. $\overline{\text{BRDYC}}$ is typically used by level two cache. At the falling edge of RESET, the states of $\overline{\text{BRDYC}}$ and $\overline{\text{BUSCHK}}$ control the drive strength on the A21–A3 (not including A31–A22), $\overline{\text{ADS}}$, $\overline{\text{HITM}}$, and W/$\overline{\text{R}}$ signals. The drive strength is weak for all states of $\overline{\text{BRDYC}}$ and $\overline{\text{BUSCHK}}$ except when $\overline{\text{BRDYC}}$ and $\overline{\text{BUSCHK}}$ are both Low, in which case the drive strength is strong. The A31–A22 signals use the weak drive strength at all times.

BREQ ***Bus Request Pending*** ***Output***

The processor asserts the BREQ signal to indicate a request for the bus. This signal is driven even when the processor floats the bus (except in Test mode). (See $\overline{\text{FLUSH}}$.)

$\overline{\text{BUSCHK}}$ ***Bus Check*** ***Input***

The $\overline{\text{BUSCHK}}$ signal allows the external system to indicate bus cycle errors. This signal, when asserted, latchs the address bus. The control signals in the machine check registers will also latch. If the MCE bit in CR4 is set, the processor will vector to the machine check exception at the end of the bus cycle. At the falling edge of RESET, the states of $\overline{\text{BRDYC}}$ and $\overline{\text{BUSCHK}}$ control the drive strength on the A21–A3 (not including A31–A22), $\overline{\text{ADS}}$, $\overline{\text{HITM}}$, and W/$\overline{\text{R}}$ signals. The drive strength is weak for all states of $\overline{\text{BRDYC}}$ and $\overline{\text{BUSCHK}}$ except $\overline{\text{BRDYC}}$ and $\overline{\text{BUSCHK}}$ both Low, in which case drive strength is strong. A31–A22 use the weak drive strength at all times.

$\overline{\text{CACHE}}$ ***Cache Status*** ***Output***

The $\overline{\text{CACHE}}$ signal is asserted for cacheable read cycles or burst writeback cycles. A burst access is always four 64-bit transfers associated with a line refill or a cache write back. Read data will not be cached if $\overline{\text{CACHE}}$ is negated during a read cycle, or if $\overline{\text{KEN}}$ is negated. $\overline{\text{KEN}}$ must be asserted during the first access of a burst transfer. If $\overline{\text{KEN}}$ is negated, a single access occurs.

CLK ***Clock*** ***Input***

The CLK signal is the bus clock for the processor, and is the primary reference for all bus cycle timings (except for test signals). It is used with the BF signal to determine the internal operating speed of the processor. The processor multiplies the clock input by 1.5 or 2. (See BF.)

D/$\overline{\text{C}}$ ***Data/Code*** ***Output***

The D/$\overline{\text{C}}$ signal, driven active with $\overline{\text{ADS}}$, is used with other control signals to determine bus cycle and special cycle types. It is floated with $\overline{\text{BOFF}}$ and bus hold. These cycles are defined in Table 5 and Table 6 on page 27.

D63–D0

Data Lines ***Input/Output***

The D63–D0 signals are the 64-bit data bus. These signals are driven during the second and subsequent clocks of write cycles, with valid bytes indicated by $\overline{\mathrm{BE7}}$–$\overline{\mathrm{BE0}}$. They are sampled when the $\overline{\mathrm{BRDY}}$ signal is asserted for read cycles. (See Switching Characteristics t_{34} and t_{35}.)

DP7–DP0

Data Parity ***Input/Output***

The DP7–DP0 signals provide even parity, one for each of the eight bytes of the data bus. The eight data parity signals correspond to the eight bytes of the data bus as follows:

- DP7: D63–D56
- DP6: D55–D48
- DP5: D47–D40
- DP4: D39–D32
- DP3: D31–D24
- DP2: D23–D16
- DP1: D15–D8
- DP0: D7–D0

These signals are driven with the data bus. Read cycles that do not provide even parity when the read data is driven result in the assertion of $\overline{\mathrm{PCHK}}$. Byte enables are negated for invalid data bytes. For systems that do not use parity, DP7–DP0 should be connected to V_{CC} through a pull-up resistor. (See $\overline{\mathrm{PCHK}}$ and Switching Characteristics t_{34} and t_{35}.)

$\overline{\mathrm{EADS}}$

Valid External Address ***Input***

The $\overline{\mathrm{EADS}}$ signal indicates that a valid address is driven on the address bus during inquire cycles. $\overline{\mathrm{EADS}}$ has an internal pull-up resistor. (See Switching Characteristics t_{16a} and t_{17}.)

EWBE

External Write Buffer Empty ***Input***

External system logic notifies the processor of pending buffered write cycles by negating the $\overline{\mathrm{EWBE}}$ signal. The processor will hold writes to exclusive or modified cache lines until $\overline{\mathrm{EWBE}}$ is asserted.

$\overline{\mathrm{FERR}}$

Floating-Point Error ***Output***

The $\overline{\mathrm{FERR}}$ signal is asserted as a result of an unmasked floating-point error. It is only floated during test.

$\overline{\text{FLUSH}}$

Cache Flush ***Input***

Asserting $\overline{\text{FLUSH}}$ will flush the internal caches. For acceptance, $\overline{\text{FLUSH}}$ must meet the required setup and hold times for one or more clocks. Instruction and data caches will be invalidated. Any modified data in the data cache will be written back. A flush acknowledge cycle will follow the invalidation to notify external logic that the internal caches have been flushed. The $\overline{\text{FLUSH}}$ signal is also sampled at the falling edge of RESET. If sampled Low, the processor will operate in Tri-State Test mode.

$\overline{\text{FRCMC}}$

Functional Redundancy Check
Master/Checker ***Input***

$\overline{\text{FRCMC}}$ is used to configure the processor as a Master or Checker. $\overline{\text{FRCMC}}$ is only sampled at RESET. Sampling $\overline{\text{FRCMC}}$ High configures the AMD-K5 processor for Master mode operation, and sampling $\overline{\text{FRCMC}}$ Low configures the processor for Checker operation. The processor follows standard bus protocol in Master mode. It floats all outputs, with the exception of $\overline{\text{IERR}}$ and TDO, in Checker mode. In Checker mode, all signals are inputs and their values are compared with predicted values.

$\overline{\text{HIT}}$

Hit ***Output***

The $\overline{\text{HIT}}$ signal is asserted when an inquire cycle hits a valid line in the instruction or data cache. This signal can be sampled two clock cycles after $\overline{\text{EADS}}$ has been sampled as asserted.

$\overline{\text{HITM}}$

Hit to a Modified Line ***Output***

The $\overline{\text{HITM}}$ signal is asserted when an inquire cycle hits a modified line in the data cache. This signal can be sampled two clock cycles after $\overline{\text{EADS}}$ has been sampled as asserted. $\overline{\text{HITM}}$ will remain asserted until the modified line has been written back.

HLDA

Hold Acknowledge ***Output***

The HLDA signal is driven to acknowledge a bus hold request. The bus is floated when HLDA is asserted. HLDA will be negated one clock cycle after HOLD is negated. (See HOLD.)

HOLD

Bus Hold Request ***Input***

The HOLD signal is used to request the processor bus. When this signal is asserted, the processor will complete all pending bus cycles, float the bus, and assert the HLDA signal. This signal is not recognized during locked cycles. (See Switching Characteristics t_{24} and t_{25b}.)

$\overline{\text{IERR}}$

Internal Error ***Output***

$\overline{\text{IERR}}$ indicates internal parity errors and functional redundancy errors. Internal parity errors will cause $\overline{\text{IERR}}$ to be asserted for one clock, and the processor will halt. Functional redundancy errors, when configured as a Checker, will cause $\overline{\text{IERR}}$ to be asserted in the second clock after the mismatched output value was detected.

$\overline{\text{IGNNE}}$

Ignore Numeric Error ***Input***

The $\overline{\text{IGNNE}}$ signal is used in conjunction with the NE bit in CR0 to control response to numeric errors in the floating-point unit. Numeric errors are handled internally when the NE bit is set. When the NE bit is not set, errors are reported if $\overline{\text{IGNNE}}$ is asserted and ignored when negated. (See Switching Characteristics t_{28} and t_{29}.)

INIT

Initialize ***Input***

The processor will perform a warm initialization when the INIT signal is asserted. The INIT signal is similar to the RESET signal except that the data buffers, data cache, floating-point registers, instruction cache, and SMBASE registers are not modified. The processor will perform a self-test if the INIT signal is sampled High at the falling edge of RESET.

INTR

Maskable Interrupt ***Input***

The INTR signal is used to generate interrupts. The interrupt number is transferred to the processor during the interrupt acknowledge cycle. To ensure that interrupts are acknowledged, the INTR signal must be asserted until a locked interrupt acknowledge cycle is complete. The INTR can be masked by clearing the IF bit in the EFLAGS register. (See Switching Characteristics t_{26} and t_{27}.)

INV *Invalidation* *Input*

The INV signal is used to designate the MESI protocol state of the cache line for inquire cycles that result in hits. This signal is sampled on the same clock that $\overline{\text{EADS}}$ is asserted. Sampling INV Low will result in the shared state, while sampling INV High will result in the invalid state.

$\overline{\text{KEN}}$ *Cache Enable* *Input*

$\overline{\text{KEN}}$ is asserted to enable caching. Caching is disabled when $\overline{\text{KEN}}$ is negated. Returning $\overline{\text{KEN}}$ asserted with the first $\overline{\text{BRDY}}$ or $\overline{\text{NA}}$ of a cacheable cycle causes the line to be placed in the cache. Returning it negated transforms the cycle into a non-cacheable, single-cycle read. $\overline{\text{KEN}}$ has a small internal pull-up resistor. (See Switching Characteristics t_{18a} and t_{19}.)

$\overline{\text{LOCK}}$ *Bus Lock* *Output*

The $\overline{\text{LOCK}}$ signal is asserted to indicate locked cycles, and is asserted during the first clock of a locked cycle. It is negated after $\overline{\text{BRDY}}$ is sampled for the last locked bus cycle. A HOLD request will not be acknowledged during locked cycles, but AHOLD and $\overline{\text{BOFF}}$ are allowed during locked cycles.

M/$\overline{\text{IO}}$ *Memory/ Input-Output* *OUTPUT*

The M/$\overline{\text{IO}}$ signal is used with other control signals to determine bus cycle type. These cycles are defined in Table 5 and Table 6 on page 27. M/$\overline{\text{IO}}$ is driven active with $\overline{\text{ADS}}$.

$\overline{\text{NA}}$ *Next Address* *Input*

$\overline{\text{NA}}$ is asserted when external memory is prepared to accept a pipelined cycle. $\overline{\text{NA}}$ does not generate pipelined cycles when $\overline{\text{LOCK}}$ is asserted, during writeback cycles, or when there are no pending internal cycles. Furthermore, locked or writeback cycles are not pipelined. $\overline{\text{KEN}}$ and WB/$\overline{\text{WT}}$ are sampled when $\overline{\text{NA}}$ or $\overline{\text{BRDY}}$ is asserted, whichever comes first.

NMI *Non-maskable Interrupt* *Input*

Asserting the NMI signal generates a non-maskable interrupt. The NMI input is rising-edge sensitive. The NMI signal must be held Low for at least one clock before its rising edge.

PCD *Page Cache Disable* *Output*

The PCD signal provides cacheability status by reporting the contents of the PCD bit in CR3, the page directory, or the page table entry. PCD reflects the state of the PCD bit in CR3 if non-paged cycles occur. In Real mode or Protected mode when paging is disabled, PCD reflects the state of the CD bit in CR0.

$\overline{\text{PCHK}}$ *Parity Status* *Output*

The $\overline{\text{PCHK}}$ signal is asserted to indicate a data parity error for data read cycles. It may be sampled for parity status on the second clock after $\overline{\text{BRDY}}$ is sampled as asserted. Except during Test mode, $\overline{\text{PCHK}}$ is never floated.

$\overline{\text{PEN}}$ *Parity Enable* *Input*

$\overline{\text{PEN}}$, when asserted on a parity error, causes the address and control signals of the cycle to be latched into the machine check registers. The MCE bit in CR4, if set, will cause a vector to the machine check exception before another instruction is executed.

PRDY *Probe Ready* *Output*

The processor asserts PRDY to acknowledge the system logic's assertion of R/$\overline{\text{S}}$ or execution of the Test Access Port (TAP) instruction, USEHDT, and to indicate the processor's entry into the Hardware Debug Tool (HDT) mode for debugging.

PWT *Page Write-Through* *Output*

The PWT signal provides writeback status by reporting the contents of the PWT bit in CR3, the page directory, or the page table entry. The PWT signal reflects the state of the PWT bit in CR3 when non-paged cycles occur or paging is disabled. In Real mode or Protected mode, when paging is disabled, PWT will be zero.

RESET *Reset* *Input*

The processor will reset when the RESET signal is asserted. The processor cannot begin execution until at least 1 ms after V_{CC}, BF, and CLK have stabilized. The operating mode is determined by the state of the $\overline{\text{FLUSH}}$, INIT, and $\overline{\text{FRCMC}}$ signals during the falling edge of RESET. (See $\overline{\text{FLUSH}}$, INIT, $\overline{\text{FRCMC}}$, and Switching Characteristics t_{36} and t_{37}.)

R/$\overline{\text{S}}$ *Run/STOP* *Input*

The R/$\overline{\text{S}}$ signal provides an edge-sensitive interrupt to stop normal execution. A falling-edge transition halts execution at the next instruction boundary. A rising-edge transition, which must not occur before PRDY is asserted, resumes execution.

SCYC *Split Cycle* *Output*

SCYC indicates split cycles when $\overline{\text{LOCK}}$ is asserted. This signal indicates that more than two cycles will be locked together for misaligned locked transfers.

$\overline{\text{SMI}}$ *System Management Interrupt* *Input*

$\overline{\text{SMI}}$ allows external logic to request a non-maskable system management interrupt. Asserting this signal will cause the processor to suspend normal execution and enter System Management Mode (SMM) at the next instruction boundary.

$\overline{\text{SMIACT}}$ *SMI Active* *Output*

$\overline{\text{SMIACT}}$ is asserted when the processor is operating in SMM.

$\overline{\text{STPCLK}}$ *Stop Clock* *Input*

$\overline{\text{STPCLK}}$, when asserted, causes the processor to complete the current instruction and issue a stop grant bus cycle. Once the stop grant is issued, the processor stops the clock, retaining the ability to execute inquire cycles.

TCK *Test Clock* *Input*

TCK is a test clock signal. It conforms to the IEEE-1149.1 boundary scan interface.

TDI *Test Data Input* *Input*

The TDI signal is a serial input for test data and TAP instructions. The instructions or data are sampled on the rising edge of the TCK signal.

TDO *Test Data Output* *Output*

The TDO signal is a serial output for test data and TAP instructions. TDO is updated on the falling edge of the TCK signal.

TMS ***Test Mode Select*** ***Input***

The TMS signal is used to select the TAP Test modes. This signal is sampled on the rising edge of the TCK. TMS has an internal pull-up resistor.

$\overline{\text{TRST}}$ ***Test Reset*** ***Input***

Asserting $\overline{\text{TRST}}$ initializes the TAP controller.

W/$\overline{\text{R}}$ ***Write/Read*** ***Output***

The W/$\overline{\text{R}}$ signal is used with other control signals to distinguish bus cycles and special cycles. These cycles are defined in Table 5 and Table 6 on page 27. W/$\overline{\text{R}}$ is driven active with $\overline{\text{ADS}}$, and floated with $\overline{\text{BOFF}}$ and bus hold.

WB/$\overline{\text{WT}}$ ***Writeback/Writethrough*** ***Input***

The state of WB/$\overline{\text{WT}}$ determines the MESI cache protocol state of a data line during cache line fills. When the signal is driven High, the cache line will be loaded in the exclusive state. When the signal is driven Low, the cache line will be loaded in the shared state.

Table 1. Input Pins

Name	Type	Note	Name	Type	Note
$\overline{\text{A20M}}$	Asynchronous	Note 1	$\overline{\text{IGNNE}}$	Asynchronous	
AHOLD	Synchronous		INIT	Asynchronous	
BF	Synchronous	Note 2	INTR	Asynchronous	
$\overline{\text{BOFF}}$	Synchronous		INV	Synchronous	Note 5
$\overline{\text{BRDY}}$	Synchronous		$\overline{\text{KEN}}$	Synchronous	Note 6
$\overline{\text{BRDYC}}$	Synchronous		$\overline{\text{NA}}$	Synchronous	
$\overline{\text{BUSCHK}}$	Synchronous	Note 3	NMI	Asynchronous	

Notes:

1. *$\overline{\text{A20M}}$ may change during RESET or during a serializing event like an I/O write. A state change at other times will result in incorrect address generation on subsequent memory cycles.*
2. *BF and $\overline{\text{FRCMC}}$ are normally connected to V_{CC} or V_{SS} by a jumper. For correct operation, any change on these signals should be followed by a RESET.*
3. *$\overline{\text{BUSCHK}}$ is sampled in every clock. Any asserted sample is remembered and takes effect on the same clock as the last $\overline{\text{BRDY}}$.*
4. *These are sampled in the same clock as $\overline{\text{BRDY}}$.*
5. *This is sampled in the same clock as $\overline{\text{EADS}}$.*
6. *These are sampled with the first $\overline{\text{BRDY}}$ or $\overline{\text{NA}}$ and must meet setup to every clock*

Table 1. Input Pins (continued)

Name	Type	Note	Name	Type	Note
CLK	Clock		$\overline{\text{PEN}}$	Synchronous	Note 4
$\overline{\text{EADS}}$	Synchronous		RESET	Asynchronous	
$\overline{\text{EWBE}}$	Synchronous	Note 4	R/$\overline{\text{S}}$	Asynchronous	
$\overline{\text{FLUSH}}$	Asynchronous		$\overline{\text{SMI}}$	Asynchronous	
$\overline{\text{FRCMC}}$	Asynchronous	Note 2	$\overline{\text{STPCLK}}$	Asynchronous	
HOLD	Synchronous		WB/$\overline{\text{WT}}$	Synchronous	Note 6

Notes:

1. *$\overline{\text{A20M}}$ may change during RESET or during a serializing event like an I/O write. A state change at other times will result in incorrect address generation on subsequent memory cycles.*
2. *BF and $\overline{\text{FRCMC}}$ are normally connected to V_{CC} or V_{SS} by a jumper. For correct operation, any change on these signals should be followed by a RESET.*
3. *$\overline{\text{BUSCHK}}$ is sampled in every clock. Any asserted sample is remembered and takes effect on the same clock as the last $\overline{\text{BRDY}}$.*
4. *These are sampled in the same clock as $\overline{\text{BRDY}}$.*
5. *This is sampled in the same clock as $\overline{\text{EADS}}$.*
6. *These are sampled with the first $\overline{\text{BRDY}}$ or $\overline{\text{NA}}$ and must meet setup to every clock*

Table 2. Output Pins

Name	Floated At (Note 1)	Name	Floated At (Note 1)
A4–A3	Bus Hold, Address Hold, $\overline{\text{BOFF}}$	HLDA	Always Driven
$\overline{\text{ADS}}$	Bus Hold, $\overline{\text{BOFF}}$	$\overline{\text{IERR}}$	Always Driven
$\overline{\text{ADSC}}$	Bus Hold, $\overline{\text{BOFF}}$	$\overline{\text{LOCK}}$	Bus Hold, $\overline{\text{BOFF}}$
$\overline{\text{APCHK}}$	Always Driven	M/$\overline{\text{IO}}$	Bus Hold, $\overline{\text{BOFF}}$
$\overline{\text{BE7}}$–$\overline{\text{BE0}}$	Bus Hold, $\overline{\text{BOFF}}$	PCD	Bus Hold, $\overline{\text{BOFF}}$
BREQ	Always Driven	$\overline{\text{PCHK}}$	Always Driven
$\overline{\text{CACHE}}$	Bus Hold, $\overline{\text{BOFF}}$	PRDY	Always Driven
D/$\overline{\text{C}}$	Bus Hold, $\overline{\text{BOFF}}$	PWT	Bus Hold, $\overline{\text{BOFF}}$
$\overline{\text{FERR}}$	Always Driven	SCYC	$\overline{\text{LOCK}}$ not asserted, Bus Hold, $\overline{\text{BOFF}}$
$\overline{\text{HIT}}$	Always Driven	$\overline{\text{SMIACT}}$	Always Driven
$\overline{\text{HITM}}$	Always Driven	W/$\overline{\text{R}}$	Bus Hold, $\overline{\text{BOFF}}$

Notes:

1. *All outputs float during Tri-State test mode.*

Table 3. Input/Output Pins

Name	When Floated
A31–A5	Bus Hold, Address Hold, $\overline{\text{BOFF}}$
AP	Bus Hold, Address Hold, $\overline{\text{BOFF}}$
D63–D0	Bus Hold, $\overline{\text{BOFF}}$
DP7–DP0	Bus Hold, $\overline{\text{BOFF}}$

Table 4. Test Pins

Name	Type	Note
TCK	Input	
TDI	Input	Sampled on the rising edge of TCK.
TDO	Output	Driven on the falling edge of TCK.
TMS	Input	Sampled on the rising edge of TCK.
$\overline{\text{TRST}}$	Input	

Table 5. Bus Cycle Definition

Bus Cycle Initiated	Generated by CPU				Generated by System
	M/$\overline{\text{IO}}$	D/$\overline{\text{C}}$	W/$\overline{\text{R}}$	$\overline{\text{CACHE}}$	$\overline{\text{KEN}}$
Code Read, Instruction Cache Line Fill	1	0	0	0	0
Code Read, Noncacheable	1	0	0	1	x
Code Read, Noncacheable	1	0	0	x	1
Encoding for Special Cycle	0	0	1	1	x
Interrupt Acknowledge	0	0	0	1	x
I/O Read, Noncacheable	0	1	0	1	x
I/O Write, Noncacheable	0	1	1	1	x
Memory Read, Data Cache Line Fill	1	1	0	0	0
Memory Read, Noncacheable	1	1	0	1	x
Memory Read, Noncacheable	1	1	0	x	1
Memory Write, Data Cache Writeback	1	1	1	0	x
Memory Write, Noncacheable	1	1	1	1	x

Table 6. Special Cycles

Special Cycle	A4	$\overline{\text{BE7}}$	$\overline{\text{BE6}}$	$\overline{\text{BE5}}$	$\overline{\text{BE4}}$	$\overline{\text{BE3}}$	$\overline{\text{BE2}}$	$\overline{\text{BE1}}$	$\overline{\text{BE0}}$	M/$\overline{\text{IO}}$	D/$\overline{\text{C}}$	W/$\overline{\text{R}}$	$\overline{\text{CACHE}}$	$\overline{\text{KEN}}$
Branch Trace	0	1	1	0	1	1	1	1	1	0	0	1	1	x
Flush (INVD, WBINVD execution)	0	1	1	1	1	1	1	0	1	0	0	1	1	x
Flush Acknowledge ($\overline{\text{FLUSH}}$ asserted Low)	0	1	1	1	0	1	1	1	1	0	0	1	1	x
Halt	0	1	1	1	1	1	0	1	1	0	0	1	1	x
Shutdown	0	1	1	1	1	1	1	1	0	0	0	1	1	x
Stop Clock Acknowledge	1	1	1	1	1	1	0	1	1	0	0	1	1	x
Writeback (WBINVD execution)	0	1	1	1	1	0	1	1	1	0	0	1	1	x

8 Processor Operation

8.1 Power-On Configuration

The AMD-K5 processor signals at reset are listed in Table 7.

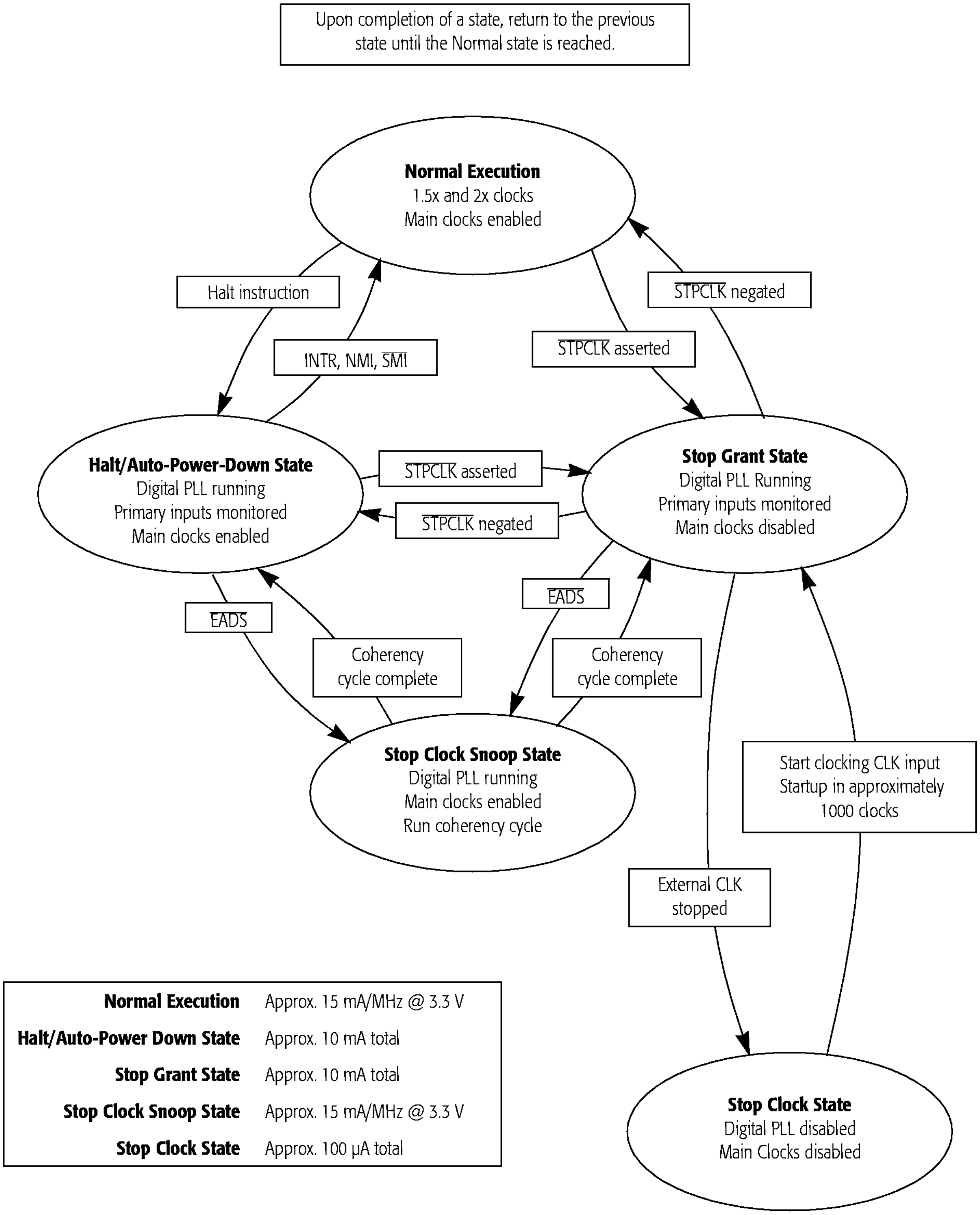
Table 7. Signals at Reset

Output	State at Reset	Output	State at Reset
Address	Float	$\overline{\text{FERR}}$	1
$\overline{\text{ADS}}$	1	$\overline{\text{HIT}}$	1
$\overline{\text{APCHK}}$	1	$\overline{\text{HITM}}$	1
$\overline{\text{BE7}}$–$\overline{\text{BE0}}$	Undefined	HLDA	0
$\overline{\text{BRDY}}$	1	$\overline{\text{LOCK}}$	1
$\overline{\text{BRDYC}}$	1	M/$\overline{\text{IO}}$	Undefined
BREQ	0	PCD	Undefined
CACHE	Undefined	$\overline{\text{PCHK}}$	1
D/$\overline{\text{C}}$	Undefined	PRDY	0
Data	Float	PWT	Undefined
DP7–DP0	Float	W/$\overline{\text{R}}$	Undefined

8.2 Clock State

The AMD-K5 processor uses the Enhanced 486 protocol to control the clock. This protocol provides for stopping the clock from hardware using the $\overline{\text{STPCLK}}$ control signal, or from software using the HALT instruction. During the clock-stopped states, cache coherency is maintained by temporarily enabling the clock for snoop processing and recognizing HOLD/HLDA arbitration sequences.

A state transition diagram for a stop clock state machine implementing five clocking states—the Enhanced 486 protocol—is illustrated in Figure 3 on page 29.



Normal Execution	Approx. 15 mA/MHz @ 3.3 V
Halt/Auto-Power Down State	Approx. 10 mA total
Stop Grant State	Approx. 10 mA total
Stop Clock Snoop State	Approx. 15 mA/MHz @ 3.3 V
Stop Clock State	Approx. 100 µA total

Figure 3. State Transition Diagram for Stop Clock State Machine

Normal Execution State

In this state, the AMD-K5 processor operates at full speed. All clocks are running.

Halt/Auto-Power-Down State

In this state, most internal clocks are stopped. The Phase Lock Loop (PLL) is operating and certain bus interface components are clocked. Instruction execution is disabled. This aids in timely detection of inquire cycles and HOLD/HLDA sequences, while greatly reducing power consumption.

The Halt/Auto-Power-Down State is entered from normal execution state by executing the HALT instruction in Real mode or Protected mode. The clock state will return to normal execution state when an interrupt, non-maskable interrupt, system management interrupt, power-on reset, or soft reset is detected (INTR, NMI, $\overline{\text{SMI}}$, RESET, or INIT, respectively). The clock state may temporarily transition from Halt/Auto-Power-Down State to Stop Clock Snoop State to process an inquire cycle or to Stop Grant State in response to a $\overline{\text{STPCLK}}$. In these cases, the clock state will return to Halt/Auto-Power-Down State and wait for one of the interrupt conditions when the secondary condition is removed.

Stop Grant State

In this state, most internal clocks are stopped. The PLL is operating and certain bus interface components are clocked. Instruction execution is disabled. This allows timely detection of inquire cycles and HOLD/HLDA sequences, while greatly reducing power consumption.

The Stop Grant State is entered from Normal Execution State or Halt/Auto-Power-Down State by asserting the $\overline{\text{STPCLK}}$ pin. When $\overline{\text{STPCLK}}$ is sampled as asserted, the current instruction is completed, all processing is stopped, a Stop Grant bus cycle is generated, and the clock is shut down. The clock state will return to its previous state when $\overline{\text{STPCLK}}$ is negated. Once asserted, $\overline{\text{STPCLK}}$ must not be negated until the Stop Grant Acknowledge special cycle is seen. The clock state may temporarily transition from Stop Grant State to Stop Clock Snoop State to process an inquire cycle, or to Stop Clock State to process a Stop Clock request. In these cases, the clock state will return to Stop Grant State when the secondary condition is removed.

$\overline{\text{STPCLK}}$ is treated as the lowest priority external interrupt. If a higher priority external interrupt exists (power-on reset, soft reset, flush, system management interrupt, non-maskable interrupt, or maskable interrupt), recognition of $\overline{\text{STPCLK}}$ is delayed until the interrupt processing is complete. However, assertion of a higher priority interrupt will not cause the Stop Grant State to be exited.

Stop Clock Snoop State

In this state, all internal clocks are running and an inquire cycle is being performed. Instruction execution is disabled and HOLD/HLDA operate normally.

Stop Clock Snoop State is entered from Halt/Auto-Power-Down State or Stop Grant State when an inquire cycle is detected. This is a temporary state, lasting only until the coherency operation (snoop/miss, snoop/invalidate or snoop/writeback) is complete. The clock state will then return to the previous state. (See Figure 24 on page 72.)

Stop Clock State

In this state, all internal clocks are stopped, the PLL is shut down, and all execution is disabled. If HOLD is asserted while the clock is running, HLDA will be generated and the buses floated. If HOLD is negated, HLDA will be negated and the buses will be driven to their previous state without regard to whether the clock is running. This is the lowest power state.

The Stop Clock State is entered from the Stop Grant State by stopping the CLK. The clock state returns to Stop Grant State when the CLK is again started. The time required to restart the CLK and enter the Stop Clock State is approximately 1000 clock cycles.

8.3 Cache Protocol

Internal Cache

The AMD-K5 processor has a 16-Kbyte dual-tagged instruction cache with 32-byte lines and an 8-Kbyte dual-tagged data cache of 32-byte lines. Cache lines refill in four transfer burst cycles from memory and align along 32-byte lines.

The operating mode is software-controlled, and on-chip caches must be enabled by software. This is accomplished by clearing or setting the CD and NW bits of CR0.

Any area of memory can be cached. Software can prevent areas of memory from being cached by setting the PCD bit in the corresponding page table entry. Hardware can prevent areas of memory from being cached through the $\overline{\text{KEN}}$ pin.

The AMD-K5 processor uses the MESI protocol—2 bits per cache line—in its data cache to ensure consistency in multiprocessing systems. The physical tags of both the instruction and data cache are accessed and compared during each inquire cycle to maintain a consistent copy of data.

Cacheability

The PCD and PWT bits in the page directory and page-table entry control caching on a page-by-page basis. The PCD and PWT bits manage page caching and drive processor PCD and PWT output pins.

PCD affects the cacheability of pages in the internal cache. The PWT bit determines whether the writethrough or writeback policy is used for this particular page.

Copy-Back Buffers

A one-line copy-back buffer is employed within the AMD-K5 processor to temporarily hold a modified entry being replaced in the data cache. The replaced line is stored in the copy-back buffer at the same time the read request for the replacement line is sent externally. Following completion of the read access, the modified line in the copy-back buffer is written back to memory. The copy-back buffer is snooped during inquire cycles.

A requested-word-first protocol is implemented by the AMD-K5 processor. Following receipt of the first data item, execution continues while the following three entries of the line are being fetched. The line is not marked valid until the last entry is stored in the cache.

8.4 Data Cache Coherency

Throughout this discussion, the MESI states may be abbreviated as follows:

M—Modified Exclusive State
E—Exclusive State
S—Shared State
I—Invalid State

Cache Invalidation

$\overline{\text{FLUSH}}$ writes back all modified lines and then invalidates all cache lines and generates a Flush Acknowledge special cycle to instruct the L2 cache to invalidate all lines.

The INVD instruction invalidates the entire cache and generates a Flush special cycle to instruct the L2 cache to invalidate all lines.

The WBINVD instruction writes back and invalidates all cache lines, generates a Write Back special cycle to instruct the L2 cache to write back all lines, and then generates a Flush special cycle to instruct the L2 cache to invalidate all lines.

Read Cycles

The cache response to processor-generated reads is described in Table 8. Processor reads that hit in the data cache require no external data cycle. The data is provided by the cache. Processor reads that miss in the data cache generate a read-allocate operation, including an external bus cycle. The action of the cache is dependent on the system response to that cycle. The cache state transition for read cycles is also described in Table 8.

A read allocate begins by selecting the way in the cache to be replaced at random.

Table 8. Processor Reads to Data Cache

State	$\overline{\text{CACHE}}$	$\overline{\text{KEN}}$	WB/$\overline{\text{WT}}$	PWT	Next State	Note
M	x	x	x	x	M	1
E	x	x	x	x	E	1
S	x	x	x	x	S	1
I	0	0	1	0	E	2
I	0	0	0	x	S	3
I	1	x	x	x	I	4
I	x	1	x	x	I	4, 5

Notes:

1. *A read cycle hit: Data is provided directly from the cache.*
2. *A read cycle miss: Selects the line for replacement; writes back the replaced line if it is modified (otherwise, discards the line). The line is cached as writeback.*
3. *A read cycle miss: Selects the line for replacement; writes back the replaced line if it is modified (otherwise discards the line). The line is cached as writethrough.*
4. *A read cycle miss: The line is not cacheable.*
5. *Within the cache directory, the Invalid state indicates that the cache entry contains no valid data. For purposes of hit/miss determination, the Invalid state indicates that the referenced cache line is not present in the cache. When a line is selected for replacement, all invalid ways are selected before any valid data is displaced from the cache.*

If the selected line is *not* modified, the data is discarded and the read of the new line is begun. When the first quad word of the new line is received, it is forwarded to the execution units. When all four quad words are available, they are copied to the cache line at the selected way and the cache status is updated.

If the selected line is modified, the read of the new line is begun at the same time the contents of the replaced line are copied to the copy-back buffer. When the first quad word of the new line is received, it is forwarded to the execution units. Execution continues concurrently as the rest of the block is received. When all four quad words are available, they are copied to the cache line at the selected way and the cache status is updated. Concurrently, the contents of the replaced line are written to memory.

Write Cycles

Processor writes that hit in modified or exclusive lines in the data cache require no external data cycle. The data is updated in the cache. Processor writes that hit shared lines of the data cache update the data cache and memory. The status returned with the writethrough bus cycle determines the final state of the line.

If write allocate is enabled in the AMD-K5 processor, processor writes that miss in the data cache generate an external data cache read cycle followed by a write hit. If write allocate is not enabled in the AMD-K5 processor, write misses generate an external write cycle only.

Write Allocate

Write allocate is an operating mode of the AMD-K5 processor that causes cache write misses to either proceed as normal write misses or to be converted to data cache line fills followed by cache write hits. The write allocate feature provides improved performance on repeat accesses to write-allocated data cache lines. The load/store unit in the processor determines whether each cache write miss is write-allocatable by whether it falls in or out of the ranges specified in the memory range registers.

For details on the implementation of write allocate, refer to the *AMD-K5 Processor Software Development Guide*, order# 20007.

Before the write cycle occurs for a write miss with write allocate enabled, an external data cache read cycle occurs that follows the normal rules for read allocate, and the intermediate state of the filled data cache line depends on the result of the read cycle as shown in Table 8. The final state of the data cache line is determined as shown in Table 9 by the transition from the intermediate read state (M, E, S, or I) to the final state (M, E, S, or I) after the write hit to the cache line.

Note: *In write allocate mode, replaced data cache lines are handled in the same way as during read allocate.*

Table 9. Writes to Data Cache

State	$\overline{\text{CACHE}}$	$\overline{\text{KEN}}$	WB/$\overline{\text{WT}}$	PWT	Next State	Note
M	x	x	x	x	M	1
E	x	x	x	x	M	2
S	0	0	1	0	E	3
S	0	0	0	x	S	3
	0	0	x	1		
I	x	x	x	x	I	4

Notes:

1. *A write hit to modified line: writes data to the cache.*
2. *A write hit to exclusive line: writes data to the cache.*
3. *A write hit to shared line: writes data to the cache and memory; invalidates any shared copy in the other cache.*
4. *If write allocate mode is not enabled, an invalid line always remains invalid. If write allocate mode is enabled, the intermediate state of the filled data cache line depends on the result of the read cycle as shown in Table 8, and the final state of the data cache line is determined by the intermediate state as applied to this table.*

External Inquire Cycles

The processor supports inquire cycles for both instruction and data caches to maintain cache coherency. Inquire cycles are initiated with the assertion of $\overline{\text{EADS}}$ and result in a snoop to both the instruction and data caches. The snoop operation is performed using the physical tag arrays that are maintained for this purpose. The snoop operation runs concurrently with internal processor operation. The results of the snoop operation are indicated on the $\overline{\text{HIT}}$ and $\overline{\text{HITM}}$ pins. The results of the inquire cycles are described in Table 10. (See Figure 25 on page 72 and Figure 26 on page 73.)

Table 10. Inquire Cycles to Data Cache

State	INV	Next State	Note
M	0	S	Snoop hit to modified line: Assert HIT and HITM, Write back modified data to memory, Negate HITM, Transition cache state when complete.
	1	I	
E	0	S	Snoop hit to unmodified line: Assert HIT, Transition cache state
	1	I	
S	0	S	Snoop hit to unmodified line: Assert $\overline{\text{HIT}}$, Transition cache state
	1	I	
I	x	I	Snoop miss: Negate $\overline{\text{HIT}}$.

Instruction Cache Coherency

The instruction cache protocol is a subset of the data cache protocol where only Invalid and Shared states are implemented. Read hits provide the data to the processor. Read misses result in a read allocate operation that loads the line into the cache and the data is provided to the processor. The first data is provided as soon as it arrives from memory.

Write cycles are never generated to the instruction cache, but inquire cycles may hit in the instruction cache, resulting in the cache line being invalidated.

Self-Modifying Code and the Cache

A snoop write hit to the instruction cache is treated as self-modifying code. The cache line is invalidated and all instructions in the instruction pipeline are flushed. Execution restarts at the instruction following the one causing the snoop. This guarantees exact execution of cacheable self-modifying code. For non-cacheable code, a jump should be placed between the modification of the code and its execution.

8.5 External Bus Description

The AMD-K5 processor external bus is identical to the P54C 64-bit bus, and will run at 1.5x or 2.0x multiples of the external bus frequency. The bus state transitions are illustrated in Figure 4.

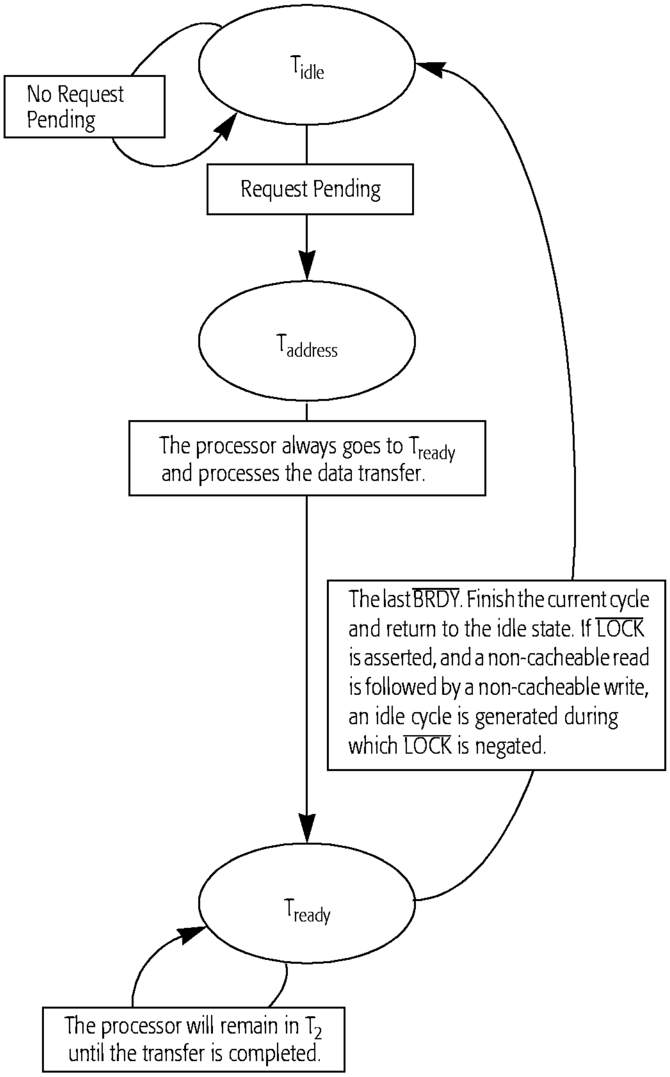

T_{idle}
No Request Pending
Request Pending
$T_{address}$
The processor always goes to T_{ready} and processes the data transfer.
The last $\overline{BRDY}$. Finish the current cycle and return to the idle state. If $\overline{LOCK}$ is asserted, and a non-cacheable read is followed by a non-cacheable write, an idle cycle is generated during which $\overline{LOCK}$ is negated.
T_{ready}
The processor will remain in T_2 until the transfer is completed.

Figure 4. Bus State Transitions

Memory Organization

Physical memory address space ranges from 0000_0000h to FFFF_FFFFh. Memory space is organized in 64-bit sections. Each 64-bit section has 8 bytes at consecutive memory addresses. The first address of each group is evenly divisible by 8, and each group is addressed by A31–A3. Since the protocol does not implement A2–A0 when interfacing to 32-bit, 16-bit, or 8-bit memories, the lower portions of the address must be determined by decoding the eight byte-enable signals. The address space of I/O begins at 0000_0000h and ends at 0000_FFFFh. I/O space is organized as a sequence of 8-bit quantities.

Memory objects can be 8, 16, 32, or 64 bits. I/O objects are 8, 16, or 32 bits. Both appear as fields on the 64-bit data bus.

Data is transferred on the byte lines corresponding to the address. 16-bit or 32-bit objects crossing a 32-bit boundary, or 64-bit objects crossing a 64-bit boundary, are misaligned and will require multiple cycles to transfer.

The byte-enable signals and the data lines correspond in the following manner:

- $\overline{BE7}$: D63–D56
- $\overline{BE6}$: D55–D48
- $\overline{BE5}$: D47–D40
- $\overline{BE4}$: D39–D32
- $\overline{BE3}$: D31–D24
- $\overline{BE2}$: D23–D16
- $\overline{BE1}$: D15–D8
- $\overline{BE0}$: D7–D0

8.6 Bus Cycles

Bus cycles encode normal read and write accesses to code or data space and handle special events such as interrupt acknowledge. The type of cycle is determined by the $\overline{CACHE}$, D/$\overline{C}$, M/$\overline{IO}$, and W/$\overline{R}$ outputs. The processor encodes information with the byte-enable signals for special bus cycles. (See Table 6 on page 27.)

If M/$\overline{IO}$ is asserted Low or PCD is driven High in any cycle, $\overline{CACHE}$ is not asserted. The processor uses a burst transfer of four 64-bit accesses, corresponding to the 32-byte line size of the caches, for bus cycles involving cache line movement.

Table 11 shows the order of burst accesses expected by the external protocol.

Table 11. Addressing of the AMD-K5 Processor Burst Order

If 1st Address = 0	then 8	then 10	then 18
If 1st Address = 8	then 0	then 18	then 10
If 1st Address = 10	then 18	then 0	then 8
If 1st Address = 18	then 10	then 8	then 0

Single Transfer Cycles

Single transfer cycles are initiated with the assertion of $\overline{\text{ADS}}$ while negating the cache signal. The cycle is completed when the $\overline{\text{BRDY}}$ signal is asserted by the external system. A single transfer cycle requires a minimum of two external clock cycles.

Timing for a single write transfer cycle is illustrated in Figure 5. (See Figures 28, 29, 30, and 31 beginning on page 74.)

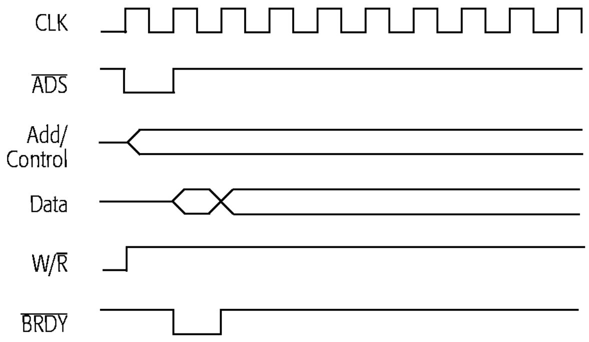


Figure 5. Single Writes (Zero Wait States)

Burst Read Cycles

The size of a burst read access is always 32 bytes sent as four 64-bit transfers. A burst read access is indicated by the assertion of the $\overline{\text{CACHE}}$ signal, but if the external memory system subsequently does not assert $\overline{\text{KEN}}$, the access will be converted to a single access. Data is sampled during the same clock that $\overline{\text{BRDY}}$ is asserted. Wait states can be added by negating $\overline{\text{BRDY}}$.

The initial address and the byte enables are not changed after the initial access of a burst. External hardware must be configured to determine the subsequent addresses of the burst in accordance with the ordering specified in Table 11. $\overline{\text{PCHK}}$ is driven two clocks following an associated data transfer to the processor to indicate a data parity error. (See Figure 31 on page 75 and Figure 32 on page 76.)

Burst Write Cycles

Like a burst read access, a burst write access is indicated by the assertion of the $\overline{\text{CACHE}}$ pin. Burst write cycles (an example of which is given in Figure 6) only occur for writebacks of modified lines in the processor data cache. These transfers are always four accesses. The address order for writeback cycles is always 0, 8, 10, 18. All other accesses, including unaligned accesses that cross 64-bit aligned boundaries, are sent as single accesses or a series of single accesses. Negating $\overline{\text{BRDY}}$ until the external memory system is ready to receive data adds additional wait states, if they are needed. The processor ceases driving the current data element upon receiving the BRDY signal. (See Figure 33 on page 76.)

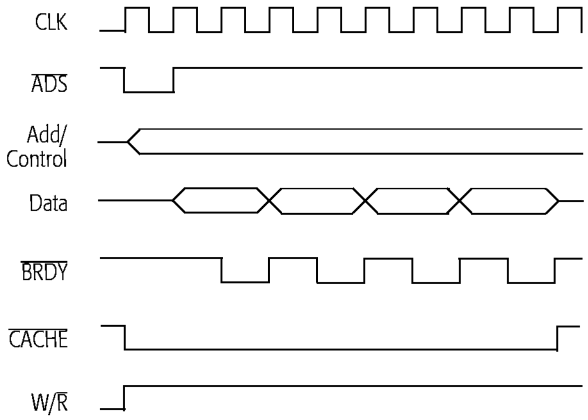


Figure 6. Burst Write (One Wait State)

The external signal $\overline{\text{KEN}}$ is ignored for burst write cycles since these are previously cached lines. Writebacks can occur as a result of the following:

- Replacement of a data cache entry that is modified
- An inquire cycle that hits in a modified line
- Assertion of the WBINVD instruction
- Assertion of the external signal $\overline{\text{FLUSH}}$

Only one line is sent for inquire or replacement accesses. Assertion of $\overline{\text{FLUSH}}$ or execution of WBINVD results in the modified lines in the entire cache being written back as a series of single line writes. An inquire or replacement access results in a writeback of only one line.

$\overline{\text{BOFF}}$ or AHOLD/ HOLD/HLDA During Burst Transfers

$\overline{\text{BOFF}}$ or AHOLD can be asserted during a burst transfer. The processor will abort a cycle if $\overline{\text{BOFF}}$ is asserted in the middle of the cycle. When $\overline{\text{BOFF}}$ is negated, the cycle is restarted from the beginning.

If AHOLD is asserted, the processor responds by floating the address pins in the next clock cycle. The system can then drive the address and assert $\overline{\text{EADS}}$ to generate an inquire cycle while the data cycle continues. Assertion of HOLD can occur at any time, but HLDA will not be asserted until pending cycles are completed.

To avoid excessive power drain, AHOLD should not be negated when BRDY is asserted during a write cycle, and when ADS is asserted at the beginning of a writeback cycle.

Use of BOFF

BOFF causes the processor to float its local bus on the next clock cycle and to terminate the current bus cycle (see Figure 7). BOFF is sampled every clock cycle. If both BOFF and BRDY are asserted during the same clock cycle, BRDY is ignored and the associated data transfer must be re-initiated. If BOFF is asserted while ADS is asserted, the processor floats ADS, even though it is in its asserted state. This situation must not be interpreted as the start of a cycle by the system.

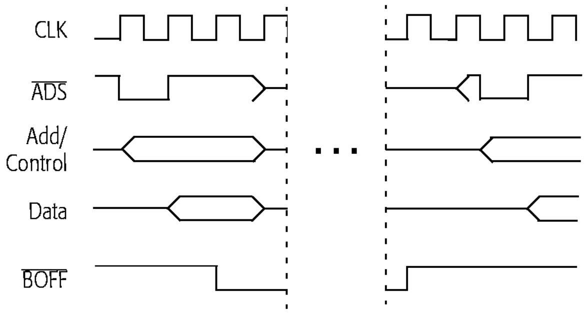


Figure 7. BOFF Timing

KEN must be reasserted by the system to enable caching on any cycle that was previously aborted by BOFF. If a burst cycle is aborted by the assertion of BOFF in the middle of the access, the initial state of KEN when the access began will be used when the cycle is restarted. KEN should be reasserted if caching is enabled for the cycle.

Any cycles aborted due to BOFF are recorded behind a pending writeback cycle that is scheduled in response to a snoop hit to a modified line. For example, if a cache line fill is aborted due to BOFF, and an external cycle hits a modified line, the cache line fill is completed after the modified line is written back.

Locked Operations

A locked cycle, illustrated in Figure 8, uses the $\overline{\text{LOCK}}$ pin to indicate that the processor is performing a read-modify-write, and that both the read operation and write operation must be allowed to complete as a combined operation. (See Figure 36, 40, and 42 beginning on page 78.)

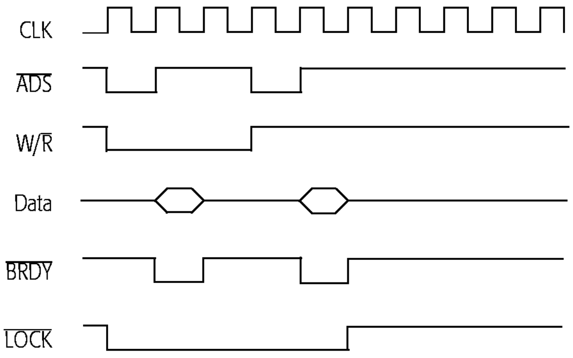


Figure 8. Locked Cycles

When the program generates a locked access, the processor first looks in the data cache. If the locked object is modified in the cache, it is written back to memory and invalidated. It is then accessed using a locked memory cycle. Since combined operations can access misaligned objects, locked operations can result in multiple writebacks, multiple locked reads, and multiple locked writes. When unaligned locked operations are performed, SCYC is asserted

$\overline{\text{LOCK}}$ during HOLD and $\overline{\text{BOFF}}$

An assertion of HOLD after a locked operation has initiated is ignored by the processor until after the entire locked operation has completed. Following completion, HLDA is asserted.

If $\overline{\text{BOFF}}$ is asserted during the read portion of a locked access, $\overline{\text{LOCK}}$ will float and the entire locked access will be restarted after $\overline{\text{BOFF}}$ is negated. If $\overline{\text{BOFF}}$ is asserted during the write portion of a locked access, $\overline{\text{LOCK}}$ will float and only the write will be restarted after $\overline{\text{BOFF}}$ is negated.

$\overline{\text{LOCK}}$ Operations during Inquire Cycles

Inquire cycles can be performed as usual during locked operations. Inquire cycles during atomic locked read and write operations are only allowed from the external inquire. No writebacks will be seen because the processor has already evicted the modified line.

The LOCK pin is asserted for the duration of locked accesses. Note also that at least one dead cycle will always be present between consecutive locked atomic read-modify-write operations. This will be noted by the negating of the $\overline{\text{LOCK}}$ pin for at least one clock period between consecutive locked accesses.

Locked Operation to Cached Lines

When a locked operation to a cached line occurs, the processor invalidates the line and determines whether the line is modified. If the line is modified, it is written back to memory. $\overline{\text{LOCK}}$ is not asserted during the writeback operation. $\overline{\text{LOCK}}$ is then asserted and the locked read-modify-write operations are performed. The line is not cached during these operations. SCYC is asserted for misaligned locked transfers.

Bus Hold

HOLD, illustrated in Figure 9 on page 44, is used to inform the processor that another bus device desires to be bus master. If HOLD is asserted, the processor completes all pending bus cycles and acknowledges release of the bus by asserting HLDA. When the bus is released, the processor floats the following outputs:

- A31–A3
- $\overline{\text{ADS}}$
- AP
- $\overline{\text{BE7}}$–$\overline{\text{BE3}}$
- $\overline{\text{CACHE}}$
- D/$\overline{\text{C}}$
- D63–D0
- DP7–DP0
- $\overline{\text{LOCK}}$
- M/$\overline{\text{IO}}$
- PCD
- PWT
- SCYC
- W/$\overline{\text{R}}$

These are the same outputs that are floated when $\overline{\text{BOFF}}$ is asserted. These outputs provide status information, but do not participate in the external memory system access.

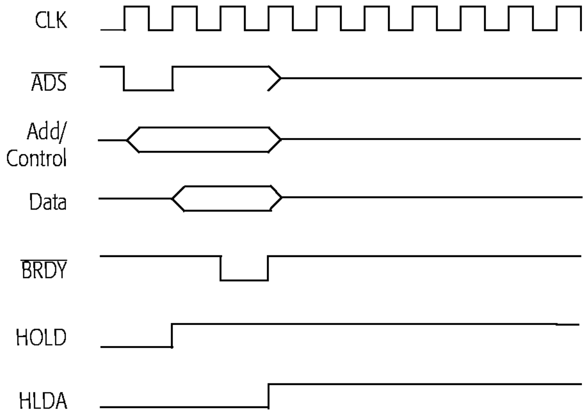


Figure 9. HOLD/HLDA Cycle

HLDA is negated one clock after HOLD is negated. Hold is not recognized during locked cycles, but is recognized during BOFF. An external master must monitor BOFF as well as HLDA to determine bus ownership.

Bus Error Support using PCHK and APCHK

PCHK and APCHK are used for checking data parity and address parity. Data parity is driven into the processor on pins DP7–DP0 during reads, and is driven out of the same pins during writes. The processor indicates a data parity error by asserting PCHK two clocks after the validation of parity by BRDY.

The AP signal provides even parity for the address bus. The processor indicates an inquire parity error by asserting APCHK two clock cycles after the address is validated by EADS.

Special Bus Cycles

Several bus cycles are supported by the AMD-K5 processor, as illustrated in Table 6 on page 27. The byte enables are encoded to define the type of cycle. Figure 39 on page 79 is a timing diagram of a generic special bus cycle.

Flush Operations

The FLUSH input is used by external logic to cause the processor to write back any modified lines in the data cache, and to invalidate all entries in both the data cache and the instruction cache. A special cycle is executed by the processor to indicate completion of the FLUSH operation. The FLUSH input is treated as a high-priority asynchronous interrupt, and is acknowledged only on instruction boundaries.

Interrupt Acknowledge

An interrupt acknowledge cycle, shown in Figure 10 on page 45, is a special cycle generated to acknowledge receipt of an interrupt at the INTR input. The processor generates an interrupt acknowledge cycle in a locked pair of transactions. The first transaction acknowledges the interrupt to the external system. The second transaction provides the interrupt vector to the processor. An idle cycle is generated between the transactions. An interrupt acknowledge cycle is completed upon assertion of $\overline{\text{BRDY}}$. (See Figure 40.)

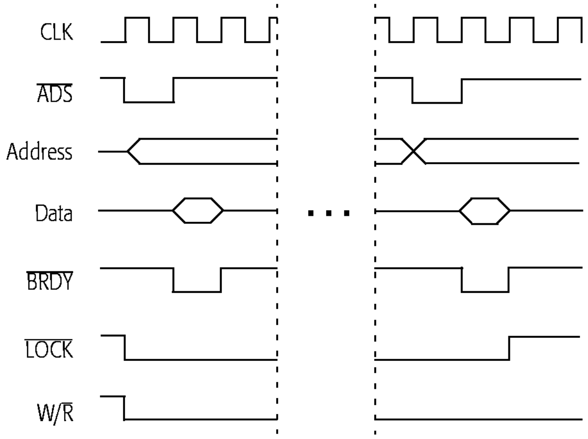


Figure 10. Interrupt Acknowledge Cycles

Inquire Cycles

An inquire cycle is employed to allow the system to determine whether a particular line is cached and modified. After obtaining ownership of the address bus using $\overline{\text{BOFF}}$, AHOLD, or HOLD, the system drives the physical address of the line on A31–A5, and marks the address valid with $\overline{\text{EADS}}$.

If the processor detects a hit in its instruction or data cache, the processor asserts the $\overline{\text{HIT}}$ signal two clock cycles after the assertion of $\overline{\text{EADS}}$ (see Figure 11 on page 46). If the line is modified (see Figure 12 on page 46), the processor asserts the $\overline{\text{HITM}}$ signal two clocks after the assertion of $\overline{\text{EADS}}$, and writes back the modified line. $\overline{\text{EADS}}$ is ignored during the writeback of the modified line. Initiation of the writeback of the modified line will occur no earlier than two clock cycles after $\overline{\text{HITM}}$ is asserted.

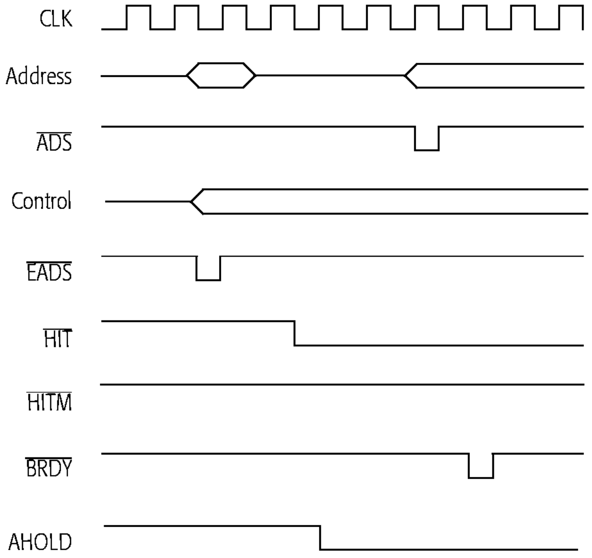


Figure 11. Inquire Cycle (Hit to a Non-Modified Line)

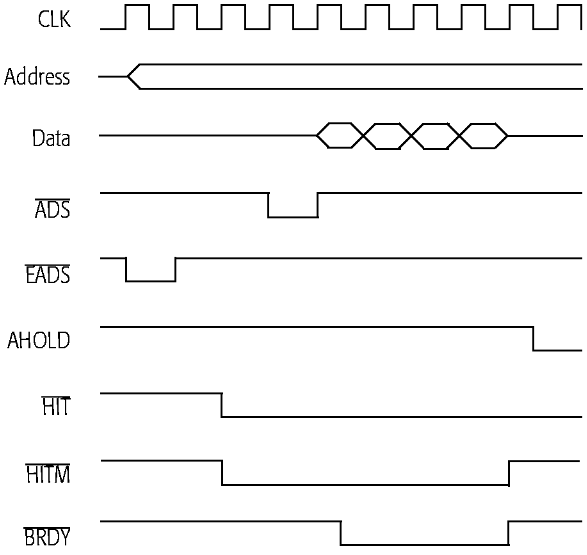


Figure 12. Inquire Cycle (Hit to a Modified Line)

The $\overline{\text{HIT}}$ signal retains its state between inquire cycles. The $\overline{\text{HITM}}$ signal remains asserted until the writeback of the modified line completes. Following completion of the writeback operation, the processor negates $\overline{\text{HITM}}$.

Pipelining

The following pipeline cycles are supported by AMD-K5 processors model 1 and model 2 with stepping level of 4 and above:

- Cacheable instruction cache cycle into a cacheable instruction cache cycle
- Cacheable instruction cache cycle into a cacheable data cache cycle
- Cacheable instruction cache cycle into a non-cacheable data cache cycle (could be I/O)
- Cacheable instruction cache cycle into a non-cacheable instruction cache cycle
- Non-cacheable instruction cache cycle into a cacheable data cache cycle
- Non-cacheable instruction cache cycle into a non-cacheable data cache cycle
- Cacheable data cache cycle into a cacheable instruction cache cycle
- Cacheable data cache cycle into a non-cacheable instruction cache cycle
- Non-cacheable data cache cycle into a cacheable instruction cache cycle
- Non-cacheable data cache cycle into a non-cacheable instruction cache cycle
- Write cycle (could be I/O) into a write cycle (could be I/O)
- Write cycle (could be I/O) into a cacheable instruction cache cycle
- Write cycle (could be I/O) into a non-cacheable instruction cache cycle
- Write cycle (could be I/O) into a cacheable data cache cycle
- Write cycle (could be I/O) into a non-cacheable data cache cycle

Pipelining is not supported for the following cycles:

- Non-cacheable instruction cache cycle into a non-cacheable instruction cache cycle
- Non-cacheable instruction cache cycle into a write cycle (could be I/O)
- Cacheable instruction cache cycle into a write cycle (could be I/O)
- Non-cacheable data cache cycle into a write cycle (could be I/O)
- Cacheable data cache cycle into a write cycle (could be I/O)
- Cacheable data cache cycle into a cacheable data cache cycle
- Cacheable data cache cycle into a non-cacheable data cache cycle
- Non-cacheable data cache cycle into a non-cacheable data cache cycle
- Non-cacheable data cache cycle into a cacheable data cache cycle

Pipelining Timing Diagrams

The timing diagrams in Figure 13 and Figure 14 illustrate pipelining.

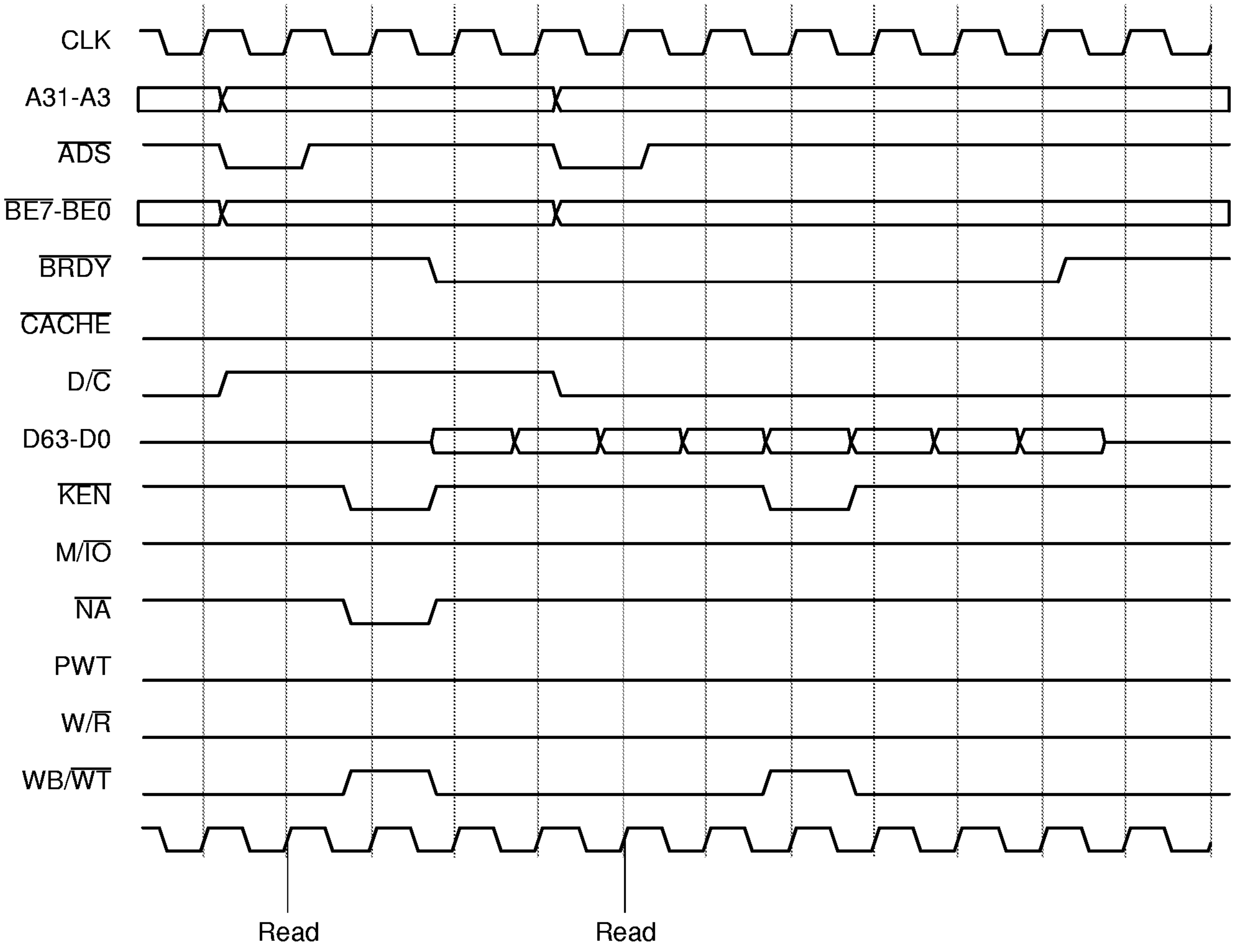


Figure 13. Pipelined Cacheable Data Cache Cycle into a Cacheable Instruction Cache Cycle

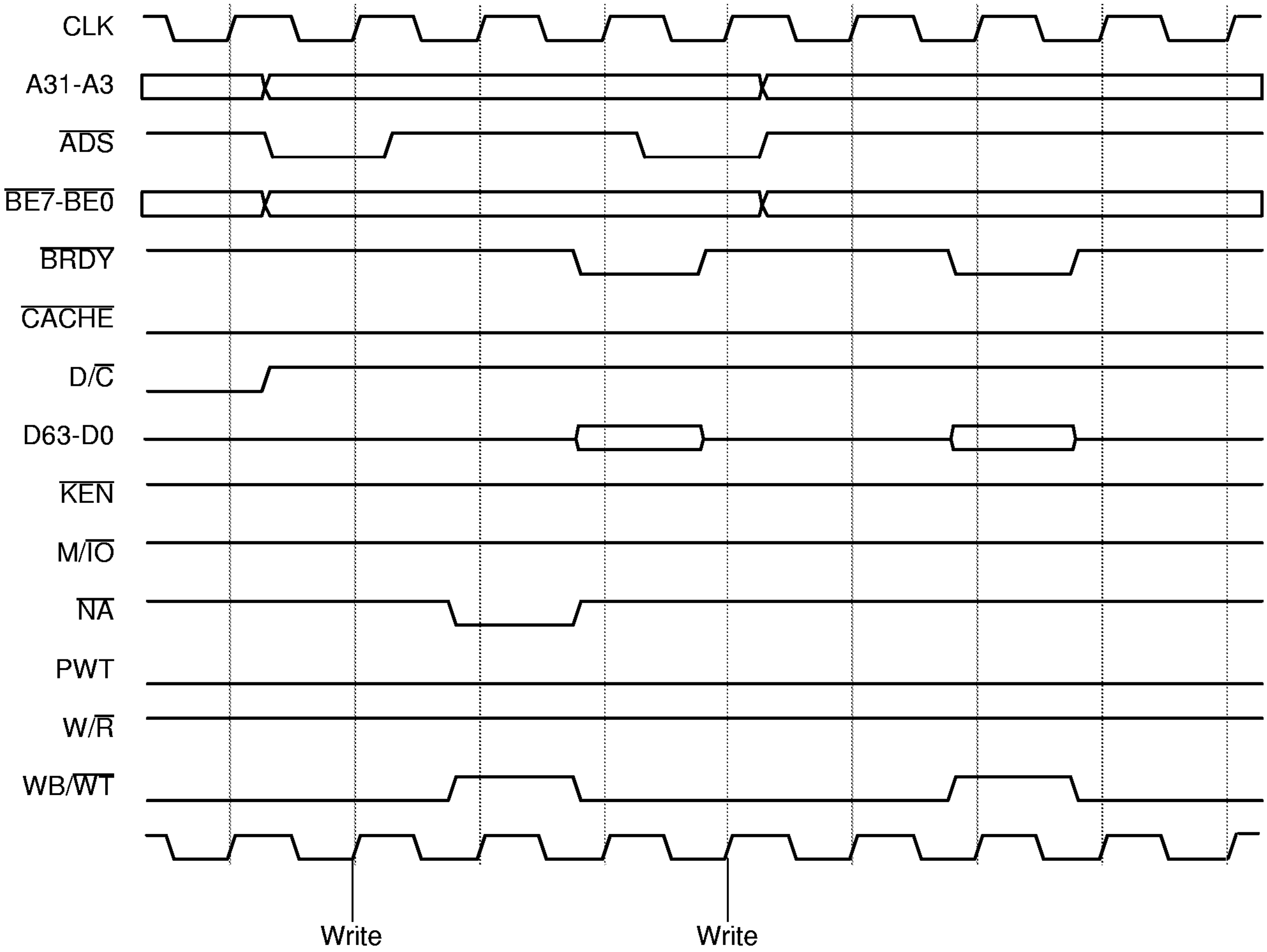


Figure 14. Pipelined Write Cycle (Could be I/O) into a Write Cycle (Could be I/O)

8.7 System Management Mode

System Management Mode (SMM) is a distinct processor mode—initiated by $\overline{\text{SMI}}$—that allows the system designer to add software-controlled features that operate transparently to the operating system and application programs, such as power management. I/O Restart and Halt Auto-Restart are also provided for transparent power management of I/O peripherals. The system designer may use the $\overline{\text{SMIACT}}$ signal to provide protection to the $\overline{\text{SMI}}$ handler code and CPU state information.

Processing System Management Interrupts

When the processor receives an $\overline{\text{SMI}}$, normal operation will be interrupted in the following manner:

1. $\overline{\text{SMIACT}}$ is asserted, informing the system that it must enable the SMRAM.
2. Once SMRAM is available, the processor saves its state beginning at 3FFFFh. The save area map is provided in Table 12 (given that the default SMIBASE is 30000h).
3. Once the normal execution state is saved in SMRAM, the processor enters SMM.
4. The processor will jump to SMRAM address 38000h to execute the $\overline{\text{SMI}}$ handler, which will perform any required system management.
5. When the Resume (RSM) instruction is received, the $\overline{\text{SMI}}$ handler restores the processor normal execution state from SRAM, negate the $\overline{\text{SMIACT}}$ signal, and resume execution.

System Management Interrupt

$\overline{\text{SMI}}$ is triggered on a clock falling edge. It is non-maskable and may be asserted asynchronously, but it will be recognized in the first cycle meeting set-up and hold times. To assure asynchronous recognition, $\overline{\text{SMI}}$ should be asserted for at least two clocks and negated for at least two clocks.

$\overline{\text{SMI}}$ interrupts occur on instruction boundaries. $\overline{\text{SMI}}$ is not affected by the IF bit in the EFLAGS register. The $\overline{\text{SMI}}$ signal will be masked internally when the $\overline{\text{SMI}}$ is recognized until the RSM instruction is executed. $\overline{\text{SMI}}$ has a higher priority than NMI. It is not masked during an NMI. (See Figure 41 on page 80.)

Table 12. SMM Save Area Map

Address	Contents	Address	Contents
FFFCh	CR0	FF74h	LDT Attribute
FFF8h	CR3	FF70h	LDT Base
FFF4h	EFLAGS	FF6Ch	LDT Limit
FFF0h	EIP	FF68h	GS Attributes
FFECh	EDI	FF64h	GS Base
FFE8h	ESI	FF60h	GS Limit
FFE4h	EBP	FF5Ch	FS Attributes
FFE0h	ESP	FF58h	FS Base
FFDCh	EBX	FF54h	FS Limit
FFD8h	EDX	FF50h	DS Attributes
FFD4h	ECX	FF4Ch	DS Base
FFD0h	EAX	FF48h	DS Limit
FFCCh	DR6	FF44h	SS Attributes
FFC8h	DR7	FF40h	SS Base
FFC4h	TR	FF3Ch	SS Limit
FFC0h	LDTR	FF38h	CS Attributes
FFBCh	GS	FF34h	CS Base
FFB8h	FS	FF30h	CS Limit
FFB4h	DS	FF2Ch	ES Attributes
FFB0h	SS	FF28h	ES Base
FFACh	CS	FF24h	ES Limit
FFA8h	ES	FF20h	Reserved
FFA4h	I/O Trap Word	FF1Ch	
FFA0h	Reserved	FF18h	
FF9Ch	I/O Trap EIP	FF14h	CR2
FF98h	Reserved	FF10h	CR4
FF94h		FF0Ch	I/O restart ESI
FF90h	IDT Base	FF08h	I/O restart ECX
FF8Ch	IDT Limit	FF04h	I/O restart EDI
FF88h	GDT Base	FF02h	Halt Restart
FF84h	GDT Limit	FF00h	I/O Trap Restart
FF80h	TR Attribute	FEFCh	SMM Rev ID
FF7Ch	TR Base	FEF8h	SMM Base Address
FF78h	TR Limit	FE00h–FEF4h	reserved

Initial State Upon Entering SMM

Table 13 shows the initial state of the processor upon entering SMM.

The default SMBASE value may be changed following reset to store the $\overline{\text{SMI}}$ handler code and CPU state information in a different region of memory. If the SMBASE value is changed, the next entry to the $\overline{\text{SMI}}$ handler routine will occur relative to the new SMBASE value.

Table 13. Initial State Upon Entering SMM

Register	Initial Contents			
	Selector	Base	Attributes	Limit
CS	3000h	0003_0000h	16-bit, expand-up	4 Gbytes
DS	0000h	0000_0000h	16-bit, expand-up	4 Gbytes
ES	0000h	0000_0000h	16-bit, expand-up	4 Gbytes
FS	0000h	0000_0000h	16-bit, expand-up	4 Gbytes
GS	0000h	0000_0000h	16-bit, expand-up	4 Gbytes
SS	0000h	0000_0000h	16-bit, expand-up	4 Gbytes
General-Purpose	Unmodified			
EFLAGS	0000_0002h			
EIP	0000_8000h			
CR0	Bits 0, 2, 3, 31 cleared (PE, EM, TS, PG). Others are unmodified.			
CR4	0000_0000h			
GDTR	Unmodified			
LDTR	Unmodified			
IDTR	Unmodified			
TR	Unmodified			
DR7	0000_0400h			
DR6	Undefined			

I/O Instruction Restart

The $\overline{\text{SMI}}$ handler may allow the RSM instruction to restart the interrupted I/O instruction by using the I/O instruction restart word. If the value contained by the I/O instruction restart word is 0FFh, the processor re-executes the I/O instruction trapped by $\overline{\text{SMI}}$.

The I/O instruction is not re-executed if the I/O restart word contains the value 000h. The value 000h is written in the I/O restart word when entering SMM. Processor operation is unpredictable if the I/O instruction restart word is written when the processor has not generated an $\overline{\text{SMI}}$ on an I/O instruction boundary. The $\overline{\text{SMI}}$ handler for the second request must not set the I/O instruction restart word if the system executes back-to-back $\overline{\text{SMI}}$ requests.

Halt Auto Restart

On entry to the $\overline{\text{SMI}}$ routine, the Halt Auto Restart word (FF02h) has the value 0001h if the processor was halted when the $\overline{\text{SMI}}$ occurred. Otherwise, it has a value of 0000h.

If the value is 0001h, the $\overline{\text{SMI}}$ routine may cause a return to the HALT instruction by returning without modifying the Halt Auto Restart word. It may cause a return to the instruction after the halt instruction by clearing the Halt Auto Restart word.

8.8 Am486® and AMD-K5 Processor Bus Differences

The AMD-K5 processor:

- Data bus is 64 bits, versus the Am486 processor's 32 bits
- Has eight byte-enables and eight data parity pins
- Does not support non-cacheable burst cycles
- Supports $\overline{\text{FLUSH}}$ as an edge-triggered input
- Supports a writeback cache protocol using MESI (The new $\overline{\text{CACHE}}$, $\overline{\text{HIT}}$, $\overline{\text{HITM}}$, WB/$\overline{\text{WT}}$, and INV pins are defined to support this protocol.)
- Maintains the state of the internal caches and FPU, while performing the reset function with the INIT pin
- Supports SMM with the input signal $\overline{\text{SMI}}$ and the output signal $\overline{\text{SMIACT}}$
- Does not allow invalidations every clock or while driving the address bus
- Supports parity checking on addresses and data
- Does not support dynamic bus sizing. This eliminates the need for the Am486 processor signals $\overline{\text{BS8}}$ and $\overline{\text{BS16}}$
- Includes the SCYC signal to indicate a split cycle during locked operations. A split cycle crosses a cache line boundary during an atomic operation due to a misaligned reference

- Adds the $\overline{\text{EWBE}}$ input to indicate an empty external write buffer (This supports strong store ordering between the processor and the external memory system. All writes to exclusive/modified lines are held until $\overline{\text{EWBE}}$ is asserted to indicate that no writes are pending in the external memory system.)
- On read-modify-write cycles, guarantees an idle cycle between consecutive locked accesses
- Implements non-cacheable code prefetches as eight bytes instead of 16 bytes (Each is treated as a single 8-byte access when non-cacheable.)
- Supports JTAG pins TCK, TDI, TDO, TMS, and $\overline{\text{TRST}}$
- Supports external breakpoints with the pins BP3–BP0
- Requires some writebacks and line fills to be run as burst cycles (With no $\overline{\text{BLAST}}$ pin, burst writebacks cannot be terminated in the middle of the burst.)
- Drives burst length information with the $\overline{\text{CACHE}}$ pin (This pin always indicates a fixed burst length of four 64-bit accesses. The corresponding pin is $\overline{\text{BLAST}}$ on the Am486 processor—where the burst is typically four 32-bit transfers—but can be longer with narrower width memories.)
- Supports simple Master/Slave modes through the pins $\overline{\text{FRCMC}}$ and $\overline{\text{IERR}}$
- Aborts a cycle if $\overline{\text{BOFF}}$ is asserted in the middle of the cycle (When $\overline{\text{BOFF}}$ is negated, the cycle restarts from the beginning. The Am486 processor restarts the cycle at the point it was aborted.)

8.9 P54C and AMD-K5 Processor Bus Differences

The AMD-K5 processor has two possible drive strengths, weak and strong. These strengths are equivalent to weak and strong on the Pentium processor. The recommended, default drive strength on the AMD-K5 processor is weak.

For detailed difference information, refer to Appendix A of the *AMD-K5 Processor Technical Reference Manual*, order# 18524, or the AMD-K5 Processor Application Note, "Comparison of the AMD-K5, Pentium, and 486 Processors," order# 20025.

9 Electrical Data

9.1 Power and Grounding

Power Connections

The AMD-K5 processor includes 53 V_{CC} and 53 V_{SS} pins for clean, on-chip power distribution at high frequencies. Power and ground connections must be made to all external V_{CC} and GND pins, respectively. All V_{CC} pins must be connected to the circuit board V_{CC} plane, and all V_{SS} pins must be connected to the circuit board GND plane. Table 14 on page 58 provides the DC characteristics of the processor.

Connection Recommendations

- Emphasize decoupling capacitance near the AMD-K5 processor
- Driving address and data buses into large capacitive loads at high frequencies can cause transient power surges
- Low inductance capacitors and circuit paths provide the best performance at high frequencies
- Inductance can be reduced by shortening circuit board paths as much as possible
- Capacitors specifically for PGA packaging are commercially available
- NC pins shall be unconnected
- Always connect unused inputs to an appropriate signal level
- Unused active Low inputs should be connected to V_{CC} through a pull-up resistor
- Pull-up resistors of 20 kΩ should be used
- Unused active High inputs should be connected to GND

9.2 Absolute Maximum Ratings

Case Temperature under Bias	–65°C to +110°C
Storage Temperature	–65°C to +150°C
Voltage on any pin (not to exceed 4.6 V)	–0.5 V to V_{CC} +0.5 V
CLK input (5-V tolerant)	–0.5 V to 6.5 V
Supply voltage	–0.5 V to +3.8 V

Note: *Stresses above those listed under Absolute Maximum Ratings may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to Absolute Maximum Ratings for extended periods may affect device reliability. All voltage levels are with respect to ground.*

9.3 Operating Ranges

See "Ordering Information" on page 5 for the standard products that are available for the AMD-K5 processor.

Commercial (C) Devices

T_{CASE}	0°C to +70°C
V_{CC}	3.525 V ± 2% (Refer to OPN)

Note: *Operating ranges define those limits between which the functionality of the device is guaranteed.*

Table 14. DC Characteristics over Commercial Operating Ranges

Symbol	Parameter Description	Advance Info		Comments
		Min	Max	
V_{IL}	Input Low Voltage	–0.3 V	+0.8 V	
V_{IH}	Input High Voltage	2.0 V	V_{CC}+0.3 V	
V_{OL}	Output Low Voltage		0.4 V	I_{OL} = 4-mA load
V_{OH}	Output High Voltage	2.4 V		I_{OH} = 1-mA load
I_{CC}	Power Supply Current—Model 0		44.0 mA/MHz	V_{CC} = 3.6 V Note 1
	Power Supply Current—Models 1 and 2		39.0 mA/MHz	V_{CC} = 3.6 V Note 6
I_{LI}	Input Leakage Current		±15 µA	Note 2
I_{LO}	Output Leakage Current		±15 µA	Note 2
I_{IL}	Input Leakage Current Bias with Pull-up (Low)		400 µA	Note 3
I_{IH}	Input Leakage Current Bias with Pull-up (High)		200 µA	Note 4
C_{IN}	Input Capacitance		15 pF	Note 5
C_{OUT}	Output Capacitance		20 pF	Note 5
C_{OUT}	I/O Capacitance		25 pF	Note 5
C_{CLK}	CLK Capacitance		15 pF	Note 5
C_{TIN}	Test Input Capacitance		15 pF	Note 5
C_{TOUT}	Test Output Capacitance		20 pF	Note 5
C_{TCK}	TCK Capacitance		15 pF	Note 5

Notes:

1. *Typical supply current for model 0: 36 mA/MHz (2700 mA at PR75, 3240 mA at PR90, and 3600 mA at PR100).*
2. *This parameter is for inputs or I/O without an internal pull-up resistor and $0 \le V_{IN} \le V_{CC}$.*
3. *This parameter is for inputs with pull-ups and V_{IL} = 0.40 V.*
4. *This parameter is for inputs with pull-downs and V_{IH} = 2.4 V.*
5. *This parameter is determined by design.*
6. *Typical supply current for models 1 and 2: 30 mA/MHz (2700 mA at PR120, 3000 mA at PR133, and 3500 mA at PR166).*

10 Switching Characteristics

The AMD-K5 processor commercial switching characteristics, provided in Table 15 through Table 26 on page 67, are measured at the voltage levels indicated by Figure 16 on page 68. They are measured relative to the rising edge of the CLK signal, as defined by Figure 16 through Figure 23. Output delays are specified as a function of minimum and maximum limits, with minimum delay times provided to external circuitry as hold times. A synchronous input signal must be stable for correct AMD-K5 processor operation during sampling.

10.1 66-MHz Bus Operation

Table 15. CLK Switching Characteristics for 66-MHz Bus Operation

Symbol	Parameter Description	Advance Info		Figure	Comments
		Min	Max		
	Frequency	33.3 MHz	66.6 MHz		
t_1	CLK Period	15 ns	30.0 ns	16	
t_{1a}	CLK Period Stability		± 250 ps		Note 1
t_2	CLK High Time	4.0 ns		16	@ 2.0 V, Note 1
t_3	CLK Low Time	4.0 ns		16	@ 0.8 V, Note 1
t_4	CLK Fall Time	0.15 ns	1.5 ns	16	2.0–0.8 V, Note 1
t_5	CLK Rise Time	0.15 ns	1.5 ns	16	0.8–2.0 V, Note 1

Notes:

1. Not 100% tested; determined by design characterization.

Table 16. Delay Timing for 66-MHz Bus Operation

Symbol	Parameter Description	Advance Info		Figure	Comments
		Min	Max		
t_{6a}	$\overline{\text{ADSC}}$, PWT, PCD, $\overline{\text{CACHE}}$, SCYC Valid Delay	1.0 ns	7.0 ns	17	
t_{6b}	AP Valid Delay	1.0 ns	8.5 ns	17	
t_{6c}	A31–A17 Valid Delay	0.6 ns	6.3 ns	17	
t_{6d}	A16–A3 Valid Delay	0.5 ns	6.3 ns	15	
t_{6e}	$\overline{\text{ADS}}$ Valid Delay	1.0 ns	6.0 ns	17	
t_{6f}	$\overline{\text{BE7}}$–$\overline{\text{BE0}}$ Valid Delay	0.9 ns	7.0 ns	15	
t_{6g}	$\overline{\text{LOCK}}$ Valid Delay	0.9 ns	7.0 ns	15	
t_{6h}	M/$\overline{\text{IO}}$ Valid Delay	0.8 ns	5.9 ns	15	
t_{6i}	D/$\overline{\text{C}}$, W/$\overline{\text{R}}$ Valid Delay	0.8 ns	7.0 ns	15	
t_7	$\overline{\text{ADS}}$, $\overline{\text{ADSC}}$, AP, A31–A3, $\overline{\text{BE7}}$–$\overline{\text{BE0}}$, $\overline{\text{CACHE}}$, D/$\overline{\text{C}}$, $\overline{\text{LOCK}}$, M/$\overline{\text{IO}}$, PWT, PCD, SCYC, W/$\overline{\text{R}}$ Float Delay		10.0 ns	19	
t_{8a}	$\overline{\text{APCHK}}$, $\overline{\text{IERR}}$, $\overline{\text{FERR}}$ Valid Delay	1.0 ns	8.3 ns	17	
t_{8b}	$\overline{\text{PCHK}}$ Valid Delay	1.0 ns	7.0 ns	17	
t_{9a}	BREQ, HLDA Valid Delay	1.0 ns	8.0 ns	17	
t_{9b}	$\overline{\text{SMIACT}}$ Valid Delay	1.0 ns	7.3 ns	17	
t_{10a}	$\overline{\text{HIT}}$ Valid Delay	1.0 ns	6.8 ns	17	
t_{10b}	$\overline{\text{HITM}}$ Valid Delay	0.7 ns	6.0 ns	17	
t_{11}	PRDY Valid Delay	1.0 ns	8.0 ns	17	
t_{12}	D63–D0, DP7–DP0 Write Data Valid Delay	1.3 ns	7.5 ns	17	
t_{13}	D63–D0, DP7–DP0 Write Data Float Delay		10.0 ns	19	

Table 17. Switching Characteristics for 66-MHz Bus Operation

Symbol	Parameter Description	Advance Info		Figure	Comments
		Min	Max		
t_{14}	A31–A5 Setup Time	6.0 ns		18	
t_{15}	A31–A5 Hold Time	1.0 ns		18	
t_{16a}	INV, AP Setup Time	5.0 ns		18	
t_{16b}	$\overline{\text{EADS}}$ Setup Time	5.0 ns		18	
t_{17}	$\overline{\text{EADS}}$, INV, AP Hold Time	1.0 ns		18	
t_{18a}	$\overline{\text{KEN}}$ Setup Time	5.0 ns		18	
t_{18b}	WB/$\overline{\text{WT}}$, $\overline{\text{NA}}$ Setup Time	4.5 ns		18	
t_{19}	$\overline{\text{KEN}}$, WB/$\overline{\text{WT}}$, $\overline{\text{NA}}$ Hold Time	1.0 ns		18	
t_{20}	$\overline{\text{BRDY}}$, $\overline{\text{BRDYC}}$ Setup Time	5.0 ns		18	
t_{21}	$\overline{\text{BRDY}}$, $\overline{\text{BRDYC}}$ Hold Time	1.0 ns		18	
t_{22}	AHOLD, $\overline{\text{BOFF}}$ Setup Time	5.5 ns		18	
t_{23}	AHOLD, $\overline{\text{BOFF}}$ Hold Time	1.0 ns		18	
t_{24}	$\overline{\text{BUSCHK}}$, $\overline{\text{EWBE}}$, HOLD Setup Time	5.0 ns		18	
t_{24a}	$\overline{\text{PEN}}$ Setup Time	4.8 ns		16	
t_{25a}	$\overline{\text{BUSCHK}}$, $\overline{\text{EWBE}}$, $\overline{\text{PEN}}$ Hold Time	1.0 ns		18	
t_{25b}	HOLD Hold Time	1.5 ns		18	
t_{26}	$\overline{\text{A20M}}$, INTR, $\overline{\text{STPCLK}}$ Setup Time	5.0 ns		18	
t_{27}	$\overline{\text{A20M}}$, INTR, $\overline{\text{STPCLK}}$ Hold Time	1.0 ns		18	
t_{28}	INIT, $\overline{\text{FLUSH}}$, NMI, $\overline{\text{SMI}}$, $\overline{\text{IGNNE}}$ Setup Time	5.0 ns		18	
t_{29}	INIT, $\overline{\text{FLUSH}}$, NMI, $\overline{\text{SMI}}$, $\overline{\text{IGNNE}}$ Hold Time	1.0 ns		18	
t_{30}	INIT, $\overline{\text{FLUSH}}$, NMI, $\overline{\text{SMI}}$, $\overline{\text{IGNNE}}$ Pulse Width	2 clocks		18	Asynchronous
t_{31}	R/$\overline{\text{S}}$ Setup Time	5.0 ns		18	
t_{32}	R/$\overline{\text{S}}$ Hold Time	1.0 ns		18	
t_{33}	R/$\overline{\text{S}}$ Pulse Width	2 clocks		18	Asynchronous
t_{34}	D63–D0, DP7–DP0 Read Data Setup Time	2.8 ns		18	
t_{35}	D63–D0, DP7–DP0 Read Data Hold Time	1.5 ns		18	

10.2 60-MHz Bus Operation

Table 18. CLK Switching Characteristics for 60-MHz Bus Operation

Symbol	Parameter Description	Advance Info		Figure	Comments
		Min	Max		
	Frequency	30 MHz	60 MHz		
t_1	CLK Period	16.67 ns	33.33 ns	16	
t_{1a}	CLK Period Stability		± 250 ps		Note 1
t_2	CLK High Time	4.0 ns		16	@ 2.0 V, Note 1
t_3	CLK Low Time	4.0 ns		16	@ 0.8 V, Note 1
t_4	CLK Fall Time	0.15 ns	1.5 ns	16	2.0–0.8 V, Note 1
t_5	CLK Rise Time	0.15 ns	1.5 ns	16	0.8–2.0 V, Note 1

Notes:

1. Not 100% tested; determined by design characterization.

Table 19. Delay Timing for 60-MHz Bus Operation

Symbol	Parameter Description	Advance Info		Figure	Comments
		Min	Max		
t_{6a}	$\overline{\text{ADSC}}$, $\overline{\text{BE7}}$–$\overline{\text{BE0}}$, D/$\overline{\text{C}}$, PWT, PCD, W/$\overline{\text{R}}$, $\overline{\text{CACHE}}$, SCYC Valid Delay	1.0 ns	7.0 ns	17	
t_{6b}	AP Valid Delay	1.0 ns	8.5 ns	17	
t_{6c}	A31–A3, $\overline{\text{LOCK}}$ Valid Delay	1.1 ns	7.0 ns	17	
t_{6d}	$\overline{\text{ADS}}$, M/$\overline{\text{IO}}$ Valid Delay	1.0 ns	7.0 ns	17	
t_7	$\overline{\text{ADS}}$, $\overline{\text{ADSC}}$, AP, A31-A3, $\overline{\text{BE7}}$–$\overline{\text{BE0}}$, $\overline{\text{CACHE}}$, D/$\overline{\text{C}}$, $\overline{\text{LOCK}}$, M/$\overline{\text{IO}}$, PWT, PCD, SCYC, W/$\overline{\text{R}}$ Float Delay		10.0 ns	19	
t_{8a}	$\overline{\text{APCHK}}$, $\overline{\text{IERR}}$, $\overline{\text{FERR}}$ Valid Delay	1.0 ns	8.3 ns	17	
t_{8b}	$\overline{\text{PCHK}}$ Valid Delay	1.0 ns	7.0 ns	17	
t_{9a}	BREQ, HLDA Valid Delay	1.0 ns	8.0 ns	17	
t_{9b}	$\overline{\text{SMIACT}}$ Valid Delay	1.0 ns	7.6 ns	17	
t_{10a}	$\overline{\text{HIT}}$ Valid Delay	1.0 ns	8.0 ns	17	
t_{10b}	$\overline{\text{HITM}}$ Valid Delay	1.1 ns	6.0 ns	17	
t_{11}	PRDY Valid Delay	1.0 ns	8.0 ns	17	
t_{12}	D63–D0, DP7–DP0 Write Data Valid Delay	1.3 ns	7.5 ns	17	
t_{13}	D63–D0, DP7–DP0 Write Data Float Delay		10.0 ns	19	

Table 20. Switching Characteristics for 60-MHz Bus Operation

Symbol	Parameter Description	Advance Info		Figure	Comments
		Min	Max		
t_{14}	A31–A5 Setup Time	6.0 ns		18	
t_{15}	A31–A5 Hold Time	1.0 ns		18	
t_{16a}	INV, AP Setup Time	5.0 ns		18	
t_{16b}	$\overline{\text{EADS}}$ Setup Time	5.5 ns		18	
t_{17}	$\overline{\text{EADS}}$, INV, AP Hold Time	1.0 ns		18	
t_{18a}	$\overline{\text{KEN}}$ Setup Time	5.0 ns		18	
t_{18b}	WB/$\overline{\text{WT}}$, $\overline{\text{NA}}$ Setup Time	4.5 ns		18	
t_{19}	$\overline{\text{KEN}}$, WB/$\overline{\text{WT}}$, $\overline{\text{NA}}$ Hold Time	1.0 ns		18	
t_{20}	$\overline{\text{BRDY}}$, $\overline{\text{BRDYC}}$ Setup Time	5.0 ns		18	
t_{21}	$\overline{\text{BRDY}}$, $\overline{\text{BRDYC}}$ Hold Time	1.0 ns		18	
t_{22}	AHOLD, $\overline{\text{BOFF}}$ Setup Time	5.5 ns		18	
t_{23}	AHOLD, $\overline{\text{BOFF}}$ Hold Time	1.0 ns		18	
t_{24}	$\overline{\text{BUSCHK}}$, $\overline{\text{EWBE}}$, HOLD, $\overline{\text{PEN}}$ Setup Time	5.0 ns		18	
t_{25a}	$\overline{\text{BUSCHK}}$, $\overline{\text{EWBE}}$, $\overline{\text{PEN}}$ Hold Time	1.0 ns		18	
t_{25b}	HOLD Hold Time	1.5 ns		18	
t_{26}	$\overline{\text{A20M}}$, INTR, $\overline{\text{STPCLK}}$ Setup Time	5.0 ns		18	
t_{27}	$\overline{\text{A20M}}$, INTR, $\overline{\text{STPCLK}}$ Hold Time	1.0 ns		18	
t_{28}	INIT, $\overline{\text{FLUSH}}$, NMI, $\overline{\text{SMI}}$, $\overline{\text{IGNNE}}$ Setup Time	5.0 ns		18	
t_{29}	INIT, $\overline{\text{FLUSH}}$, NMI, $\overline{\text{SMI}}$, $\overline{\text{IGNNE}}$ Hold Time	1.0 ns		18	
t_{30}	INIT, $\overline{\text{FLUSH}}$, NMI, $\overline{\text{SMI}}$, $\overline{\text{IGNNE}}$ Pulse Width	2 clocks		18	Asynchronous
t_{31}	R/$\overline{\text{S}}$ Setup Time	5.0 ns		18	
t_{32}	R/$\overline{\text{S}}$ Hold Time	1.0 ns		18	
t_{33}	R/$\overline{\text{S}}$ Pulse Width	2 clocks		18	Asynchronous
t_{34}	D63–D0, DP7–DP0 Read Data Setup Time	3.0 ns		18	
t_{35}	D63–D0, DP7–DP0 Read Data Hold Time	2.0 ns		18	

10.3 50-MHz Bus Operation

Table 21. CLK Switching Characteristics for 50-MHz Bus Operation

Symbol	Parameter Description	Advance Info		Figure	Comments
		Min	Max		
	Frequency	25 MHz	50 MHz		
t_1	CLK Period	20.0 ns	40.0 ns	16	
t_{1a}	CLK Period Stability		± 250 ps		Note 1
t_2	CLK High Time	4.0 ns		16	@ 2.0 V, Note 1
t_3	CLK Low Time	4.0 ns		16	@ 0.8 V, Note 1
t_4	CLK Fall Time	0.15 ns	1.5 ns	16	2.0–0.8 V, Note 1
t_5	CLK Rise Time	0.15 ns	1.5 ns	16	0.8–2.0 V, Note 1

Notes:

1. Not 100% tested; determined by design characterization.

Table 22. Delay Timing for 50-MHz Bus Operation

Symbol	Parameter Description	Advance Info		Figure	Comments
		Min	Max		
t_{6a}	$\overline{\text{ADSC}}$, $\overline{\text{BE7}}$–$\overline{\text{BE0}}$, D/$\overline{\text{C}}$, PWT, PCD, W/$\overline{\text{R}}$, $\overline{\text{CACHE}}$, SCYC Valid Delay	1.0 ns	7.0 ns	17	
t_{6b}	AP Valid Delay	1.0 ns	8.5 ns	17	
t_{6c}	A31–A3, $\overline{\text{LOCK}}$ Valid Delay	1.1 ns	7.0 ns	17	
t_{6d}	$\overline{\text{ADS}}$, M/$\overline{\text{IO}}$ Valid Delay	1.0 ns	7.0 ns	17	
t_7	$\overline{\text{ADS}}$, $\overline{\text{ADSC}}$, AP, A31–A3, $\overline{\text{BE7}}$–$\overline{\text{BE0}}$, $\overline{\text{CACHE}}$, D/$\overline{\text{C}}$, $\overline{\text{LOCK}}$, M/$\overline{\text{IO}}$, PCD, PWT, SCYC, W/$\overline{\text{R}}$ Float Delay		10.0 ns	19	
t_{8a}	$\overline{\text{APCHK}}$, $\overline{\text{IERR}}$, $\overline{\text{FERR}}$ Valid Delay	1.0 ns	8.3 ns	17	
t_{8b}	$\overline{\text{PCHK}}$ Valid Delay	1.0 ns	8.3 ns	17	
t_{9a}	BREQ, HLDA Valid Delay	1.0 ns	8.0 ns	17	
t_{9b}	$\overline{\text{SMIACT}}$ Valid Delay	1.0 ns	8.0 ns	17	
t_{10a}	$\overline{\text{HIT}}$ Valid Delay	1.0 ns	8.0 ns	17	
t_{10b}	$\overline{\text{HITM}}$ Valid Delay	1.1 ns	6.0 ns	17	
t_{11}	PRDY Valid Delay	1.0 ns	8.0 ns	17	
t_{12}	D63–D0, DP7–DP0 Write Data Valid Delay	1.3 ns	8.5 ns	17	
t_{13}	D63–D0, DP7–DP0 Write Data Float Delay		10.0 ns	19	

Table 23. Switching Characteristics for 50-MHz Bus Operation

Symbol	Parameter Description	Advance Info		Figure	Comments
		Min	Max		
t_{14}	A31–A5 Setup Time	6.5 ns		18	
t_{15}	A31–A5 Hold Time	1.0 ns		18	
t_{16a}	INV, AP Setup Time	5.0 ns		18	
t_{16b}	$\overline{\text{EADS}}$ Setup Time	6.0 ns		18	
t_{17}	$\overline{\text{EADS}}$, INV, AP Hold Time	1.0 ns		18	
t_{18a}	$\overline{\text{KEN}}$ Setup Time	5.0 ns		18	
t_{18b}	WB/$\overline{\text{WT}}$, $\overline{\text{NA}}$ Setup Time	4.5 ns		18	
t_{19}	$\overline{\text{KEN}}$, WB/$\overline{\text{WT}}$, $\overline{\text{NA}}$ Hold Time	1.0 ns		18	
t_{20}	$\overline{\text{BRDY}}$, $\overline{\text{BRDYC}}$ Setup Time	5.0 ns		18	
t_{21}	$\overline{\text{BRDY}}$, $\overline{\text{BRDYC}}$ Hold Time	1.0 ns		18	
t_{22a}	$\overline{\text{BOFF}}$ Setup Time	5.5 ns		18	
t_{22b}	AHOLD Setup Time	6.0 ns		18	
t_{23}	AHOLD, $\overline{\text{BOFF}}$ Hold Time	1.0 ns		18	
t_{24}	$\overline{\text{BUSCHK}}$, $\overline{\text{EWBE}}$, HOLD, $\overline{\text{PEN}}$ Setup Time	5.0 ns		18	
t_{25a}	$\overline{\text{BUSCHK}}$, $\overline{\text{EWBE}}$, $\overline{\text{PEN}}$ Hold Time	1.0 ns		18	
t_{25b}	HOLD Hold Time	1.5 ns		18	
t_{26}	$\overline{\text{A20M}}$, INTR, $\overline{\text{STPCLK}}$ Setup Time	5.0 ns		18	
t_{27}	$\overline{\text{A20M}}$, INTR, $\overline{\text{STPCLK}}$ Hold Time	1.0 ns		18	
t_{28}	INIT, $\overline{\text{FLUSH}}$, NMI, $\overline{\text{SMI}}$, $\overline{\text{IGNNE}}$ Setup Time	5.0 ns		18	
t_{29}	INIT, $\overline{\text{FLUSH}}$, NMI, $\overline{\text{SMI}}$, $\overline{\text{IGNNE}}$ Hold Time	1.0 ns		18	
t_{30}	INIT, $\overline{\text{FLUSH}}$, NMI, $\overline{\text{SMI}}$, $\overline{\text{IGNNE}}$ Pulse Width	2 clocks		18	Asynchronous
t_{31}	R/$\overline{\text{S}}$ Setup Time	5.0 ns		18	
t_{32}	R/$\overline{\text{S}}$ Hold Time	1.0 ns		18	
t_{33}	R/$\overline{\text{S}}$ Pulse Width	2 clocks		18	Asynchronous
t_{34}	D63–D0, DP7–DP0 Read Data Setup Time	3.8 ns		18	
t_{35}	D63–D0, DP7–DP0 Read Data Hold Time	2.0 ns		18	

10.4 RESET, TCK, $\overline{\text{TRST}}$, and Test Signal Timing

Table 24. RESET Configuration Signal

Symbol	Parameter Description	Advance Info		Figure	Comments
		Min	Max		
t_{36}	RESET Setup Time	5.0 ns		20	
t_{37}	RESET Hold Time	1.0 ns		20	
t_{38}	RESET Pulse Width, V_{CC} and CLK stable	15 clocks		20	
t_{39}	RESET active after V_{CC} and CLK stable	1.0 ms		20	
t_{40}	INIT, $\overline{\text{FLUSH}}$, $\overline{\text{FRCMC}}$ Setup Time	5.0 ns		20	
t_{41}	INIT, $\overline{\text{FLUSH}}$, $\overline{\text{FRCMC}}$ Hold Time	1.0 ns		20	
t_{42a}	INIT, $\overline{\text{FLUSH}}$, $\overline{\text{FRCMC}}$ Setup Time	2 clocks		20	Asynchronous, Note 1
t_{42b}	INIT, $\overline{\text{FLUSH}}$, $\overline{\text{FRCMC}}$, $\overline{\text{BRDYC}}$, $\overline{\text{BUSCHK}}$ Hold Time	2 clocks		20	Asynchronous, Note 1
t_{42c}	$\overline{\text{BRDYC}}$, $\overline{\text{BUSCHK}}$ Setup Time	3 clocks		20	Note 1
t_{42d}	$\overline{\text{BRDYC}}$ Hold Time, RESET driven synchronously	1.0 ns		20	Note 1
t_{43a}	BF, BF0, BF1 Setup Time	1.0 ms		20	Note 1
t_{43b}	BF, BF0, BF1 Hold Time	2 clocks		20	Note 1

Notes:

1. *These are measured to RESET falling edge.*

Table 25. TCK Waveform and $\overline{\text{TRST}}$ Timing at 16 MHz

Symbol	Parameter Description	Advance Info		Figure	Comments
		Min	Max		
t_{44}	TCK Frequency		16 MHz		1X Clock
t_{45}	TCK Period	62.5 ns		21	Note 1
t_{46}	TCK High Time	25.0 ns		21	at 2.0 V, Note 3
t_{47}	TCK Low Time	25.0 ns		21	at 0.8 V, Note 3
t_{48}	TCK Fall Time		5.0 ns	21	Notes 2, 3
t_{49}	TCK Rise Time		5.0 ns	21	Notes 2, 3
t_{50}	$\overline{\text{TRST}}$ Pulse Width	40.0 ns		22	Asynchronous

Notes:

1. *TCK period is ≥ CLK period.*
2. *Rise/Fall times are measured between 0.8 V and 2.0 V. Rise/Fall times can be relaxed by 1 ns per 10-ns increase in TCK period.*
3. *Not 100% tested; determined by design characterization.*

Table 26. Test Signal Timing at 16 MHz

Symbol	Parameter Description	Advance Info		Figure	Notes
		Min	Max		
t_{51}	TDI, TMS Setup Time	5.0 ns		23	Note 2
t_{52}	TDI, TMS Hold Time	13.0 ns		23	Note 2
t_{53}	TDO Valid Delay	3.0 ns	20.0 ns	23	Note 1
t_{54}	TDO Float Delay		25.0 ns	23	Note 1
t_{55}	All Outputs (Non-Test) Valid Delay	3.0 ns	20.0 ns	23	Note 1
t_{56}	All Outputs (Non-Test) Float Delay		25.0 ns	23	Note 1
t_{57}	All Inputs (Non-Test) Setup Time	5.0 ns		23	Note 2
t_{58}	All Inputs (Non-Test) Hold Time	13.0 ns		23	Note 2

Notes:

1. *Parameter is measured from the TCK falling edge.*
2. *Parameter is measured from the TCK rising edge.*

TCK period is ≥ CLK period.

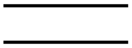
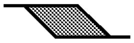
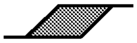
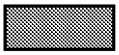
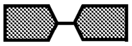
WAVEFORM	INPUTS	OUTPUTS
	Must be steady	Will be steady
	May change from High to Low	Will be changing from High to Low
	May change from Low to High	Will be changing from Low to High
	Don't care, any change permitted	Changing, State Unknown
	(Does not apply)	Center line is in a high impedance "Off" state

Figure 15. Diagrams Key

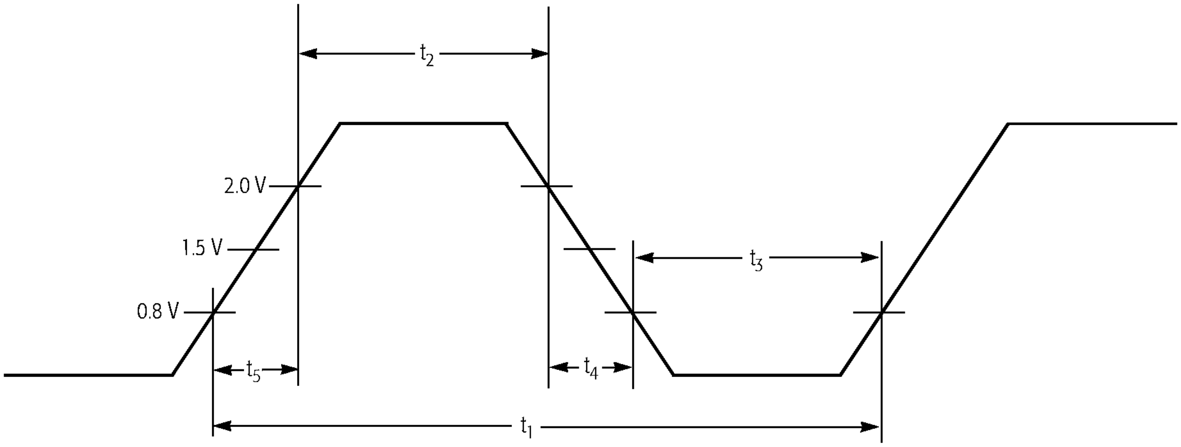
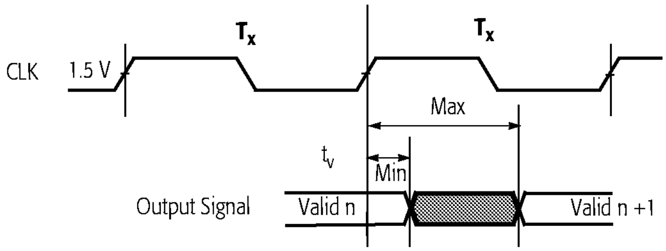
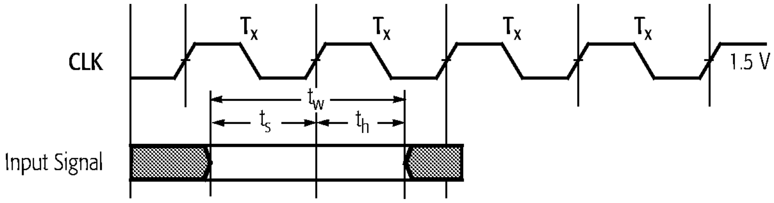


Figure 16. CLK Waveform



v = 6a, 6b, 6c, 6d, 6e, 6f, 6g, 6h, 6i, 8a, 8b, 9a, 9b, 10a, 10b, 11, 12

Figure 17. Output Valid Delay Timing

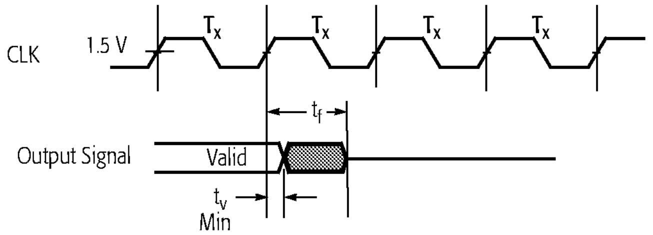


s = 14, 16a, 16b, 18a, 18b, 20, 22, 22a, 22b, 24, 24a, 26, 28, 31, 34

h = 15, 17, 19, 21, 23, 25a, 25b, 27, 29, 32, 35

w = 30, 33

Figure 18. Input Setup and Hold Timing



v = 6a, 6b, 6c, 6d, 6e, 6f, 6g, 6h, 6i, 12

f = 7, 13

Figure 19. Maximum Float Delay Timing

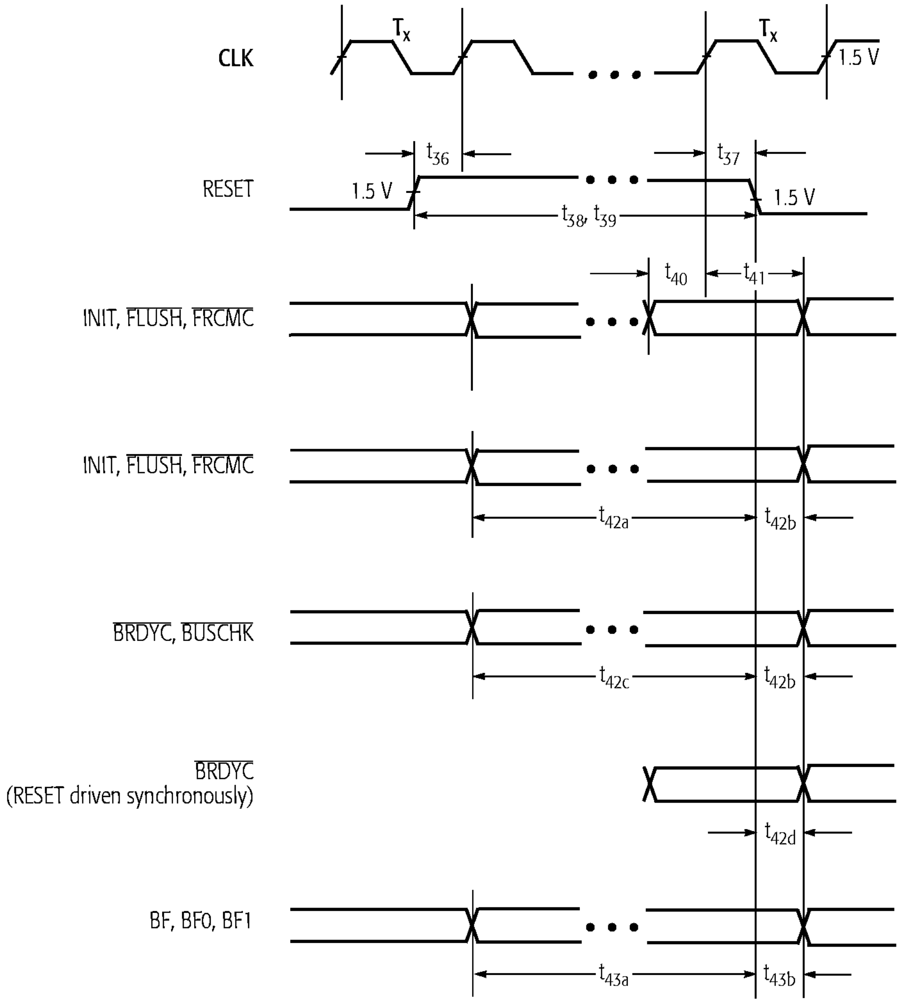

CLK
T_x
T_x
1.5 V
RESET
t_{36}
t_{37}
1.5 V
1.5 V
t_{38}, t_{39}
t_{40}
t_{41}
INIT, $\overline{\text{FLUSH}}$, $\overline{\text{FRCMC}}$
INIT, $\overline{\text{FLUSH}}$, $\overline{\text{FRCMC}}$
t_{42a}
t_{42b}
$\overline{\text{BRDYC}}$, $\overline{\text{BUSCHK}}$
t_{42c}
t_{42b}
$\overline{\text{BRDYC}}$
(RESET driven synchronously)
t_{42d}
BF, BF0, BF1
t_{43a}
t_{43b}

Figure 20. Reset and Configuration Timing

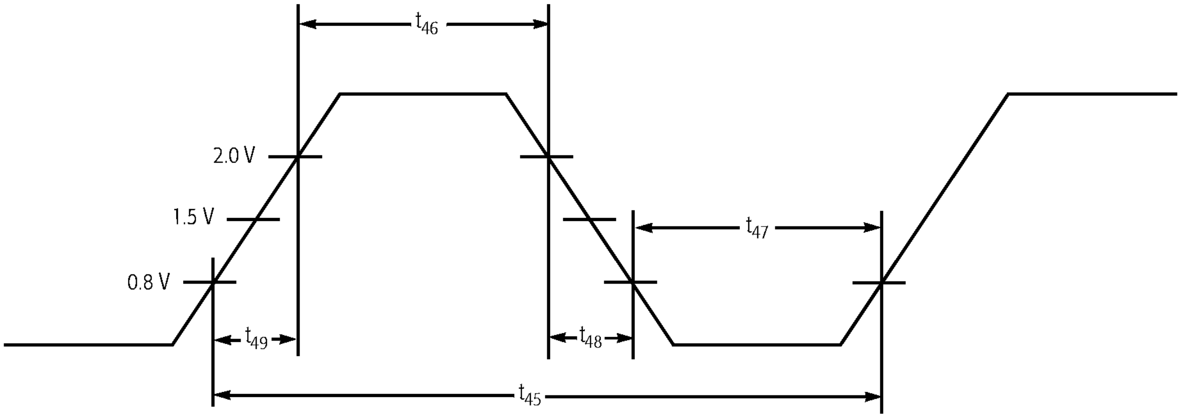


Figure 21. TCK Waveform

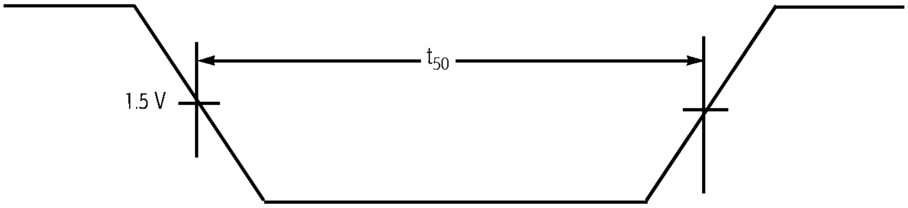


Figure 22. $\overline{\text{TRST}}$ Timing

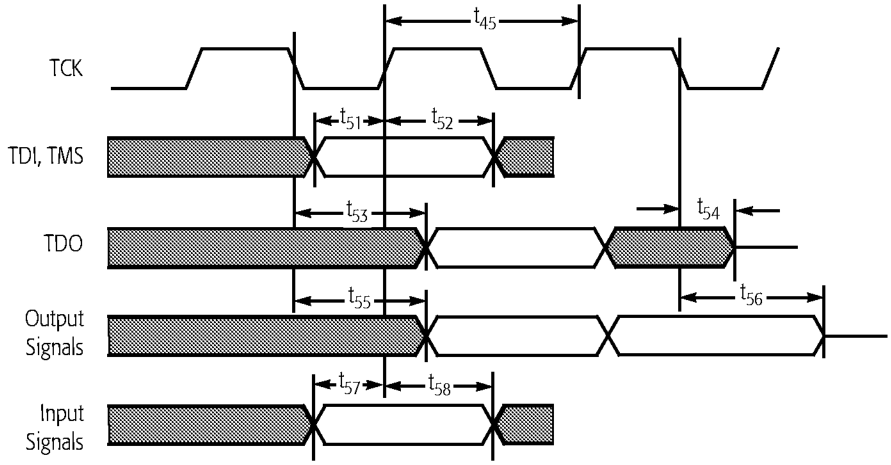


Figure 23. Test Signal Timing Diagram

11 Timing Diagrams

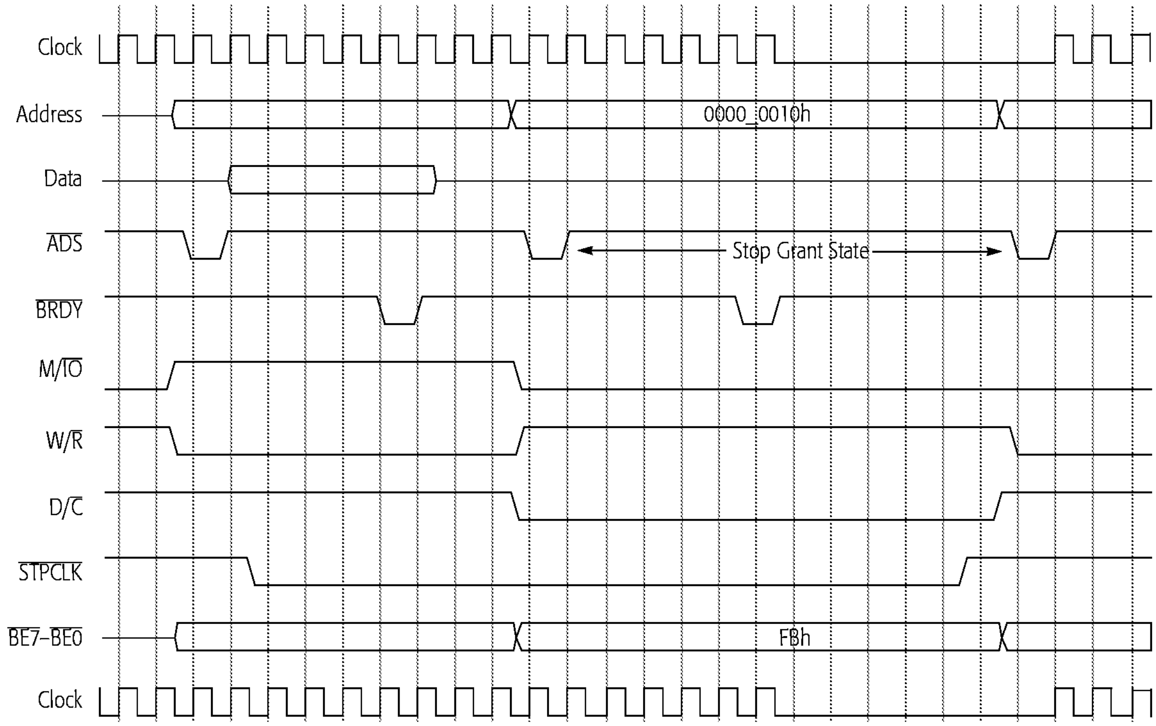


Figure 24. STPCLK Timing (Stop Grant state)

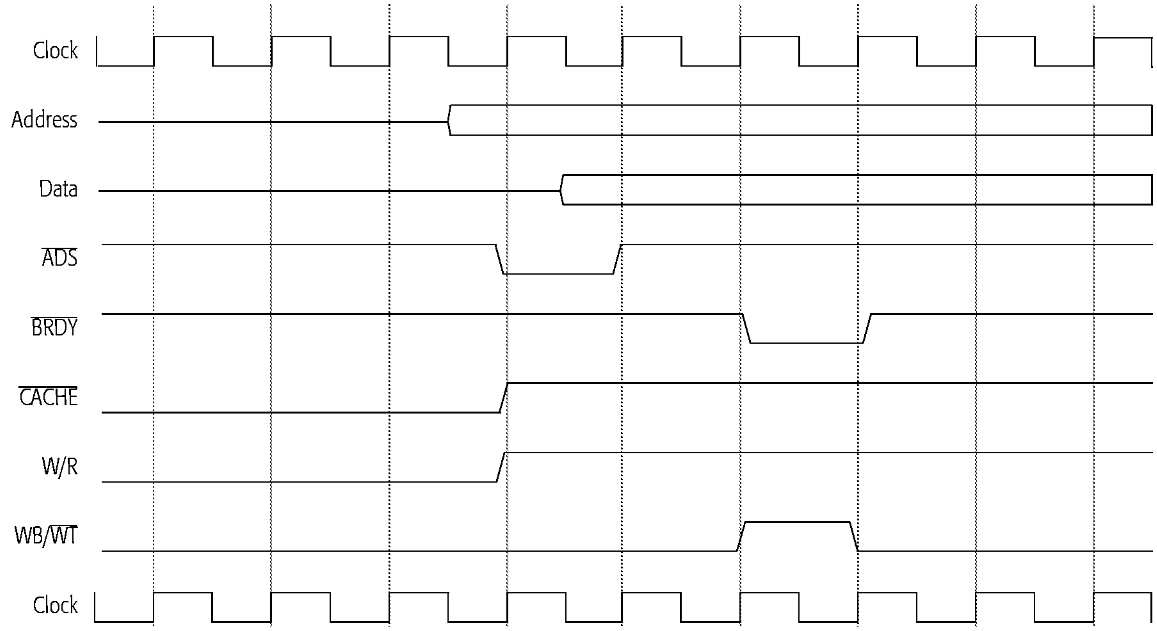


Figure 25. Transition L1 Shared Line to Exclusive

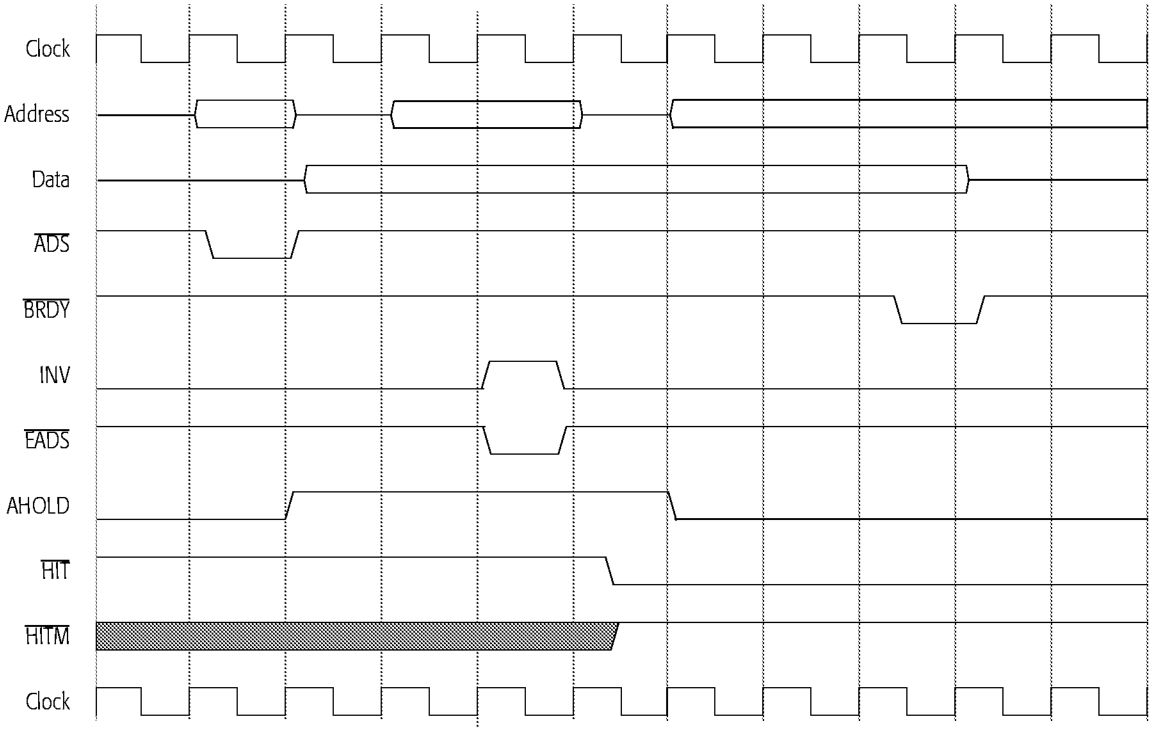


Figure 26. Invalidation to Non-Modified L1 Cache Line

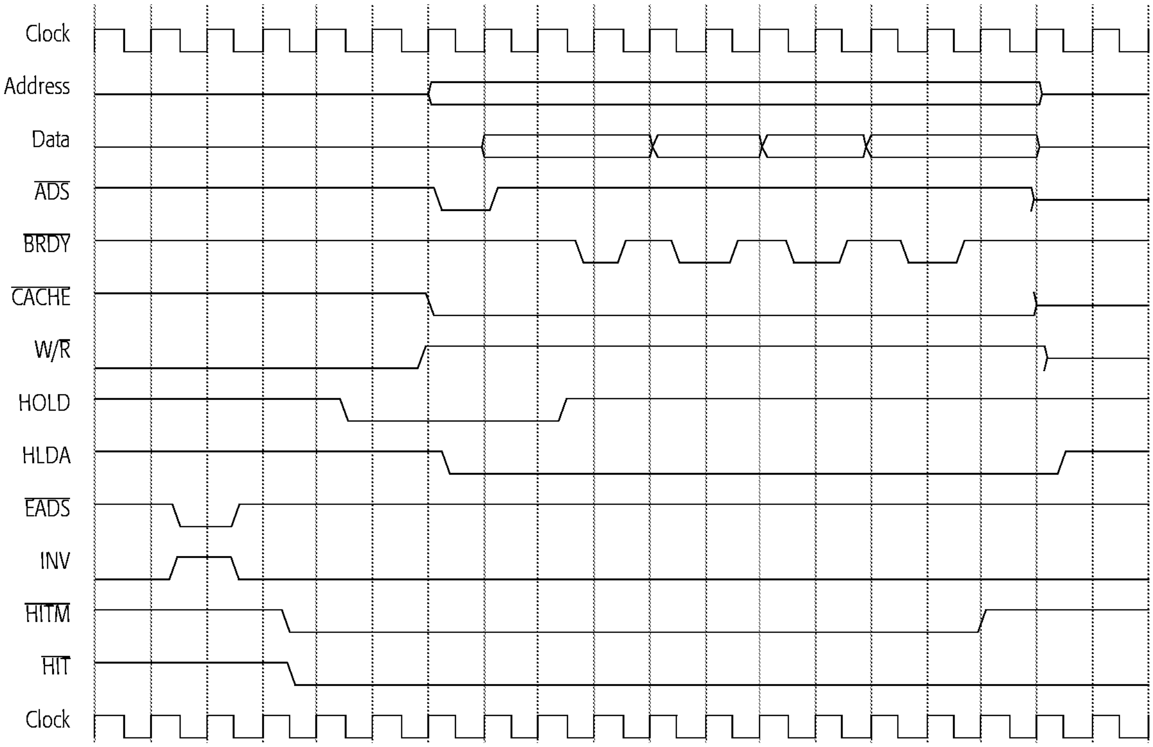


Figure 27. Invalidation to Modified Line in L1 Cache (Writeback Cycle)

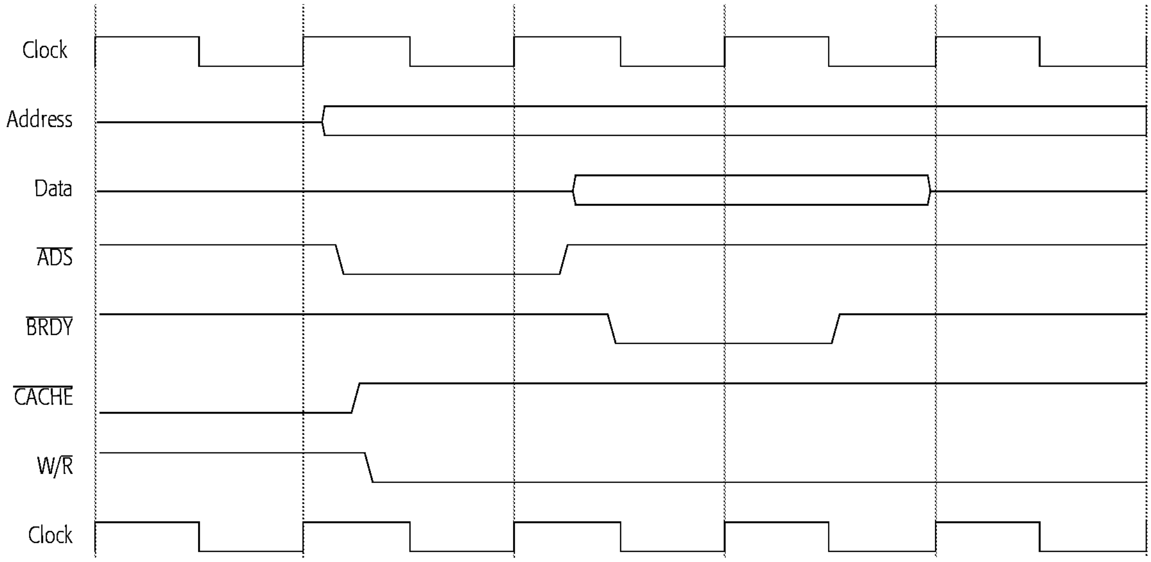


Figure 28. Single Read due to $\overline{\text{CACHE}}$ Inactive (No Wait State)

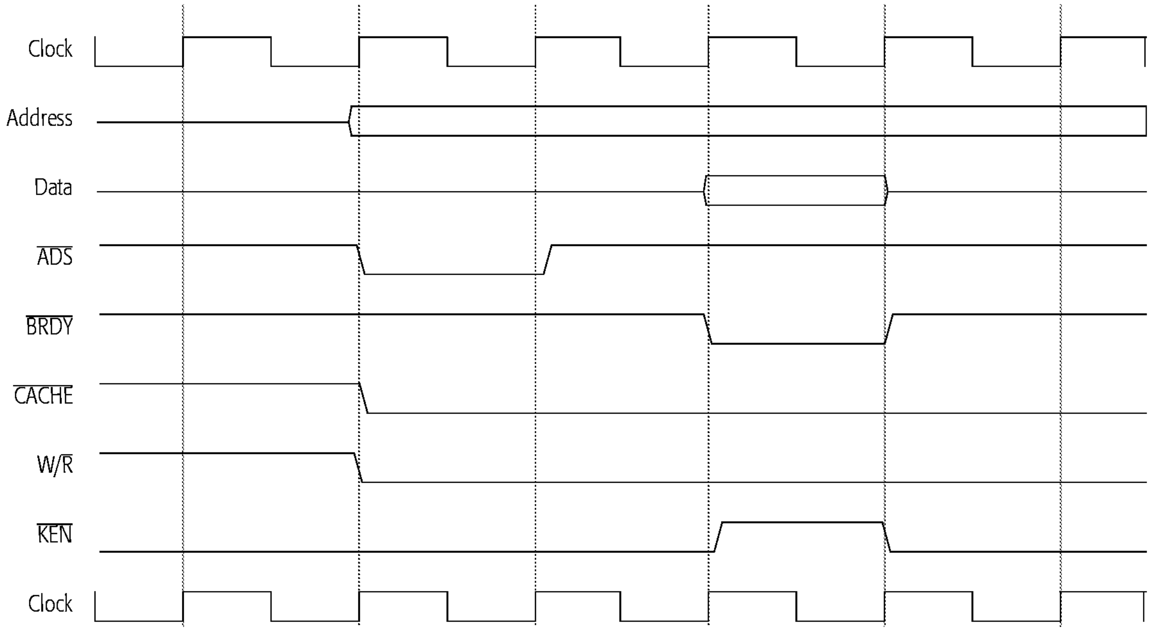


Figure 29. Single Read due to $\overline{\text{KEN}}$ Not Asserted (One Wait State)

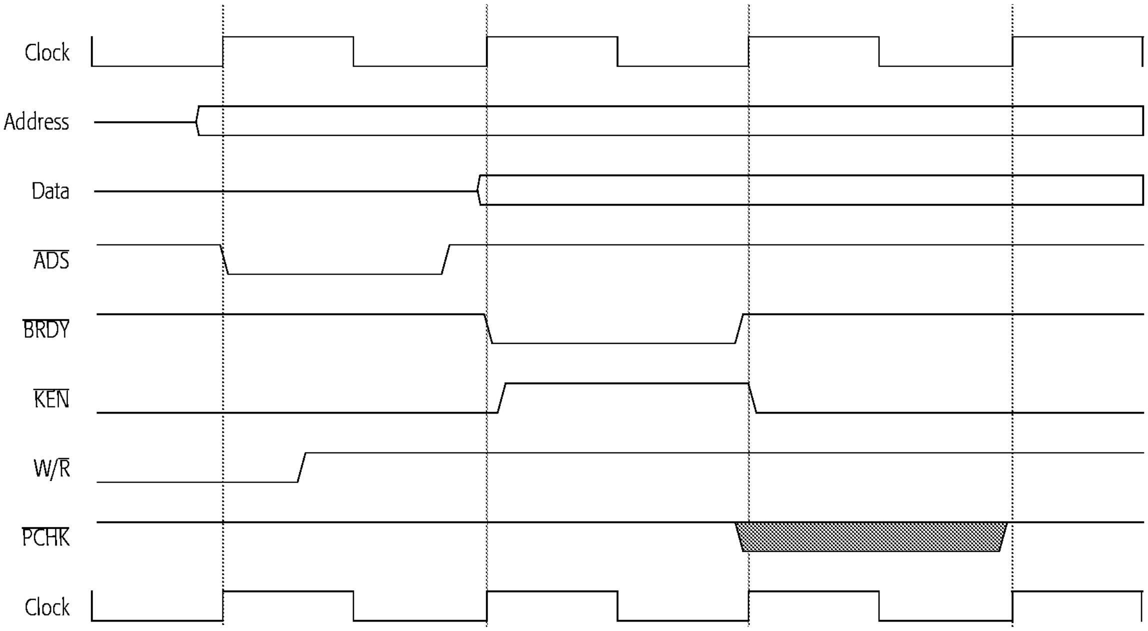


Figure 30. Single Write due to KEN Inactive (No Wait State)

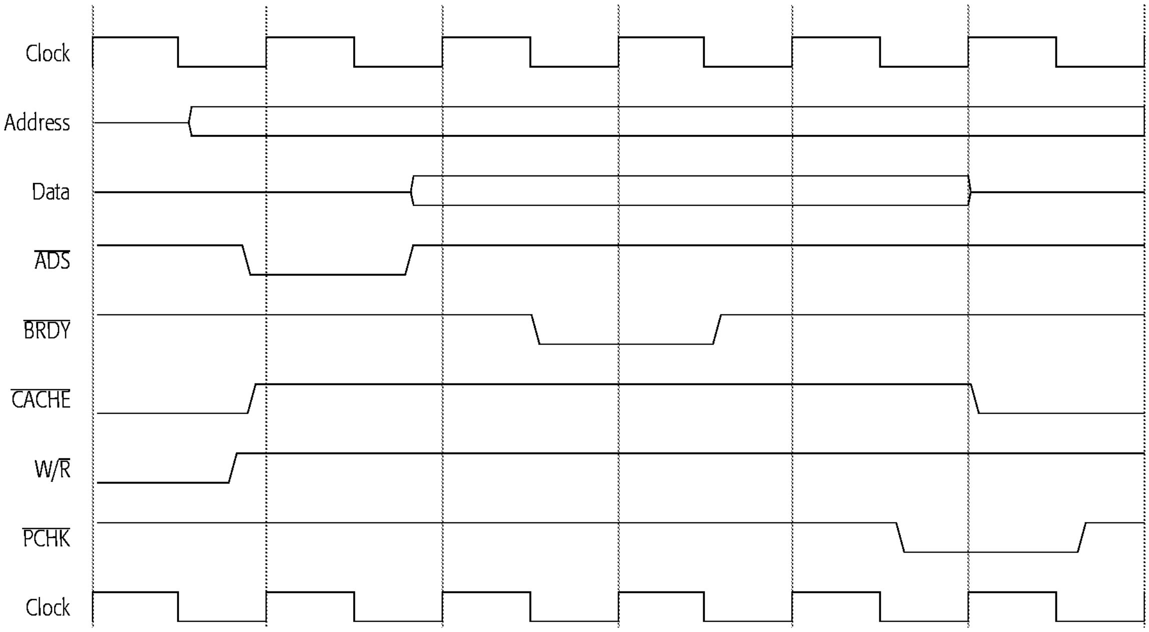


Figure 31. Single Write due to CACHE Inactive (One Wait State)

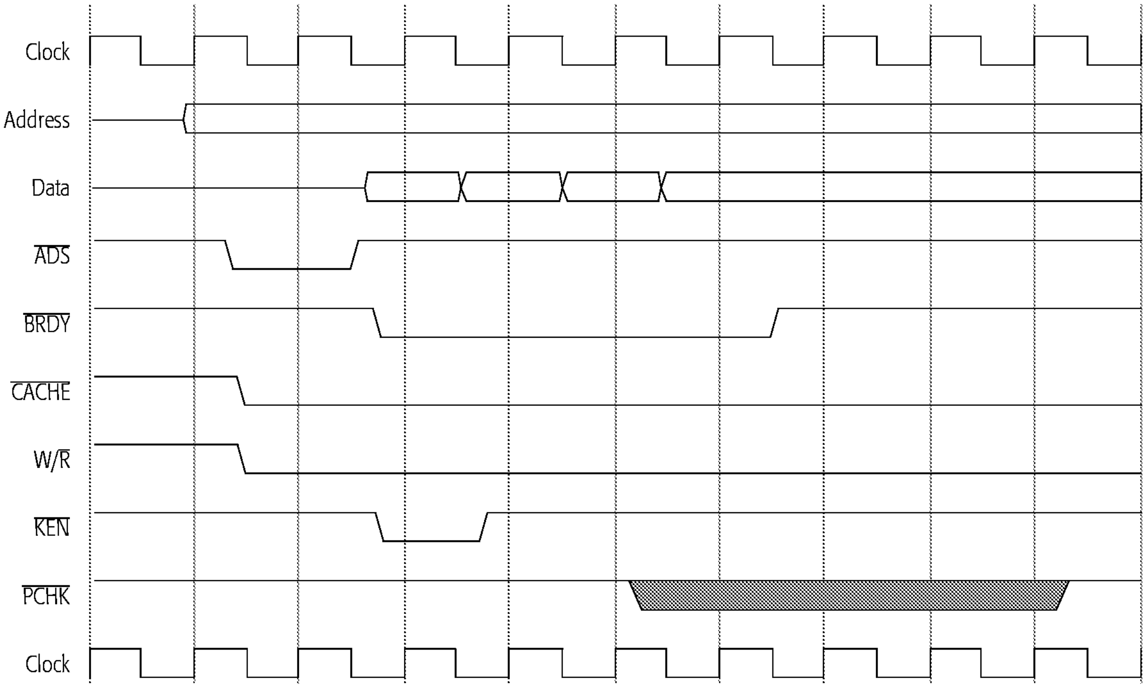


Figure 32. Burst Read (No Wait State)

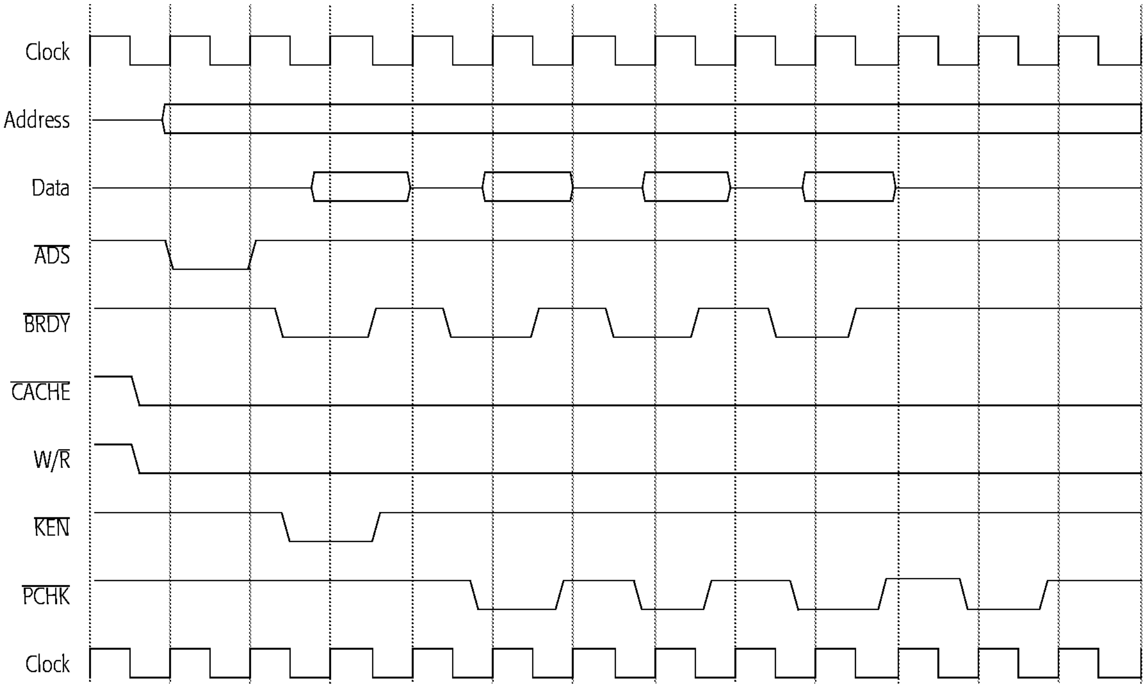


Figure 33. Burst Read (One Wait State)

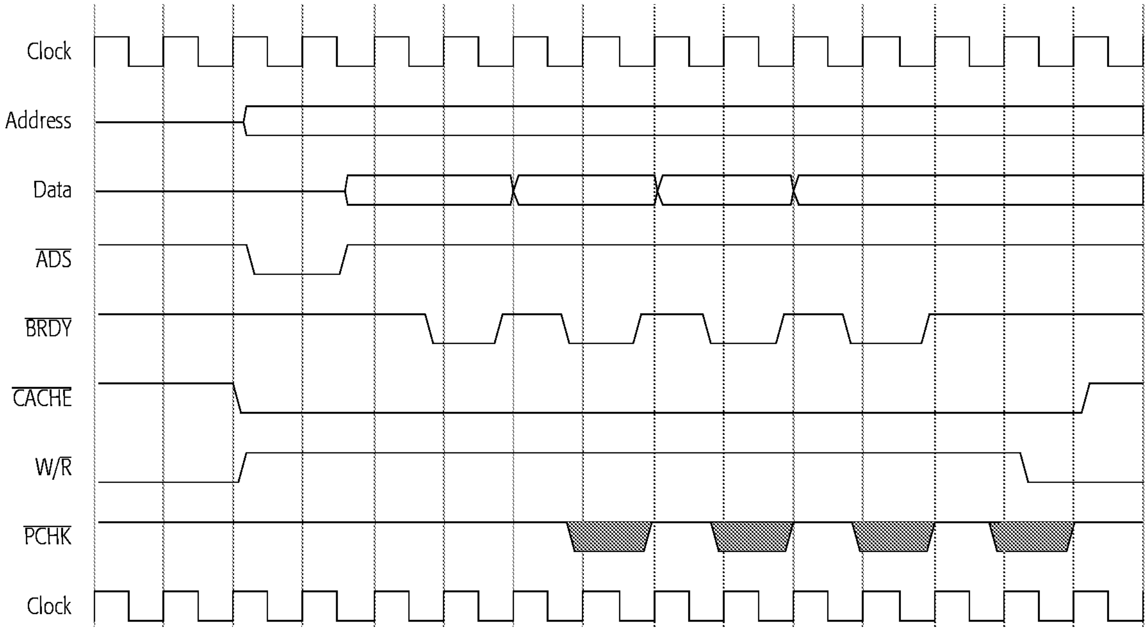


Figure 34. Burst Write (One Wait State)

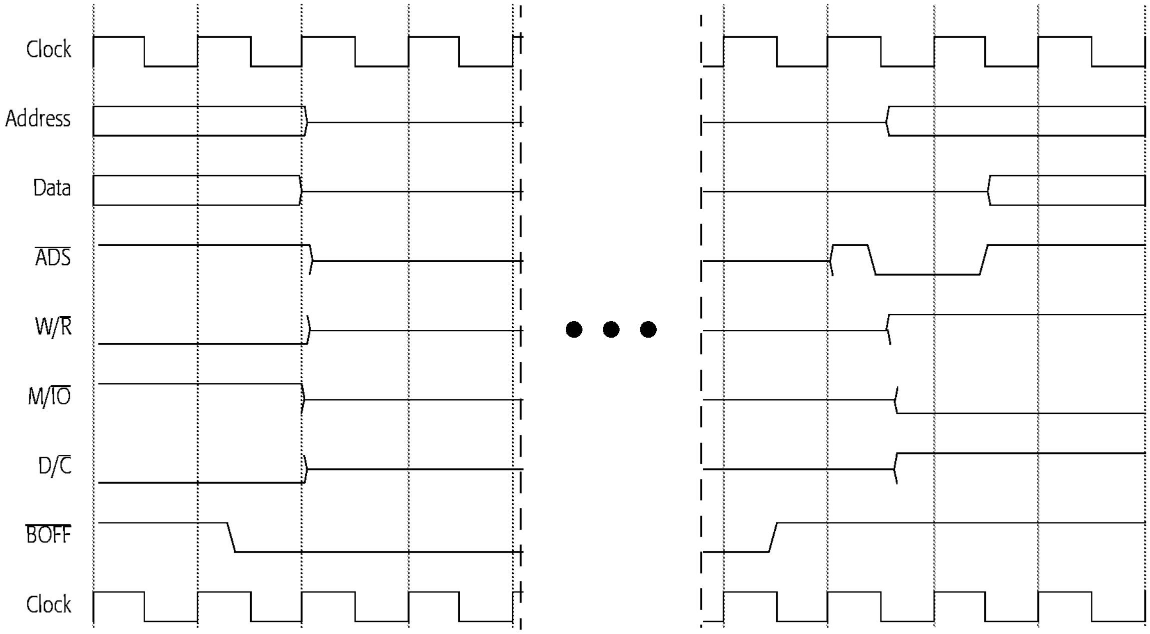


Figure 35. $\overline{\text{BOFF}}$ Timing

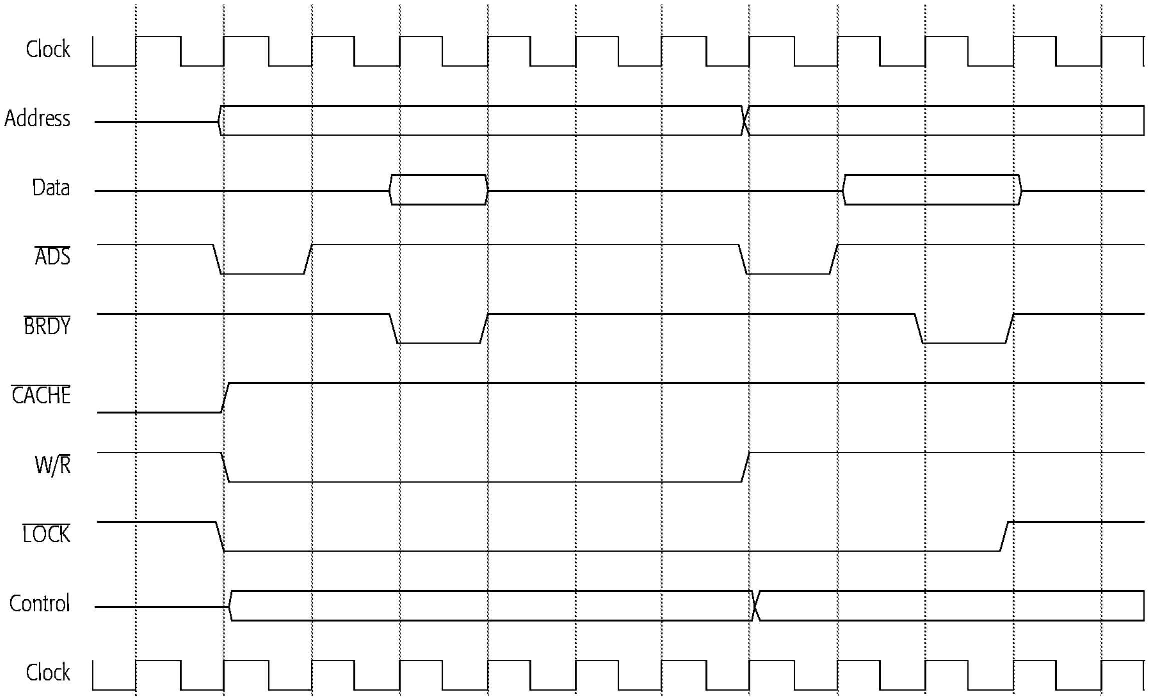


Figure 36. Locked Cycle

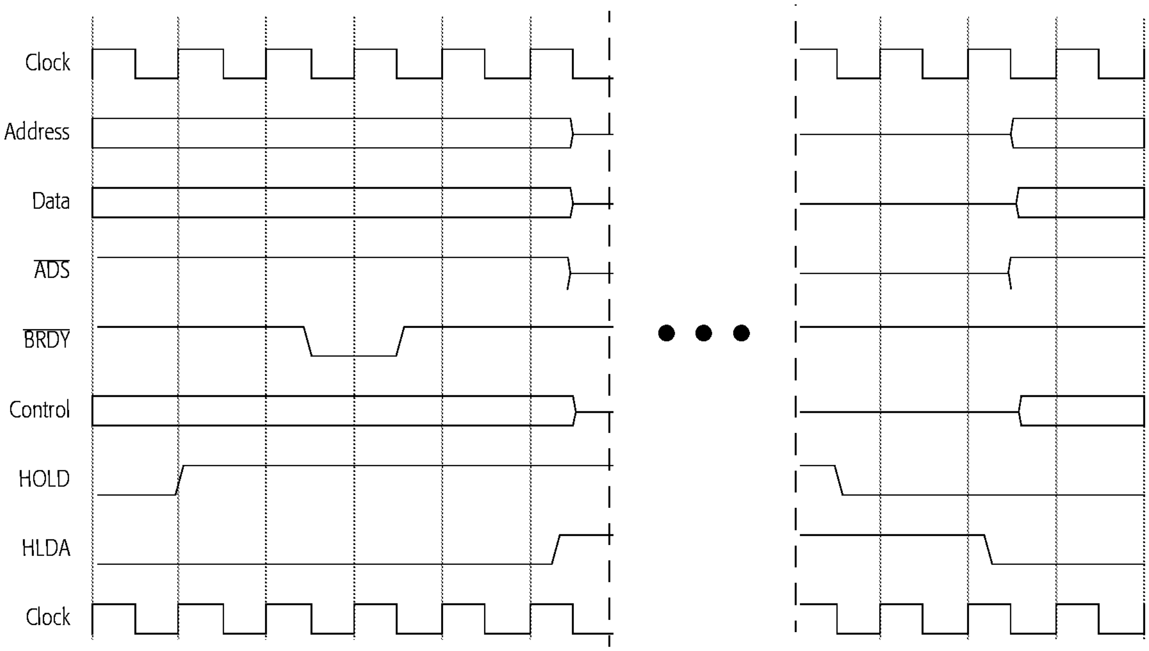


Figure 37. HOLD/HLDA Timing

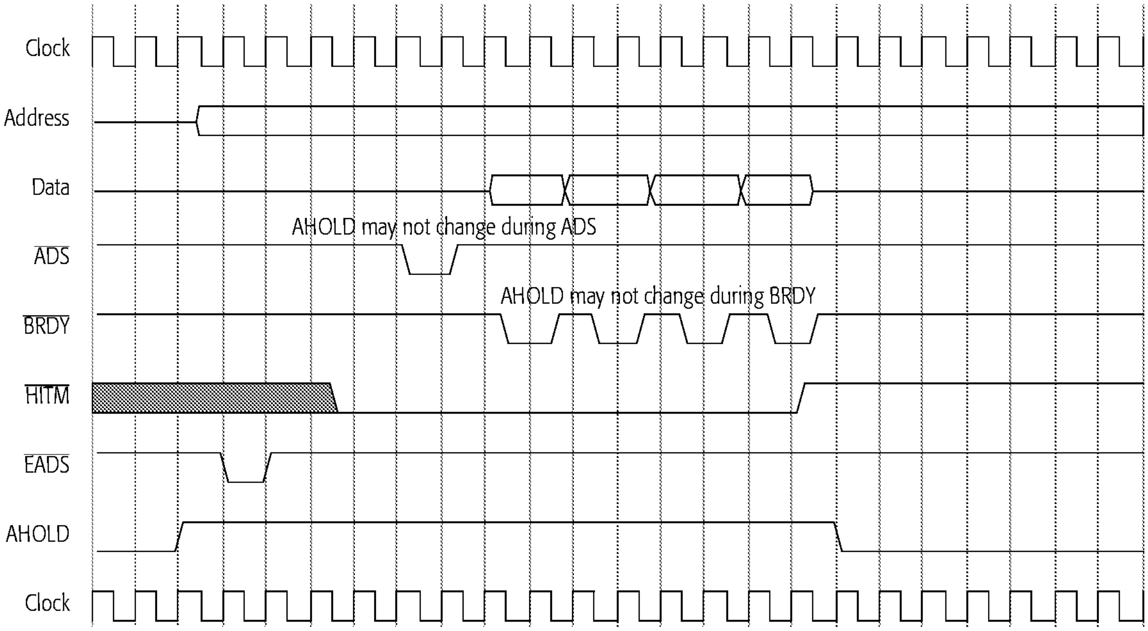

Clock
Address
Data
ADS
BRDY
HITM
EADS
AHOLD
Clock
AHOLD may not change during ADS
AHOLD may not change during BRDY

Figure 38. AHOLD Restrictions

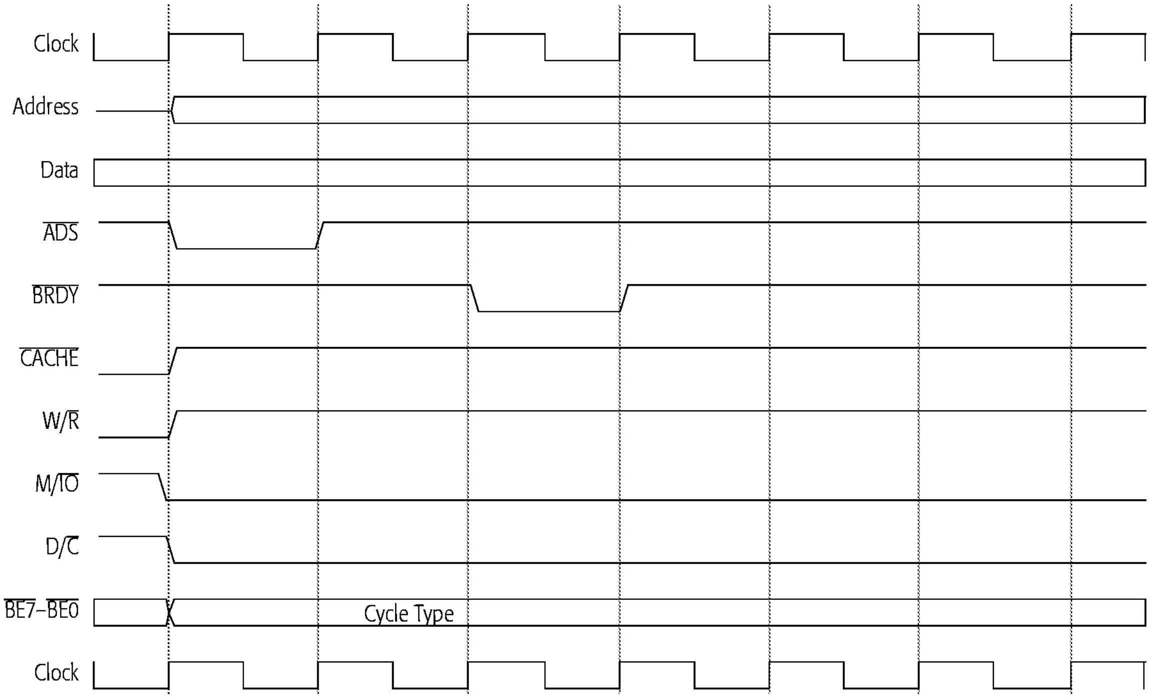


Figure 39. Special Cycle

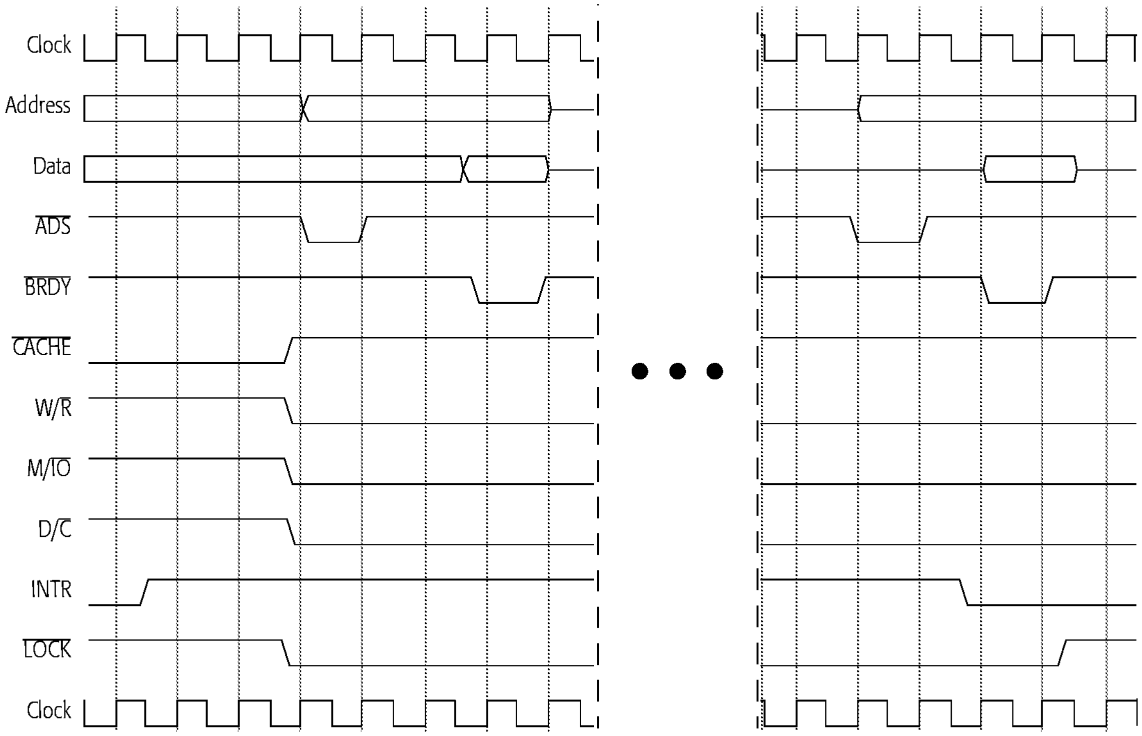


Figure 40. Interrupt Acknowledge

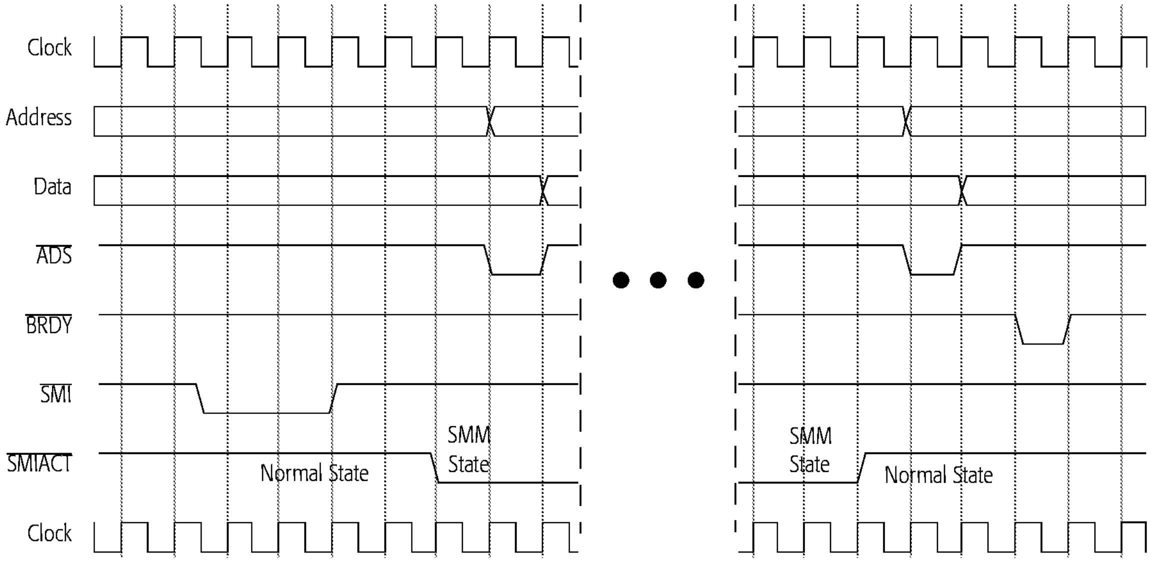


Figure 41. SMI/SMIACT Timing

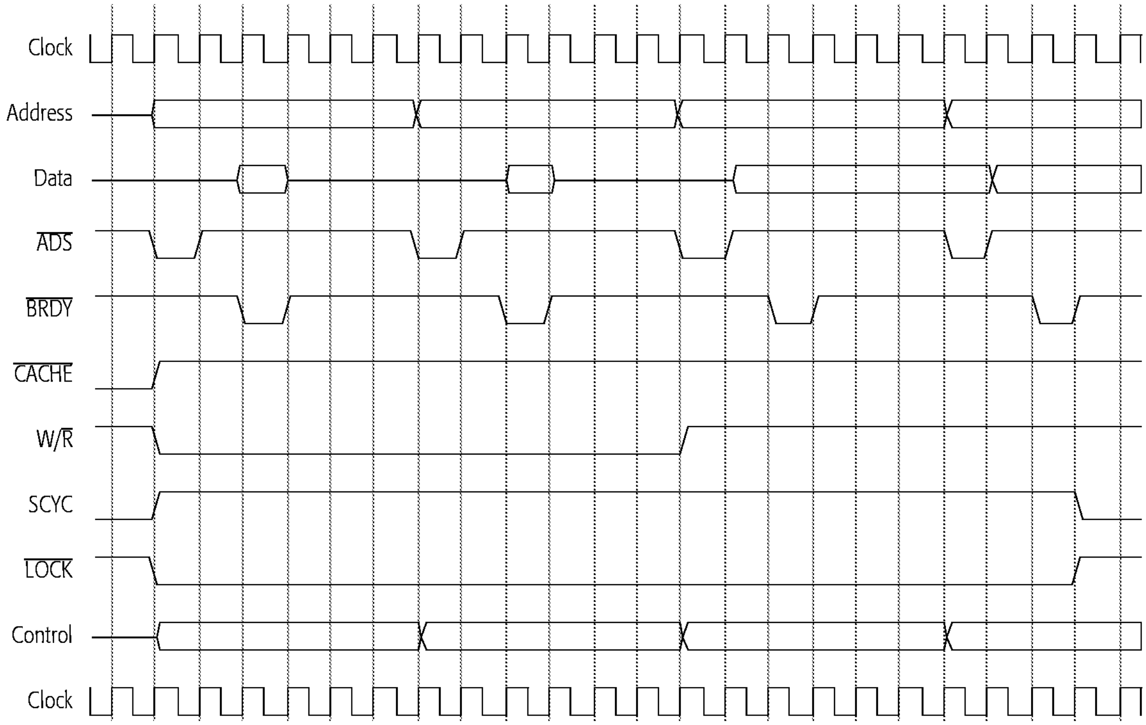


Figure 42. Split Cycle (Misaligned Locked cycle)

12 Package Thermal Specifications

The AMD-K5 processor is specified for operation when T_{CASE} (the case temperature) is within the range of 0°C to 70°C. T_{CASE} can be measured in any environment to determine whether the AMD-K5 processor is within the specified operating range. The case temperature should be measured at the center of the top surface opposite the pins.

The ambient temperature (T_A) is guaranteed as long as T_{CASE} is not violated. The ambient temperature can be calculated from θ_{CA} and from the following equation:

$$T_{CASE} = T_A + (P \cdot \theta_{CA})$$

where:

T_A, T_{CASE} = Ambient and Case Temperature
θ_{CA} = Case-to-ambient Thermal Resistance
P = Maximum Power Consumption

The value for θ_{CA} is given in Table 27 for the 1.90 sq. in., 296-pin, ceramic SPGA case. Maximum T_A is shown in Table 28 and Table 29. The values for processor frequency in Table 28 apply to the AMD-K5 processor model 0. The values for processor frequency in Table 29 apply to the AMD-K5 processor models 1 and 2.

Table 27. θ_{CA} for the AMD-K5 Processor in 296-pin SPGA Package for Typical Heat Sinks with Fans

Heat Sink With Fan (length x width x height)	θ_{CA} (°C/W)	Manufacturer - Part Number
1.885 in x 1.9 in x 1.04 in	0.81	Thermalloy, Inc. - 20961-TCM
1.95 in x 1.79 in x 1.06 in	1.3	Wakefield Engineering, Inc. - 709-100AB124
1.96 in x 1.96 in x 0.65 in	1.5	AAVID - 355455F00267

Notes:

1. *Thermal interface material (e.g., thermal grease or thermal compound) is required between the top of the processor case and the base of the heat sink.*

Table 28. Model 0 Maximum T_A in °C

Heat Sink	Airflow of 0 (0) ft/min. (m/sec)		
	PR75[1]	PR90[2]	PR100[3]
Thermalloy Heat Sink w/Fan	60.6	48.7	47.5
Wakefield Heat Sink w/Fan	54.9	41.9	39.9
AAVID Heat Sink w/Fan	52.6	39.1	36.8

Notes:

1. $T_{CASE} = 70°C$, $V_{CC} = 3.52$ V, $I_{CC} = 3300$ mA
2. $T_{CASE} = 60°C$, $V_{CC} = 3.52$ V, $I_{CC} = 3960$ mA
3. $T_{CASE} = 60°C$, $V_{CC} = 3.52$ V, $I_{CC} = 4400$ mA

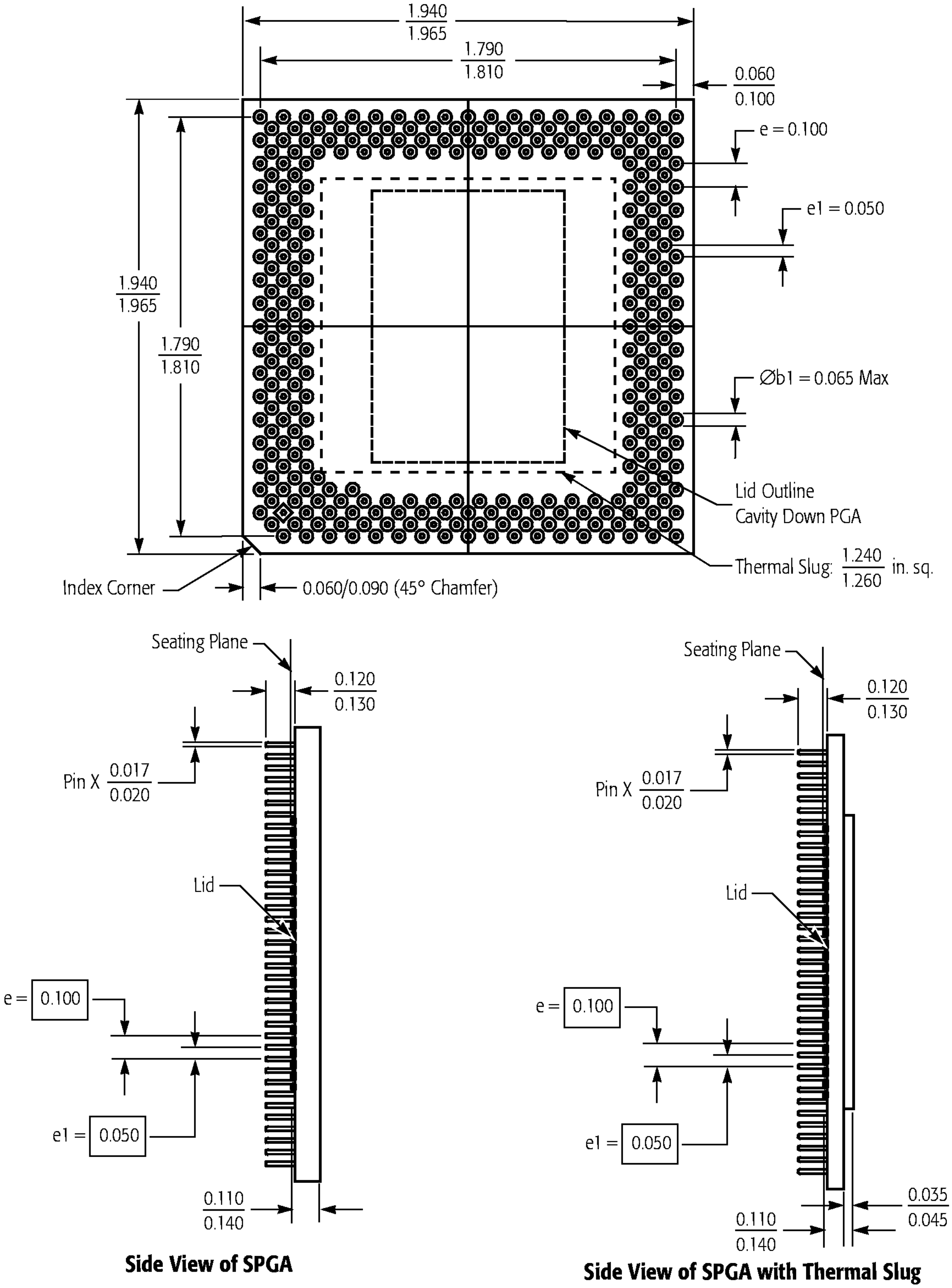
Table 29. Models 1 and 2 Maximum T_A in °C

Heat Sink	Airflow of 0 (0) ft/min. (m/sec)		
	PR120[1]	PR133[2]	PR166[3]
Thermalloy Heat Sink w/Fan	60.0	48.9	52.0
Wakefield Heat Sink w/Fan	53.9	42.2	44.2
AAVID Heat Sink w/Fan	51.5	39.4	41.0

Notes:

1. $T_{CASE} = 70°C$, $V_{CC} = 3.52$ V, $I_{CC} = 3510$ mA
2. $T_{CASE} = 60°C$, $V_{CC} = 3.52$ V, $I_{CC} = 3900$ mA
3. $T_{CASE} = 65°C$, $V_{CC} = 3.52$ V, $I_{CC} = 4550$ mA

13 Physical Dimensions



Side View of SPGA

Side View of SPGA with Thermal Slug

Figure 43. 296-Pin Ceramic Staggered Pin Grid Array (SPGA)

14 Pin Description Diagram (Model 0)

● Control pins
⏚ V_{SS} pins
△ V_{CC} pins
□ Data pins
○ Address pins
T Test pins
⊘ NC, INC (Internal No Connect) pins
⊗ Reserved pins

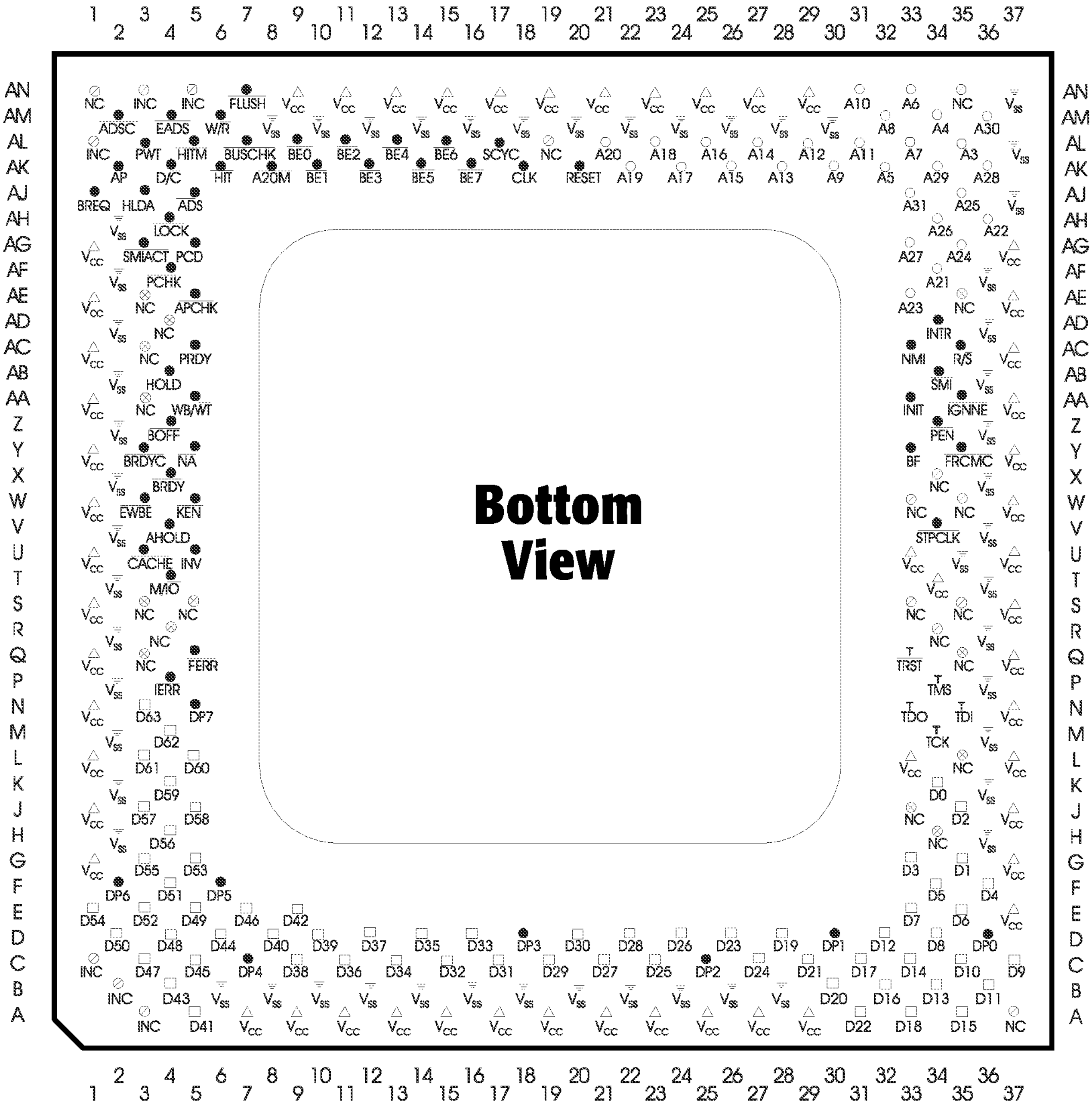

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37
AN AM AL AK AJ AH AG AF AE AD AC AB AA Z Y X W V U T S R Q P N M L K J H G F E D C B A
AN: NC INC INC FLUSH VCC VCC VCC VCC VCC VCC VCC VCC VCC VCC VCC A10 A6 NC VSS
AM: ADSC EADS W/R VSS VSS VSS VSS VSS VSS VSS VSS VSS VSS VSS A8 A4 A30
AL: INC PWT HITM BUSCHK BE0 BE2 BE4 BE6 SCYC NC A20 A18 A16 A14 A12 A11 A7 A3 VSS
AK: AP D/C HIT A20M BE1 BE3 BE5 BE7 CLK RESET A19 A17 A15 A13 A9 A5 A29 A28
AJ: BREQ HLDA ADS A31 A25 VSS
AH: VSS LOCK A26 A22
AG: VCC SMIACT PCD A27 A24 VCC
AF: VSS PCHK A21 VSS
AE: VCC NC APCHK A23 NC VCC
AD: VSS NC INTR VSS
AC: VCC NC PRDY NMI R/S VCC
AB: VSS HOLD SMI VSS
AA: VCC NC WB/WT INIT IGNNE VCC
Z: VSS BOFF PEN VSS
Y: VCC BRDYC NA BF FRCMC VCC
X: VSS BRDY NC VSS
W: VCC EWBE KEN NC NC VCC
V: VSS AHOLD STPCLK VSS
U: VCC CACHE INV VCC VSS VCC
T: VSS M/IO VCC VSS
S: VCC NC NC NC NC VCC
R: VSS NC NC VSS
Q: VCC NC FERR TRST NC VCC
P: VSS IERR TMS VSS
N: VCC D63 DP7 TDO TDI VCC
M: VSS D62 TCK VSS
L: VCC D61 D60 VCC NC VCC
K: VSS D59 D0 VSS
J: VCC D57 D58 NC D2 VCC
H: VSS D56 NC VSS
G: VCC D55 D53 D3 D1 VCC
F: DP6 D51 DP5 D5 D4
E: D54 D52 D49 D46 D42 D7 D6 VCC
D: D50 D48 D44 D40 D39 D37 D35 D33 DP3 D30 D28 D26 D23 D19 DP1 D12 D8 DP0
C: INC D47 D45 DP4 D38 D36 D34 D32 D31 D29 D27 D25 DP2 D24 D21 D17 D14 D10 D9
B: INC D43 VSS VSS VSS VSS VSS VSS VSS VSS VSS VSS VSS VSS D20 D16 D13 D11
A: INC D41 VCC VCC VCC VCC VCC VCC VCC VCC VCC VCC VCC VCC D22 D18 D15 NC
Bottom View

Figure 44. AMD-K5 Model 0 Processor Pin-Side View

15 Pin Designations (Model 0)

Functional Grouping

Address		Data		Control		Test		NC	V_{cc}	V_{ss}	Reserved
Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin No.	Pin No.	Pin No.	Pin No.
A3	AL-35	D0	K-34	$\overline{\text{A20M}}$	AK-08	TCK	M-34	A-37	A-07	B-06	H-34
A4	AM-34	D1	G-35	$\overline{\text{ADS}}$	AJ-05	TDI	N-35	R-34	A-09	B-08	J-33
A5	AK-32	D2	J-35	$\overline{\text{ADSC}}$	AM-02	TDO	N-33	S-33	A-11	B-10	L-35
A6	AN-33	D3	G-33	AHOLD	V-04	TMS	P-34	S-35	A-13	B-12	Q-03
A7	AL-33	D4	F-36	AP	AK-02	$\overline{\text{TRST}}$	Q-33	W-33	A-15	B-14	Q-35
A8	AM-32	D5	F-34	$\overline{\text{APCHK}}$	AE-05			W-35	A-17	B-16	R-04
A9	AK-30	D6	E-35	$\overline{\text{BE0}}$	AL-09			X-34	A-19	B-18	S-03
A10	AN-31	D7	E-33	$\overline{\text{BE1}}$	AK-10			AL-19	A-21	B-20	S-05
A11	AL-31	D8	D-34	$\overline{\text{BE2}}$	AL-11			AN-01	A-23	B-22	AA-03
A12	AL-29	D9	C-37	$\overline{\text{BE3}}$	AK-12			AN-35	A-25	B-24	AC-03
A13	AK-28	D10	C-35	$\overline{\text{BE4}}$	AL-13				A-27	B-26	AD-04
A14	AL-27	D11	B-36	$\overline{\text{BE5}}$	AK-14				A-29	B-28	AE-03
A15	AK-26	D12	D-32	$\overline{\text{BE6}}$	AL-15			**INC**	E-37	H-02	AE-35
A16	AL-25	D13	B-34	$\overline{\text{BE7}}$	AK-16				G-01	H-36	
A17	AK-24	D14	C-33	BF	Y-33				G-37	K-02	
A18	AL-23	D15	A-35	$\overline{\text{BOFF}}$	Z-04			A-03	J-01	K-36	
A19	AK-22	D16	B-32	$\overline{\text{BRDY}}$	X-04			B-02	J-37	M-02	
A20	AL-21	D17	C-31	$\overline{\text{BRDYC}}$	Y-03			C-01	L-01	M-36	
A21	AF-34	D18	A-33	BREQ	AJ-01			AL-01	L-33	P-02	
A22	AH-36	D19	D-28	$\overline{\text{BUSCHK}}$	AL-07			AN-03	L-37	P-36	
A23	AE-33	D20	B-30	$\overline{\text{CACHE}}$	U-03			AN-05	N-01	R-02	
A24	AG-35	D21	C-29	CLK	AK-18				N-37	R-36	
A25	AJ-35	D22	A-31	D/$\overline{\text{C}}$	AK-04				Q-01	T-02	
A26	AH-34	D23	D-26	DP0	D-36				Q-37	T-36	
A27	AG-33	D24	C-27	DP1	D-30				S-01	U-35	
A28	AK-36	D25	C-23	DP2	C-25				S-37	V-02	
A29	AK-34	D26	D-24	DP3	D-18				T-34	V-36	
A30	AM-36	D27	C-21	DP4	C-07				U-01	X-02	
A31	AJ-33	D28	D-22	DP5	F-06				U-33	X-36	
		D29	C-19	DP6	F-02				U-37	Z-02	
		D30	D-20	DP7	N-05				W-01	Z-36	
		D31	C-17	$\overline{\text{EADS}}$	AM-04				W-37	AB-02	
		D32	C-15	$\overline{\text{EWBE}}$	W-03				Y-01	AB-36	
		D33	D-16	$\overline{\text{FERR}}$	Q-05				Y-37	AD-02	
		D34	C-13	$\overline{\text{FLUSH}}$	AN-07				AA-01	AD-36	
		D35	D-14	$\overline{\text{FRCMC}}$	Y-35				AA-37	AF-02	
		D36	C-11	$\overline{\text{HIT}}$	AK-06				AC-01	AF-36	
		D37	D-12	$\overline{\text{HITM}}$	AL-05				AC-37	AH-02	
		D38	C-09	HLDA	AJ-03				AE-01	AJ-37	
		D39	D-10	HOLD	AB-04				AE-37	AL-37	
		D40	D-08	$\overline{\text{IERR}}$	P-04				AG-01	AM-08	
		D41	A-05	$\overline{\text{IGNNE}}$	AA-35				AG-37	AM-10	
		D42	E-09	INIT	AA-33				AN-09	AM-12	
		D43	B-04	INTR	AD-34				AN-11	AM-14	
		D44	D-06	INV	U-05				AN-13	AM-16	
		D45	C-05	$\overline{\text{KEN}}$	W-05				AN-15	AM-18	
		D46	E-07	$\overline{\text{LOCK}}$	AH-04				AN-17	AM-20	
		D47	C-03	M/$\overline{\text{IO}}$	T-04				AN-19	AM-22	
		D48	D-04	$\overline{\text{NA}}$	Y-05				AN-21	AM-24	
		D49	E-05	NMI	AC-33				AN-23	AM-26	
		D50	D-02	PCD	AG-05				AN-25	AM-28	
		D51	F-04	$\overline{\text{PCHK}}$	AF-04				AN-27	AM-30	
		D52	E-03	$\overline{\text{PEN}}$	Z-34				AN-29	AN-37	
		D53	G-05	PRDY	AC-05						
		D54	E-01	PWT	AL-03						
		D55	G-03	RESET	AK-20						
		D56	H-04	R/$\overline{\text{S}}$	AC-35						
		D57	J-03	SCYC	AL-17						
		D58	J-05	$\overline{\text{SMI}}$	AB-34						
		D59	K-04	$\overline{\text{SMIACT}}$	AG-03						
		D60	L-05	$\overline{\text{STPCLK}}$	V-34						
		D61	L-03	W/$\overline{\text{R}}$	AM-06						
		D62	M-04	WB/$\overline{\text{WT}}$	AA-05						
		D63	N-03								

16 Pin Description Diagram (Models 1 and 2)

- ● Control pins
- V_{SS} pins
- △ V_{CC} pins
- □ Data pins
- ○ Address pins
- **T** Test pins
- ⊘ NC, INC (Internal No Connect) pins
- ⊗ Reserved pins

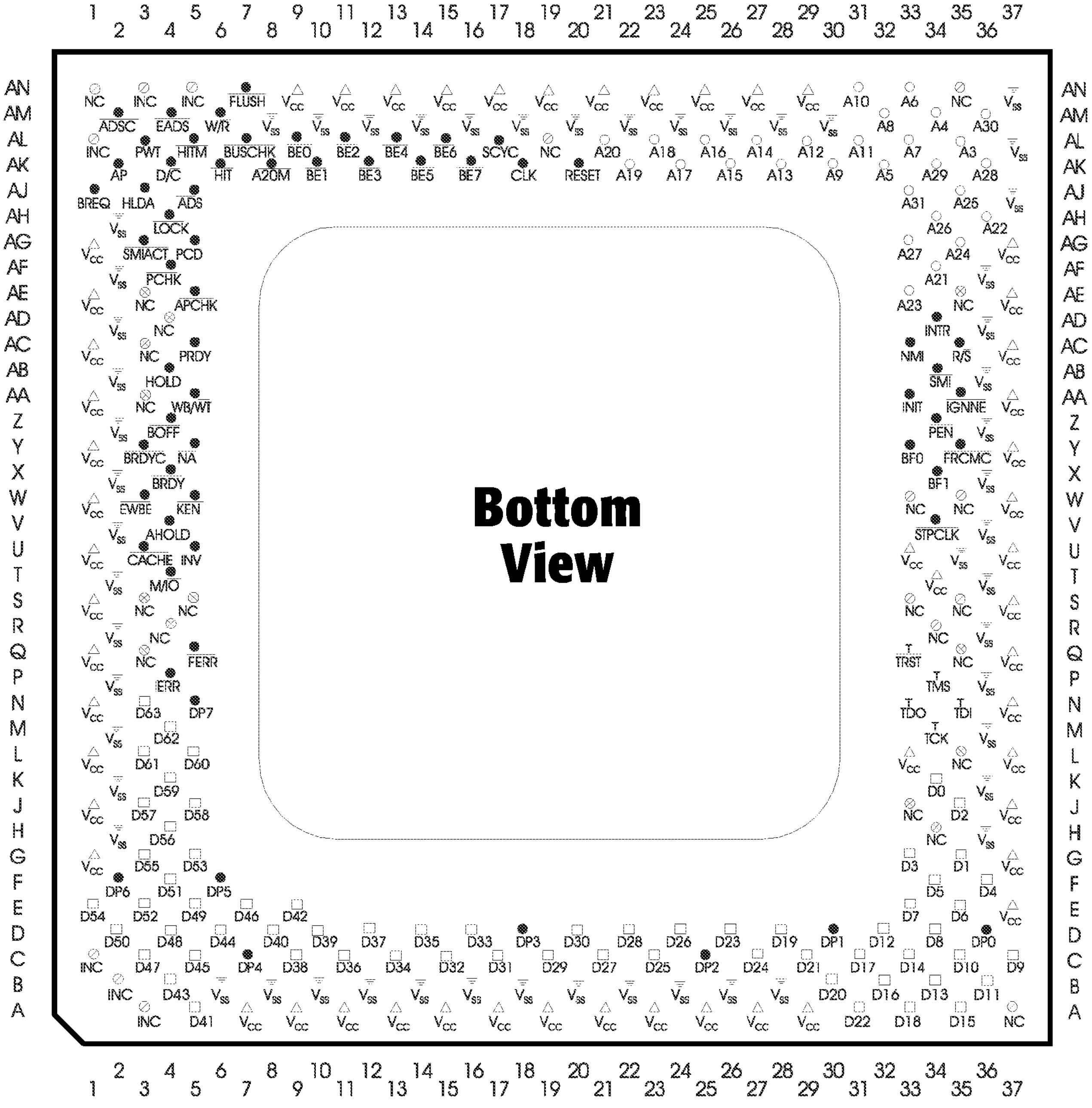

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37
AN: NC INC INC FLUSH VCC VCC VCC VCC VCC VCC VCC VCC VCC VCC VCC A10 A6 NC VSS
AM: ADSC EADS W/R VSS VSS VSS VSS VSS VSS VSS VSS VSS VSS VSS VSS A8 A4 A30
AL: INC PWT HITM BUSCHK BE0 BE2 BE4 BE6 SCYC NC A20 A18 A16 A14 A12 A11 A7 A3 VSS
AK: AP D/C HIT A20M BE1 BE3 BE5 BE7 CLK RESET A19 A17 A15 A13 A9 A5 A29 A28
AJ: BREQ HLDA ADS A31 A25 VSS
AH: VSS LOCK A26 A22
AG: VCC SMIACT PCD A27 A24 VCC
AF: VSS PCHK A21 VSS
AE: VCC NC APCHK A23 NC VCC
AD: VSS NC INTR VSS
AC: VCC NC PRDY NMI R/S VCC
AB: VSS HOLD SMI VSS
AA: VCC NC WB/WT INIT IGNNE VCC
Z: VSS BOFF PEN VSS
Y: VCC BRDYC NA BF0 FRCMC VCC
X: VSS BRDY BF1 VSS
W: VCC EWBE KEN NC NC VCC
V: VSS AHOLD STPCLK VSS
U: VCC CACHE INV VCC VSS VCC
T: VSS M/IO VCC VSS
S: VCC NC NC NC NC VCC
R: VSS NC NC VSS
Q: VCC NC FERR TRST NC VCC
P: VSS IERR TMS VSS
N: VCC D63 DP7 TDO TDI VCC
M: VSS D62 TCK VSS
L: VCC D61 D60 VCC NC VCC
K: VSS D59 D0 VSS
J: VCC D57 D58 NC D2 VCC
H: VSS D56 NC VSS
G: VCC D55 D53 D3 D1 VCC
F: DP6 D51 DP5 D5 D4
E: D54 D52 D49 D46 D42 D7 D6 VCC
D: D50 D48 D44 D40 D39 D37 D35 D33 DP3 D30 D28 D26 D23 D19 DP1 D12 D8 DP0
C: INC D47 D45 DP4 D38 D36 D34 D32 D31 D29 D27 D25 DP2 D24 D21 D17 D14 D10 D9
B: INC D43 VSS VSS VSS VSS VSS VSS VSS VSS VSS VSS VSS VSS D20 D16 D13 D11
A: INC D41 VCC VCC VCC VCC VCC VCC VCC VCC VCC VCC VCC VCC D22 D18 D15 NC
Bottom View

Figure 45. AMD-K5 Models 1 and 2 Processor Pin-Side View

17 Pin Designations (Models 1 and 2)

Functional Grouping

Address		Data		Control		Test		NC	V_{cc}	V_{ss}	Reserved
Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin No.	Pin No.	Pin No.	Pin No.
A3	AL-35	D0	K-34	$\overline{\text{A20M}}$	AK-08	TCK	M-34	A-37	A-07	B-06	H-34
A4	AM-34	D1	G-35	$\overline{\text{ADS}}$	AJ-05	TDI	N-35	R-34	A-09	B-08	J-33
A5	AK-32	D2	J-35	$\overline{\text{ADSC}}$	AM-02	TDO	N-33	S-33	A-11	B-10	L-35
A6	AN-33	D3	G-33	AHOLD	V-04	TMS	P-34	S-35	A-13	B-12	Q-03
A7	AL-33	D4	F-36	AP	AK-02	$\overline{\text{TRST}}$	Q-33	W-33	A-15	B-14	Q-35
A8	AM-32	D5	F-34	$\overline{\text{APCHK}}$	AE-05			W-35	A-17	B-16	R-04
A9	AK-30	D6	E-35	$\overline{\text{BE0}}$	AL-09			AL-19	A-19	B-18	S-03
A10	AN-31	D7	E-33	$\overline{\text{BE1}}$	AK-10			AN-01	A-21	B-20	S-05
A11	AL-31	D8	D-34	$\overline{\text{BE2}}$	AL-11			AN-35	A-23	B-22	AA-03
A12	AL-29	D9	C-37	$\overline{\text{BE3}}$	AK-12				A-25	B-24	AC-03
A13	AK-28	D10	C-35	$\overline{\text{BE4}}$	AL-13				A-27	B-26	AD-04
A14	AL-27	D11	B-36	$\overline{\text{BE5}}$	AK-14			**INC**	A-29	B-28	AE-03
A15	AK-26	D12	D-32	$\overline{\text{BE6}}$	AL-15				E-37	H-02	AE-35
A16	AL-25	D13	B-34	$\overline{\text{BE7}}$	AK-16			A-03	G-01	H-36	
A17	AK-24	D14	C-33	BF0	Y-33			B-02	G-37	K-02	
A18	AL-23	D15	A-35	BF1	X-34			C-01	J-01	K-36	
A19	AK-22	D16	B-32	$\overline{\text{BOFF}}$	Z-04			AL-01	J-37	M-02	
A20	AL-21	D17	C-31	$\overline{\text{BRDY}}$	X-04			AN-03	L-01	M-36	
A21	AF-34	D18	A-33	$\overline{\text{BRDYC}}$	Y-03			AN-05	L-33	P-02	
A22	AH-36	D19	D-28	BREQ	AJ-01				L-37	P-36	
A23	AE-33	D20	B-30	$\overline{\text{BUSCHK}}$	AL-07				N-01	R-02	
A24	AG-35	D21	C-29	$\overline{\text{CACHE}}$	U-03				N-37	R-36	
A25	AJ-35	D22	A-31	CLK	AK-18				Q-01	T-02	
A26	AH-34	D23	D-26	D/$\overline{\text{C}}$	AK-04				Q-37	T-36	
A27	AG-33	D24	C-27	DP0	D-36				S-01	U-35	
A28	AK-36	D25	C-23	DP1	D-30				S-37	V-02	
A29	AK-34	D26	D-24	DP2	C-25				T-34	V-36	
A30	AM-36	D27	C-21	DP3	D-18				U-01	X-02	
A31	AJ-33	D28	D-22	DP4	C-07				U-33	X-36	
		D29	C-19	DP5	F-06				U-37	Z-02	
		D30	D-20	DP6	F-02				W-01	Z-36	
		D31	C-17	DP7	N-05				W-37	AB-02	
		D32	C-15	$\overline{\text{EADS}}$	AM-04				Y-01	AB-36	
		D33	D-16	$\overline{\text{EWBE}}$	W-03				Y-37	AD-02	
		D34	C-13	$\overline{\text{FERR}}$	Q-05				AA-01	AD-36	
		D35	D-14	$\overline{\text{FLUSH}}$	AN-07				AA-37	AF-02	
		D36	C-11	$\overline{\text{FRCMC}}$	Y-35				AC-01	AF-36	
		D37	D-12	$\overline{\text{HIT}}$	AK-06				AC-37	AH-02	
		D38	C-09	$\overline{\text{HITM}}$	AL-05				AE-01	AJ-37	
		D39	D-10	HLDA	AJ-03				AE-37	AL-37	
		D40	D-08	HOLD	AB-04				AG-01	AM-08	
		D41	A-05	$\overline{\text{IERR}}$	P-04				AG-37	AM-10	
		D42	E-09	$\overline{\text{IGNNE}}$	AA-35				AN-09	AM-12	
		D43	B-04	INIT	AA-33				AN-11	AM-14	
		D44	D-06	INTR	AD-34				AN-13	AM-16	
		D45	C-05	INV	U-05				AN-15	AM-18	
		D46	E-07	$\overline{\text{KEN}}$	W-05				AN-17	AM-20	
		D47	C-03	$\overline{\text{LOCK}}$	AH-04				AN-19	AM-22	
		D48	D-04	M/$\overline{\text{IO}}$	T-04				AN-21	AM-24	
		D49	E-05	$\overline{\text{NA}}$	Y-05				AN-23	AM-26	
		D50	D-02	NMI	AC-33				AN-25	AM-28	
		D51	F-04	PCD	AG-05				AN-27	AM-30	
		D52	E-03	$\overline{\text{PCHK}}$	AF-04				AN-29	AN-37	
		D53	G-05	$\overline{\text{PEN}}$	Z-34						
		D54	E-01	PRDY	AC-05						
		D55	G-03	PWT	AL-03						
		D56	H-04	RESET	AK-20						
		D57	J-03	R/$\overline{\text{S}}$	AC-35						
		D58	J-05	SCYC	AL-17						
		D59	K-04	$\overline{\text{SMI}}$	AB-34						
		D60	L-05	$\overline{\text{SMIACT}}$	AG-03						
		D61	L-03	$\overline{\text{STPCLK}}$	V-34						
		D62	M-04	W/$\overline{\text{R}}$	AM-06						
		D63	N-03	WB/$\overline{\text{WT}}$	AA-05						