Mobile

AMD-K6®-III-P

Processor

Data Sheet

Publication # **22655** Rev: **C** Amendment/**0**
Issue Date: **September 1999**

The contents of this document are provided in connection with Advanced Micro Devices, Inc. ("AMD") products. AMD makes no representations or warranties with respect to the accuracy or completeness of the contents of this publication and reserves the right to make changes to specifications and product descriptions at any time without notice. No license, whether express, implied, arising by estoppel or otherwise, to any intellectual property rights is granted by this publication. Except as set forth in AMD's Standard Terms and Conditions of Sale, AMD assumes no liability whatsoever, and disclaims any express or implied warranty, relating to its products including, but not limited to, the implied warranty of merchantability, fitness for a particular purpose, or infringement of any intellectual property right.

AMD's products are not designed, intended, authorized or warranted for use as components in systems intended for surgical implant into the body, or in other applications intended to support or sustain life, or in any other application in which the failure of AMD's product could create a situation where personal injury, death, or severe property or environmental damage may occur. AMD reserves the right to discontinue or make changes to its products at any time without notice.

Trademarks

AMD, the AMD logo, K6, 3DNow!, and combinations thereof, TriLevel Cache, and Super7 are trademarks, and AMD-K6 and RISC86 are registered trademarks of Advanced Micro Devices, Inc.

MMX is a trademark of Intel Corporation.

Microsoft, Windows, and Windows NT are registered trademarks of Microsoft Corporation.

Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

Contents

List of Figures

List of Tables

Revision History

Date	Rev	Description
May 1999	A	Initial published release.
July 1999	B	Added two V_{CC2} decoupling capacitors to Figure 18, “Suggested Component Placement,” on page 75.
Sept 1999	C	Added specifications and OPNs for 400 MHz, 433 MHz, and 450 MHz frequencies in Chapter 7, “Electrical Data”, Chapter 8, “Thermal Design”, and Chapter 11, “Ordering Information”.

About This Data Sheet

The Mobile AMD-K6®-III-P Processor Data Sheet is a supplement to the *AMD-K6®-III Processor Data Sheet*, order# 21918. When combined, the two data sheets provide the complete specification of the Mobile AMD-K6-III-P processor.

1 Mobile AMD-K6®-III-P Processor

- Advanced 6-Issue RISC86® Superscalar Microarchitecture
  - Ten parallel specialized execution units
  - Multiple sophisticated x86-to-RISC86 instruction decoders
  - Advanced two-level branch prediction
  - Speculative execution
  - Out-of-order execution
  - Register renaming and data forwarding
  - Issues up to six RISC86 instructions per clock
- TriLevel Cache™ Design
  - Large On-Chip Split 64-Kbyte Level-One (L1) Cache
    - 32-Kbyte instruction cache with additional predecode cache
    - 32-Kbyte writeback dual-ported data cache
  - On-die 256-Kbyte full processor speed, backside Level-Two (L2) Cache
  - Support for external Level-Three (L3) Cache on 100-MHz frontside bus
  - MESI protocol support
- High-Performance IEEE 754-Compatible and 854-Compatible Floating-Point Unit
- Superscalar MMX™ unit supports industry-standard MMX instructions
- 3DNow!™ Technology for high-performance multimedia and 3D graphics capabilities
- Compatible with Super7™ 100-MHz frontside bus or Socket 7 66-MHz notebook design
- Socket 7-Compatible Ceramic Pin Grid Array (CPGA) Package
- Industry-Standard System Management Mode (SMM)
- IEEE 1149.1 Boundary Scan
- x86 Binary Software Compatibility
- Low Voltage 0.25-Micron Process Technology
- 21.3 Million Transistors

The Mobile AMD-K6®-III-P processor is AMD's third generation mobile AMD-K6 processor delivering the highest performance yet for x86 notebook PC systems. The Mobile AMD-K6-III-P processor is based on the advanced Mobile AMD-K6-2 core and adds AMD's unique TriLevel Cache™ design for enhanced system performance. The TriLevel Cache design provides a large 64-Kbyte L1 cache, a 256-Kbyte L2 cache operating at full processor speed on a backside bus, and up to 1 Mbyte of available L3 cache memory on the external 100-MHz frontside bus. This combination of the largest and fastest cache memory subsystem gives the Mobile AMD-K6-III-P processor its performance edge over competing x86 mobile CPU solutions.

The Mobile AMD-K6-III-P processor also incorporates a superscalar MMX unit, support for a 100-MHz frontside bus, and AMD's innovative 3DNow! technology for high-performance multimedia and 3D graphics operation.

The Mobile AMD-K6-III-P processor includes several key features for the mobile market. The processor is implemented using an AMD-developed, state-of-the-art low power 0.25-micron process technology. This process technology features a split-plane design that allows the processor core to operate at a lower voltage while the I/O portion operates at the industry-standard 3.3V level. The 0.25-micron process technology with the split-plane voltage design enables the Mobile AMD-K6-III-P processor to deliver excellent portable PC performance solutions while utilizing a lower processor core voltage, which results in lower power consumption and longer battery life. In addition, the Mobile AMD-K6-III-P processor includes the complete industry-standard System Management Mode (SMM), which is critical to system resource and power management. The Mobile AMD-K6-III-P processor also features the industry-standard Stop-Clock (STPCLK#) control circuitry and the Halt instruction, both required for implementing the ACPI power management specification. The Mobile AMD-K6-III-P processor is offered in an industry-standard Socket 7-compatible, 321-pin Ceramic Pin Grid Array (CPGA) package.

The Mobile AMD-K6-III-P processor's RISC86 microarchitecture is a decoupled decode/execution superscalar design that implements state-of-the-art design techniques to achieve leading-edge performance. Advanced design techniques implemented in the Mobile AMD-K6-III-P processor include multiple x86 instruction decode, single-clock internal RISC operations, ten execution units that support superscalar operation, out-of-order execution, data forwarding, speculative execution, and register renaming. In addition, the processor supports the industry's most advanced branch prediction logic by implementing an 8192-entry branch history table, the industry's only branch target cache, and a return address stack, which combine to deliver better than a 95% prediction rate. These design techniques enable the Mobile AMD-K6-III-P processor to issue, execute, and retire multiple x86 instructions per clock, resulting in excellent scaleable performance.

AMD's 3DNow! technology is an instruction set extension to x86 that includes 21 new instructions to improve 3D graphics operations and other single precision floating-point compute intensive operations. AMD has already shipped millions of AMD-K6 family processors with 3DNow! technology for desktop PCs, revolutionizing the 3D experience with up to four times the peak floating-point performance of previous generation solutions. AMD is now bringing this advanced capability to notebook computing, working in conjunction with advanced mobile 3D graphic controllers to reach new levels of realism in mobile computing. With support from Microsoft® and the x86 software developer community, a new generation of visually compelling applications is coming to market that support the 3DNow! technology.

The Mobile AMD-K6-III-P processor remains pin compatible with existing Socket 7 notebook solutions, however for maximum system performance, the processor works optimally in newer Super7 designs that incorporate advanced features such as support for the 100-MHz frontside bus and AGP graphics.

The Mobile AMD-K6-III-P processor has undergone extensive testing and is compatible with Windows® 98, Windows NT® and other leading operating systems. The Mobile AMD-K6-III-P processor is also compatible with more than 60,000 software applications, including the latest 3DNow! technology and MMX technology software. As the world's second-largest supplier of processors for the Windows environment, AMD has shipped more than 50 million Microsoft Windows compatible processors in the last five years.

The Mobile AMD-K6-III-P processor is the next generation in a long line of Microsoft Windows compatible processors from AMD. With its combination of state-of-the-art features, leading-edge performance, high-performance multimedia engine, x86 compatibility, and low-cost infrastructure, the Mobile AMD-K6-III-P processor is the superior choice for notebook computers.

1.1 Super7™ Platform Initiative

AMD and its industry partners are investing in the future of Socket 7 with the new Super7 platform initiative. The goal of the initiative is to maintain the competitive vitality of the Socket 7 infrastructure through a series of planned enhancements, including the development of an industry-standard 100-MHz processor bus protocol. In addition to the 100-MHz processor bus protocol, the Super7 initiative includes the introduction of chipsets that support the AGP specification, and support for a backside L2 cache and frontside L3 cache.

Super7™ Platform Enhancements:

- *100-MHz processor bus*—The Mobile AMD-K6-III-P processor supports a 100-MHz, 800 Mbyte/second frontside bus to provide a high-speed interface to Super7 platform-based chipsets. The 100-MHz interface to the frontside L3 cache and main system memory speeds up access to the frontside cache and main memory by 50 percent over the 66-MHz Socket 7 interface—resulting in a significant 10% increase in overall system performance.
- *Accelerated graphics port support*—AGP improves the performance of mid-range PCs that have small amounts of video memory on the graphics card. The industry-standard AGP specification enables a 133-MHz graphics interface and will scale to even higher levels of performance.
- *Support for backside L2 and frontside L3 cache*—The Super7 platform has the 'headroom' to support higher-performance AMD-K6 processors, with clock speeds scaling to 450 MHz and beyond.

2 Internal Architecture

2.1 Introduction

The Mobile AMD-K6-III-P processor implements advanced design techniques known as the RISC86 microarchitecture. The RISC86 microarchitecture is a decoupled decode/execution design approach that yields superior sixth-generation performance for x86-based software. This chapter describes the techniques used and the functional elements of the RISC86 microarchitecture.

2.2 Mobile AMD-K6®-III-P Processor Microarchitecture Overview

When discussing processor design, it is important to understand the terms *architecture*, *microarchitecture*, and *design implementation*. The term *architecture* refers to the instruction set and features of a processor that are visible to software programs running on the processor. The architecture determines what software the processor can run. The architecture of the Mobile AMD-K6-III-P processor is the industry-standard x86 instruction set.

The term *microarchitecture* refers to the design techniques used in the processor to reach the target cost, performance, and functionality goals. The Mobile AMD-K6 family of processors are based on a sophisticated RISC core known as the Enhanced RISC86 microarchitecture. The Enhanced RISC86 microarchitecture is an advanced, second-order decoupled decode/execution design approach that enables industry-leading performance for x86-based software.

The term *design implementation* refers to the actual logic and circuit designs from which the processor is created according to the microarchitecture specifications.

Enhanced RISC86® Microarchitecture

The Enhanced RISC86 microarchitecture defines the characteristics of the AMD-K6 family. The innovative RISC86 microarchitecture approach implements the x86 instruction set by internally translating x86 instructions into RISC86 operations. These RISC86 operations were specially designed to include direct support for the x86 instruction set while observing the RISC performance principles of fixed length encoding, regularized instruction fields, and a large register set. The Enhanced RISC86 microarchitecture used in the Mobile AMD-K6-III-P processor enables higher processor core performance and promotes straightforward extensions, such as those added in the current Mobile AMD-K6-III-P processor and those planned for the future. Instead of directly executing complex x86 instructions, which have lengths of 1 to 15 bytes, the Mobile AMD-K6-III-P processor executes the simpler and easier fixed-length RISC86 operations, while maintaining the instruction coding efficiencies found in x86 programs.

The Mobile AMD-K6-III-P processor contains parallel decoders, a centralized RISC86 operation scheduler, and ten execution units that support superscalar operation—multiple decode, execution, and retirement—of x86 instructions. These elements are packed into an aggressive and highly efficient six-stage pipeline.

Mobile AMD-K6®-III-P Processor Block Diagram. As shown in Figure 1 on page 7, the high-performance, out-of-order execution engine of the Mobile AMD-K6-III-P processor is mated to a split, level-one, 64-Kbyte, writeback cache with 32 Kbytes of instruction cache and 32 Kbytes of data cache. Backing up the level-one cache is a large, unified, level-two, 256-Kbyte, writeback cache. The level-one instruction cache feeds the decoders and, in turn, the decoders feed the scheduler. The ICU issues and retires RISC86 operations contained in the scheduler. The system bus interface is an industry-standard 64-bit Super7 and Socket 7 demultiplexed bus.

The Mobile AMD-K6-III-P processor combines the latest in processor microarchitecture to provide the highest x86 performance for today's personal computers. The Mobile AMD-K6-III-P processor offers true sixth-generation performance and x86 binary software compatibility.

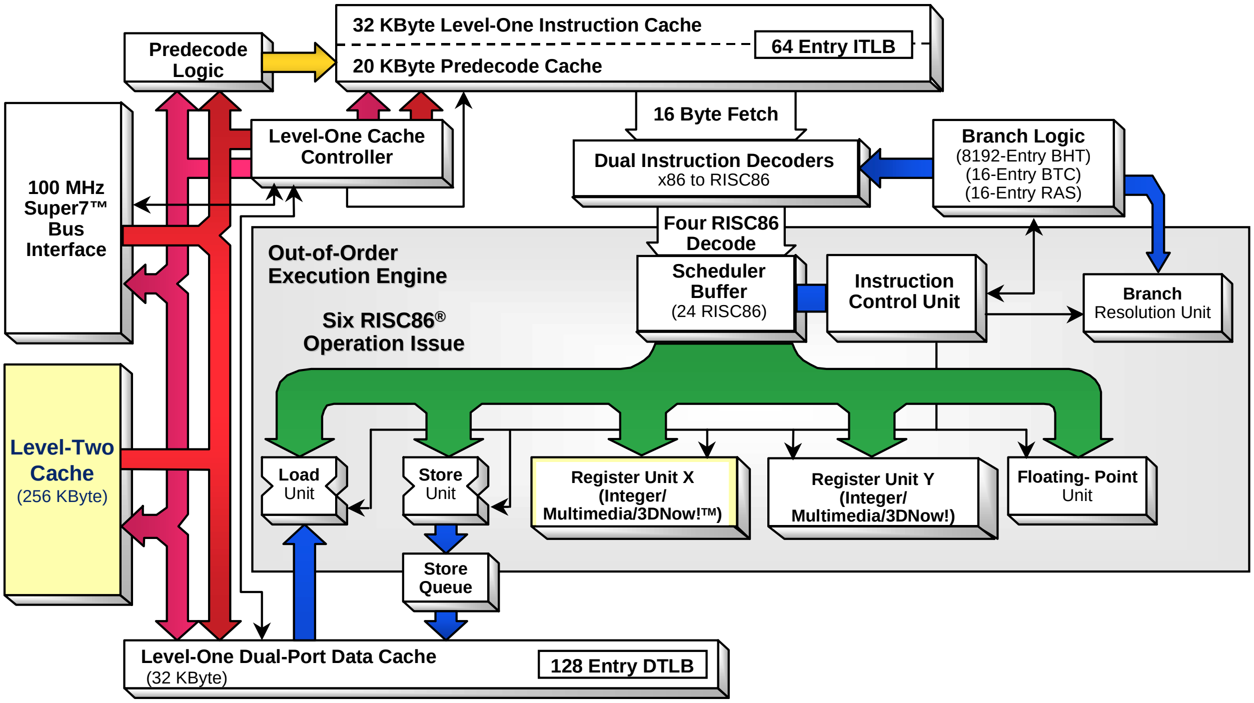


Figure 1. Mobile AMD-K6®-III-P Processor Block Diagram

Decoders. Decoding of the x86 instructions begins when the on-chip level-one instruction cache is filled. Predecode logic determines the length of an x86 instruction on a byte-by-byte basis. This predecode information is stored, along with the x86 instructions, in the level-one instruction cache, to be used later by the decoders. The decoders translate on-the-fly, with no additional latency, up to two x86 instructions per clock into RISC86 operations.

Note: *In this chapter, "clock" refers to a processor clock.*

The Mobile AMD-K6-III-P processor categorizes x86 instructions into three types of decodes—short, long, and vector. The decoders process either two short, one long, or one vector decode at a time. The three types of decodes have the following characteristics:

- Short decodes—x86 instructions less than or equal to seven bytes in length
- Long decodes—x86 instructions less than or equal to 11 bytes in length
- Vector decodes—complex x86 instructions

Short and long decodes are processed completely within the decoders. Vector decodes are started by the decoders and then completed by fetched sequences from an on-chip ROM. After decoding, the RISC86 operations are delivered to the scheduler for dispatching to the executions units.

Scheduler/Instruction Control Unit. The centralized scheduler or buffer is managed by the Instruction Control Unit (ICU). The ICU buffers and manages up to 24 RISC86 operations at a time. This equals from 6 to 12 x86 instructions. This buffer size (24) is perfectly matched to the processor's six-stage RISC86 pipeline and four RISC86-operations decode rate. The scheduler accepts as many as four RISC86 operations at a time from the decoders and retires up to four RISC86 operations per clock cycle. The ICU is capable of simultaneously issuing up to six RISC86 operations at a time to the execution units. This consists of the following types of operations:

- Memory load operation
- Memory store operation
- Complex integer, MMX or 3DNow! register operation
- Simple integer, MMX or 3DNow! register operation
- Floating-point register operation
- Branch condition evaluation

Registers. When managing the 24 RISC86 operations, the ICU uses 69 physical registers contained within the RISC86 microarchitecture. 48 of the physical registers are located in a general register file and are grouped as 24 committed or architectural registers plus 24 rename registers. The 24 architectural registers consist of 16 scratch registers and 8 registers that correspond to the x86 general-purpose registers— EAX, EBX, ECX, EDX, EBP, ESP, ESI, and EDI. There is an analogous set of registers specifically for MMX and 3DNow! operations. There are 9 MMX/3DNow! committed or architectural registers plus 12 MMX/3DNow! rename registers. The 9 architectural registers consist of one scratch register and 8 registers that correspond to the MMX registers (mm0–mm7). For more detailed information, see the *3DNow!™ Technology Manual*, order# 21928.

Branch Logic. The Mobile AMD-K6-III-P processor is designed with highly sophisticated dynamic branch logic consisting of the following:

- Branch history/Prediction table
- Branch target cache
- Return address stack

The Mobile AMD-K6-III-P processor implements a two-level branch prediction scheme based on an 8192-entry branch history table. The branch history table stores prediction information that is used for predicting conditional branches. Because the branch history table does not store predicted target addresses, special address ALUs calculate target addresses on-the-fly during instruction decode. The branch target cache augments predicted branch performance by avoiding a one clock cache-fetch penalty. This specialized target cache does this by supplying the first 16 bytes of target instructions to the decoders when branches are predicted. The return address stack is a unique device specifically designed for optimizing CALL and RETURN pairs. In summary, the Mobile AMD-K6-III-P processor uses dynamic branch logic to minimize delays due to the branch instructions that are common in x86 software.

3DNow!™ Technology. AMD has taken a lead role in improving the multimedia and 3D capabilities of the x86 processor family with the introduction of 3DNow! technology, which uses a packed, single-precision, floating-point data format and Single Instruction Multiple Data (SIMD) operations based on the MMX technology model.

2.3 Cache, Instruction Prefetch, and Predecode Bits

The writeback level-one cache on the Mobile AMD-K6-III-P processor is organized as a separate 32-Kbyte instruction cache and a 32-Kbyte data cache with two-way set associativity. The level-two cache is 256 Kbytes, and is organized as a unified, four-way set-associative cache. The cache line size is 32 bytes, and lines are fetched from external memory using an efficient pipelined burst transaction. As the level-one instruction cache is filled from the level-two cache or from external memory, each instruction byte is analyzed for instruction boundaries using predecoding logic. Predecoding annotates information (5 bits per byte) to each instruction byte that later enables the decoders to efficiently decode multiple instructions simultaneously.

Cache

The processor cache design takes advantage of a sectored organization (see Figure 2). Each sector consists of 64 bytes configured as two 32-byte cache lines. The two cache lines of a sector share a common tag but have separate pairs of MESI (Modified, Exclusive, Shared, Invalid) bits that track the state of each cache line.

Two forms of cache misses and associated cache fills can take place—a tag-miss cache fill and a tag-hit cache fill. In the case of a tag-miss cache fill, the level-one cache miss is due to a tag mismatch, in which case the required cache line is filled either from the level-two cache or from external memory, and the level-one cache line within the sector that was not required is marked as invalid. In the case of a tag-hit cache fill, the address matches the tag, but the requested cache line is marked as invalid. The required level-one cache line is filled from the level-two cache or from external memory, and the level-one cache line within the sector that is not required remains in the same cache state.

Prefetching

The Mobile AMD-K6-III-P processor conditionally performs cache prefetching which results in the filling of the required cache line first, and a prefetch of the second cache line making up the other half of the sector. From the perspective of the external bus, the two cache-line fills typically appear as two 32-byte burst read cycles occurring back-to-back or, if allowed, as pipelined cycles. The 3DNow! technology includes an instruction called PREFETCH that allows a cache line to be prefetched into the level-one data cache and the level-two cache. For more detailed information, see the *3DNow!™ Technology Manual*, order# 21928.

Predecode Bits

Decoding x86 instructions is particularly difficult because the instructions are variable-length and can be from 1 to 15 bytes long. Predecode logic supplies the five predecode bits that are associated with each instruction byte. The predecode bits indicate the number of bytes to the start of the next x86 instruction. The predecode bits are stored in an extended instruction cache alongside each x86 instruction byte as shown in Figure 2. The predecode bits are passed with the instruction bytes to the decoders where they assist with parallel x86 instruction decoding.

Tag Address	Cache Line 0	Byte 31	Predecode Bits	Byte 30	Predecode Bits	....	....	Byte 0	Predecode Bits	MESI Bits
	Cache Line 1	Byte 31	Predecode Bits	Byte 30	Predecode Bits	....	....	Byte 0	Predecode Bits	MESI Bits

Figure 2. Cache Sector Organization

2.4 Instruction Fetch and Decode

Instruction Fetch

The processor can fetch up to 16 bytes per clock out of the level-one instruction cache or branch target cache. The fetched information is placed into a 16-byte instruction buffer that feeds directly into the decoders (see Figure 3). Fetching can occur along a single execution stream with up to seven outstanding branches taken.

The instruction fetch logic is capable of retrieving any 16 contiguous bytes of information within a 32-byte boundary. There is no additional penalty when the 16 bytes of instructions lie across a cache line boundary. The instruction bytes are loaded into the instruction buffer as they are consumed by the decoders. Although instructions can be consumed with byte granularity, the instruction buffer is managed on a memory-aligned word (two bytes) organization. Therefore, instructions are loaded and replaced with word granularity. When a control transfer occurs—such as a JMP instruction—the entire instruction buffer is flushed and reloaded with a new set of 16 instruction bytes.

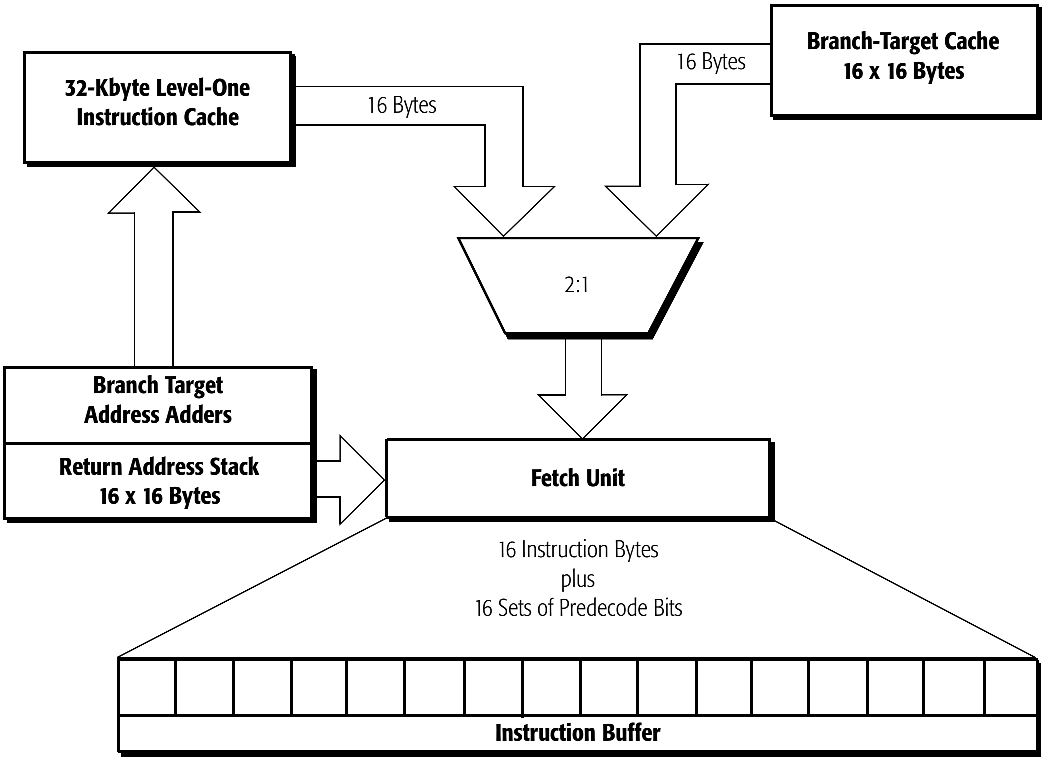


Figure 3. The Instruction Buffer

Instruction Decode

The Mobile AMD-K6-III-P processor decode logic is designed to decode multiple x86 instructions per clock (see Figure 4). The decode logic accepts x86 instruction bytes and their predecode bits from the instruction buffer, locates the actual instruction boundaries, and generates RISC86 operations from these x86 instructions.

RISC86 operations are fixed-length internal instructions. Most RISC86 operations execute in a single clock. RISC86 operations are combined to perform every function of the x86 instruction set. Some x86 instructions are decoded into as few as zero RISC86 operations—for instance a NOP—or one RISC86 operation—a register-to-register add. More complex x86 instructions are decoded into several RISC86 operations.

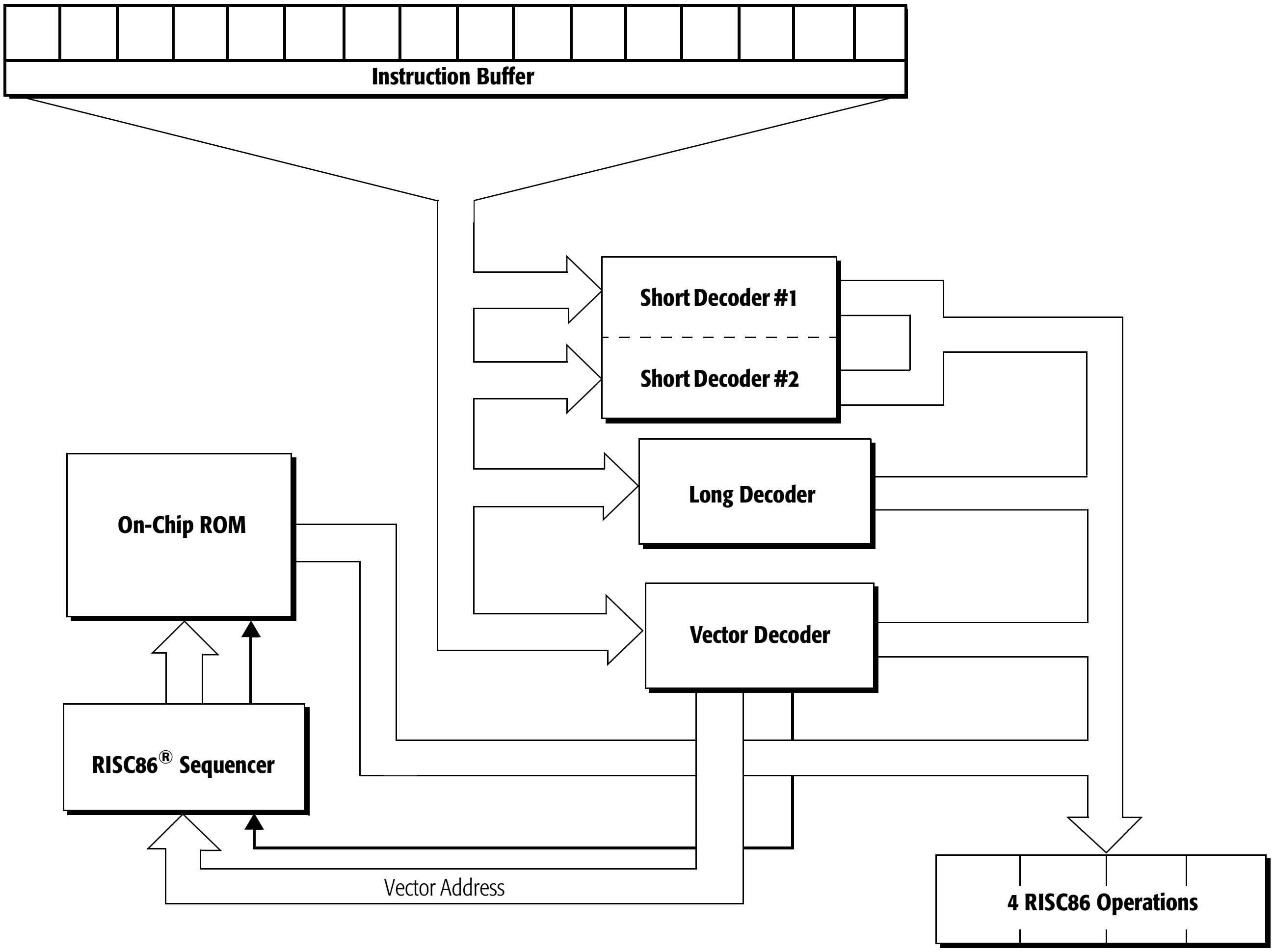


Figure 4. Mobile AMD-K6®-III-P Processor Decode Logic

The Mobile AMD-K6-III-P processor uses a combination of decoders to convert x86 instructions into RISC86 operations. The hardware consists of three sets of decoders—two parallel short decoders, one long decoder, and one vector decoder. The two parallel short decoders translate the most commonly-used x86 instructions (moves, shifts, branches, ALU, FPU) and the extensions to the x86 instruction set (including MMX and 3DNow! instructions) into zero, one, or two RISC86 operations each. The short decoders only operate on x86 instructions that are up to seven bytes long. In addition, they are designed to decode up to two x86 instructions per clock. The commonly-used x86 instructions that are greater than seven bytes but not more than 11 bytes long, and semi-commonly-used x86 instructions that are up to seven bytes long are handled by the long decoder.

The long decoder only performs one decode per clock and generates up to four RISC86 operations. All other translations (complex instructions, serializing conditions, interrupts and exceptions, etc.) are handled by a combination of the vector decoder and RISC86 operation sequences fetched from an on-chip ROM. For complex operations, the vector decoder logic provides the first set of RISC86 operations and a vector (initial ROM address) to a sequence of further RISC86 operations. The same types of RISC86 operations are fetched from the ROM as those that are generated by the hardware decoders.

Note: *Although all three sets of decoders are simultaneously fed a copy of the instruction buffer contents, only one of the three types of decoders is used during any one decode clock.*

The decoders or the on-chip RISC86 ROM always generate a group of four RISC86 operations. For decodes that cannot fill the entire group with four RISC86 operations, RISC86 NOP operations are placed in the empty locations of the grouping. For example, a long-decoded x86 instruction that converts to only three RISC86 operations is padded with a single RISC86 NOP operation and then passed to the scheduler. Up to six groups or 24 RISC86 operations can be placed in the scheduler at a time.

All of the common, and a few of the uncommon, floating-point instructions (also known as ESC instructions) are hardware decoded as short decodes. This decode generates a RISC86 floating-point operation and, optionally, an associated floating-point load or store operation. Floating-point or ESC instruction decode is only allowed in the first short decoder, but non-ESC instructions can be decoded simultaneously by the second short decoder along with an ESC instruction decode in the first short decoder.

All of the MMX and 3DNow! instructions, with the exception of the EMMS, FEMMS, and PREFETCH instructions, are hardware decoded as short decodes. The MMX instruction decode generates a RISC86 MMX operation and, optionally, an associated MMX load or store operation. A 3DNow! instruction decode generates a RISC86 3DNow! operation and, optionally, an associated load or store operation. MMX and 3DNow! instructions can be decoded in either or both of the short decoders.

2.5 Centralized Scheduler

The scheduler is the heart of the Mobile AMD-K6-III-P processor (see Figure 5 on page 15). It contains the logic necessary to manage out-of-order execution, data forwarding, register renaming, simultaneous issue and retirement of multiple RISC86 operations, and speculative execution. The scheduler's buffer can hold up to 24 RISC86 operations. This equates to a maximum of 12 x86 instructions. The scheduler can issue RISC86 operations from any of the 24 locations in the buffer. When possible, the scheduler can simultaneously issue a RISC86 operation to any available execution unit (store, load, branch, register X integer/multimedia, register Y integer/multimedia, or floating-point). In total, the scheduler can issue up to six and retire up to four RISC86 operations per clock.

The main advantage of the scheduler and its operation buffer is the ability to examine an x86 instruction window equal to 12 x86 instructions at one time. This advantage is due to the fact that the scheduler operates on the RISC86 operations in parallel and allows the Mobile AMD-K6-III-P processor to perform dynamic on-the-fly instruction code scheduling for optimized execution. Although the scheduler can issue RISC86 operations for out-of-order execution, it always retires x86 instructions in order.

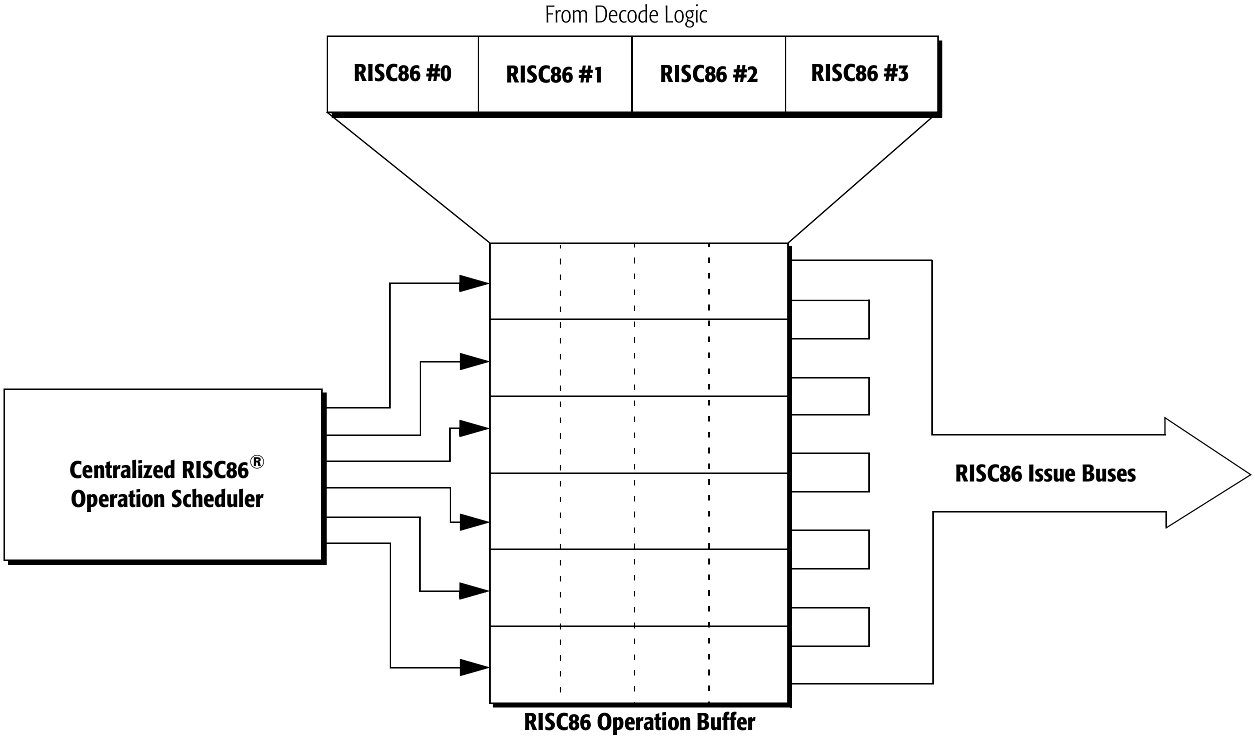


Figure 5. Mobile AMD-K6®-III-P Processor Scheduler

2.6 Execution Units

The Mobile AMD-K6-III-P processor contains ten parallel execution units—store, load, integer X ALU, integer Y ALU, MMX ALU (X), MMX ALU (Y), MMX/3DNow! multiplier, 3DNow! ALU, floating-point, and branch condition. Each unit is independent and capable of handling the RISC86 operations. Table 1 on page 16 details the execution units, functions performed within these units, operation latency, and operation throughput.

The store and load execution units are two-stage pipelined designs. The store unit performs data writes and register calculation for LEA/PUSH. Data memory and register writes from stores are available after one clock. Store operations are held in a store queue prior to execution. From there, they execute in order. The load unit performs data memory reads. Data is available from the load unit after two clocks.

The Integer X execution unit can operate on all ALU operations, multiplies, divides (signed and unsigned), shifts, and rotates.

The Integer Y execution unit can operate on the basic word and doubleword ALU operations—ADD, AND, CMP, OR, SUB, XOR, zero-extend and sign-extend operands.

Table 1. Execution Latency and Throughput of Execution Units

Functional Unit	Function	Latency	Throughput
Store	LEA/PUSH, Address (Pipelined)	1	1
	Memory Store (Pipelined)	1	1
Load	Memory Loads (Pipelined)	2	1
Integer X	Integer ALU	1	1
	Integer Multiply	2–3	2–3
	Integer Shift	1	1
Multimedia (processes MMX instructions)	MMX ALU	1	1
	MMX Shifts, Packs, Unpack	1	1
	MMX Multiply	2	1
Integer Y	Basic ALU (16-bit and 32-bit operands)	1	1
Branch	Resolves Branch Conditions	1	1
FPU	FADD, FSUB, FMUL	2	2
3DNow!	3DNow! ALU	2	1
	3DNow! Multiply	2	1
	3DNow! Convert	2	1

Register X and Y Pipelines

The functional units that execute MMX and 3DNow! instructions share pipeline control with the Integer X and Integer Y units.

The register X and Y functional units are attached to the issue bus for the register X execution pipeline or the issue bus for the register Y execution pipeline or both. Each register pipeline has dedicated resources that consist of an integer execution unit and an MMX ALU execution unit, therefore allowing superscalar operation on integer and MMX instructions. In addition, both the X and Y issue buses are connected to the 3DNow! ALU, the MMX/3DNow! multiplier and MMX shifter, which allows the appropriate RISC86 operation to be issued through either bus. Figure 6 on page 17 shows the details of the X and Y register pipelines.

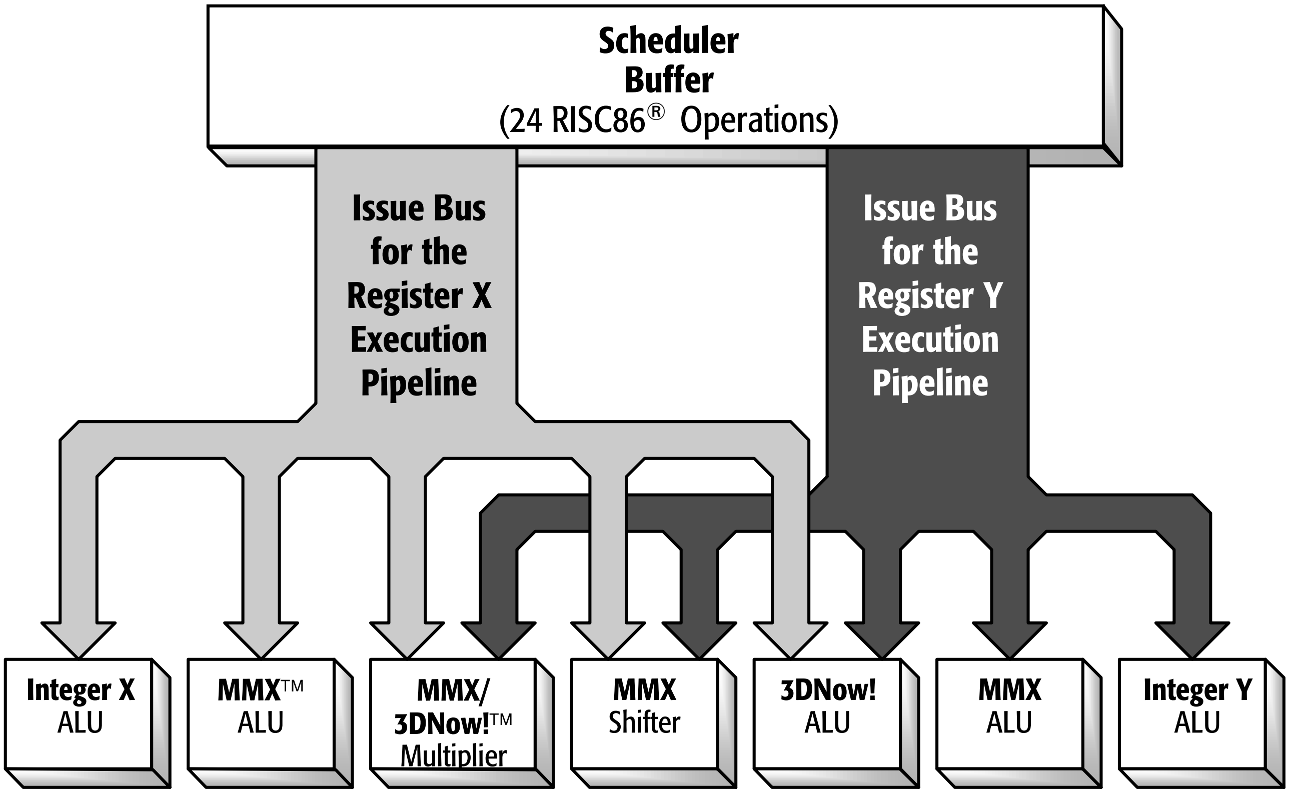


Figure 6. Register X and Y Functional Units

The branch condition unit is separate from the branch prediction logic in that it resolves conditional branches such as JCC and LOOP after the branch condition has been evaluated.

2.7 Branch-Prediction Logic

Sophisticated branch logic that can minimize or hide the impact of changes in program flow is designed into the Mobile AMD-K6-III-P processor. Branches in x86 code fit into two categories—unconditional branches, which always change program flow (that is, the branches are always taken) and conditional branches, which may or may not divert program flow (that is, the branches are taken or not-taken). When a conditional branch is not taken, the processor simply continues decoding and executing the next instructions in memory.

Typical applications have up to 10% of unconditional branches and another 10% to 20% conditional branches. The Mobile AMD-K6-III-P processor branch logic has been designed to handle this type of program behavior and its negative effects on instruction execution, such as stalls due to delayed instruction fetching and the draining of the processor pipeline. The branch logic contains an 8192-entry branch history table, a 16-entry by 16-byte branch target cache, a 16-entry return address stack, and a branch execution unit.

Branch History Table

The Mobile AMD-K6-III-P processor handles unconditional branches without any penalty by redirecting instruction fetching to the target address of the unconditional branch. However, conditional branches require the use of the dynamic branch-prediction mechanism built into the Mobile AMD-K6-III-P processor. A two-level adaptive history algorithm is implemented in an 8192-entry branch history table. This table stores executed branch information, predicts individual branches, and predicts the behavior of groups of branches. To accommodate the large branch history table, the Mobile AMD-K6-III-P processor does not store predicted target addresses. Instead, the branch target addresses are calculated on-the-fly using ALUs during the decode stage. The adders calculate all possible target addresses before the instructions are fully decoded and the processor chooses which addresses are valid.

Branch Target Cache

To avoid a one clock cache-fetch penalty when a branch is predicted taken, a built-in branch target cache supplies the first 16 bytes of instructions directly to the instruction buffer (assuming the target address hits this cache). (See Figure 3 on page 11.) The branch target cache is organized as 16 entries of 16 bytes. In total, the branch prediction logic achieves branch prediction rates greater than 95%.

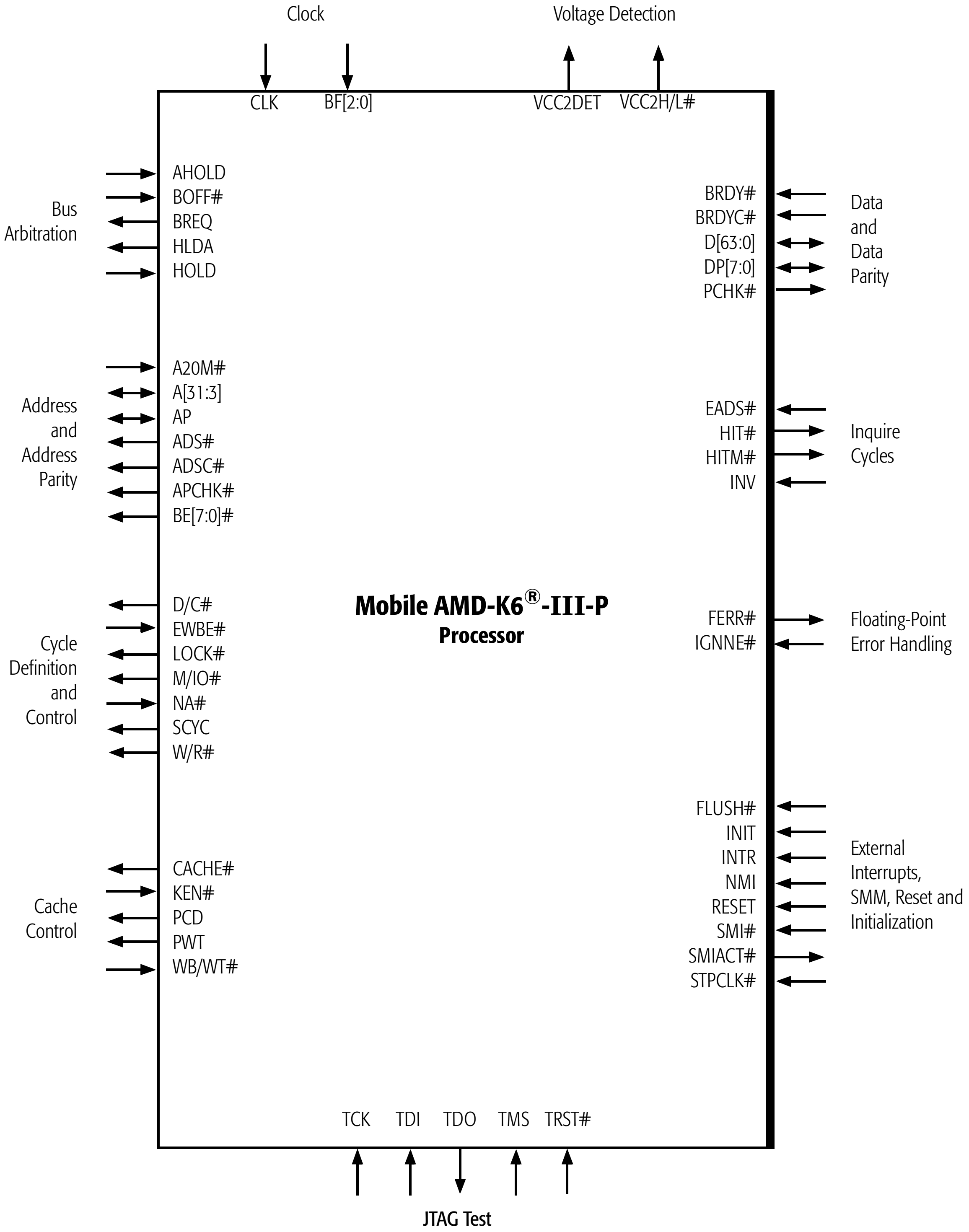
Return Address Stack

The return address stack is a special device designed to optimize CALL and RET pairs. Software is typically compiled with subroutines that are frequently called from various places in a program. This is usually done to save space. Entry into the subroutine occurs with the execution of a CALL instruction. At that time, the processor pushes the address of the next instruction in memory following the CALL instruction onto the stack (allocated space in memory). When the processor encounters a RET instruction (within or at the end of the subroutine), the branch logic pops the address from the stack and begins fetching from that location. To avoid the latency of main memory accesses during CALL and RET operations, the return address stack caches the pushed addresses.

Branch Execution Unit

The branch execution unit enables efficient speculative execution. This unit gives the processor the ability to execute instructions beyond conditional branches before knowing whether the branch prediction was correct. The Mobile AMD-K6-III-P processor does not permanently update the x86 registers or memory locations until all speculatively executed conditional branch instructions are resolved. When a prediction is incorrect, the processor backs out to the point of the mispredicted branch instruction and restores all registers. The Mobile AMD-K6-III-P processor can support up to seven outstanding branches.

3 Logic Symbol Diagram



4 Signal Descriptions

Signal Name	Pin Location	Pin Attribute	Name and Summary
A20M#	AK-08	Input	*Address Bit 20 Mask* A20M# is used to simulate the behavior of the 8086 when it is running in Real mode. The assertion of A20M# causes the processor to force bit 20 of the physical address to 0 prior to accessing the cache or driving out a memory bus cycle. The clearing of address bit 20 maps addresses that wrap above 1 Mbyte to addresses below 1 Mbyte.
A[31:3]	See "Pin Designations by Functional Grouping" on page 85.	A31-A5: Bidirectional A4-A3: Output	*Address Bus* A[31:3] contains the physical address for the current bus cycle. The processor drives addresses on A[31:3] during memory and I/O cycles, and cycle definition information during special bus cycles. The processor samples addresses on A[31:5] during inquire cycles.
ADS#	AJ-05	Output	*Address Strobe* The assertion of ADS# indicates the beginning of a new bus cycle. The address bus and all cycle definition signals corresponding to this bus cycle are driven valid off the same clock edge as ADS#.
ADSC#	AM-02	Output	*Address Strobe Copy* ADSC# has the identical function and timing as ADS#. In the event ADS# becomes too heavily loaded due to a large fanout in a system, ADSC# can be used to split the load across two outputs, which improves timing.
AHOLD	V-04	Input	*Address Hold* AHOLD can be asserted by the system to initiate one or more inquire cycles. To allow the system to drive the address bus during an inquire cycle, the processor floats A[31:3] and AP off the clock edge on which AHOLD is sampled asserted. The data bus and all other control and status signals remain under the control of the processor and are not floated.
AP	AK-02	Bidirectional	*Address Parity* AP contains the even parity bit for cache line addresses driven and sampled on A[31:5]. The term *even parity* means that the total number of 1 bits on AP and A[31:5] is even. (A4 and A3 are not used for the generation or checking of address parity because these bits are not required to address a cache line.)

Signal Name	Pin Location	Pin Attribute	Name and Summary
APCHK#	AE-05	Output	*Address Parity Check* If the processor detects an address parity error during an inquire cycle, APCHK# is asserted for one clock.
BE[7:0]#	See "Pin Designations by Functional Grouping" on page 85.	Output	*Byte Enables* BE[7:0]# are used by the processor to indicate the valid data bytes during a write cycle and the requested data bytes during a read cycle. The eight byte enables correspond to the eight bytes of the data bus as follows: ■ BE7#: D[63:56] ■ BE6#: D[55:48] ■ BE5#: D[47:40] ■ BE4#: D[39:32] ■ BE3#: D[31:24] ■ BE2#: D[23:16] ■ BE1#: D[15:8] ■ BE0#: D[7:0] The byte enables are also used to distinguish between special bus cycles as defined in Table 7 on page 34.
BF[2:0]	See "Pin Designations by Functional Grouping" on page 85.	Inputs, Internal Pullups	*Bus Frequency* BF[2:0] determine the internal operating frequency of the processor. The frequency of the CLK input signal is multiplied internally by a ratio determined by the state of these signals as shown below: State of BF[2:0] Inputs — Processor-Clock to Bus-Clock Ratio 100b — 2.5x 101b — 3.0x 111b — 3.5x 010b — 4.0x 000b — 4.5x 001b — 5.0x 011b — 5.5x 110b — 6.0x BF[2:0] have weak internal pullups and default to the 3.5 ratio if left unconnected.

Signal Name	Pin Location	Pin Attribute	Name and Summary
BOFF#	Z-04	Input	*Backoff* If BOFF# is sampled asserted, the processor unconditionally aborts any cycles in progress and transitions to a bus hold state by floating the following signals: A[31:3], ADS#, ADSC#, AP, BE[7:0]#, CACHE#, D[63:0], D/C#, DP[7:0], LOCK#, M/IO#, PCD, PWT, SCYC, and W/R#. These signals remain floated until BOFF# is sampled negated. This allows an alternate bus master or the system to control the bus.
BRDY#	X-04	Input, Internal Pullup	*Burst Ready* BRDY# is asserted to the processor by system logic to indicate either that the data bus is being driven with valid data during a read cycle or that the data bus has been latched during a write cycle. BRDY# is also used to indicate the completion of special bus cycles.
BRDYC#	Y-03	Input, Internal Pullup	*Burst Ready Copy* BRDYC# has the identical function as BRDY#. In the event BRDY# becomes too heavily loaded due to a large fanout in a system, BRDYC# can be used to reduce this loading, which improves timing.
BREQ	AJ-01	Output	*Bus Request* BREQ is asserted by the processor to request the bus in order to complete an internally pending bus cycle. The system logic can use BREQ to arbitrate among the bus participants.
CACHE#	U-03	Output	*Cacheable Access* For reads, CACHE# is asserted to indicate the cacheability of the current bus cycle. For write cycles, CACHE# is asserted to indicate the current bus cycle is a modified cache-line writeback.
CLK	AK-18	Input	*Clock* The CLK signal is the bus clock for the processor and is the reference for all signal timings under normal operation.
D/C#	AK-04	Output	*Data/Code* The processor drives D/C# during a memory bus cycle to indicate whether it is addressing data or executable code. D/C# is also used to define other bus cycles, including interrupt acknowledge and special cycles.
D[63:0]	See "Pin Designations by Functional Grouping" on page 85.	Bidirectional	*Data Bus* D[63:0] represent the processor's 64-bit data bus. Each of the eight bytes of data that comprise this bus is qualified by a corresponding byte enable.

Signal Name	Pin Location	Pin Attribute	Name and Summary
DP[7:0]	See "Pin Designations by Functional Grouping" on page 85.	Bidirectional	*Data Parity* DP[7:0] are even parity bits for each valid byte of data—as defined by BE[7:0]#—driven and sampled on the D[63:0] data bus. If the processor detects bad parity on any valid byte of data during a read cycle, PCHK# is asserted. The eight data parity bits correspond to the eight bytes of the data bus as follows: ■ DP7: D[63:56] ■ DP6: D[55:48] ■ DP5: D[47:40] ■ DP4: D[39:32] ■ DP3: D[31:24] ■ DP2: D[23:16] ■ DP1: D[15:8] ■ DP0: D[7:0] For systems that do not support data parity, DP[7:0] should be connected to V_{CC3} through pullup resistors.
EADS#	AM-04	Input	*External Address Strobe* System logic asserts EADS# during a cache inquire cycle to indicate that the address bus contains a valid address.
EWBE#	W-03	Input	*External Write Buffer Empty* The system logic can negate EWBE# to the processor to indicate that its external write buffers are full and that additional data cannot be stored at this time. This causes the processor to delay the following activities until EWBE# is sampled asserted: ■ The commitment of write hit cycles to cache lines in the modified state or exclusive state in the processor's cache ■ The decode and execution of an instruction that follows a currently-executing serializing instruction ■ The assertion or negation of SMIACT# ■ The entering of the Halt state and the Stop Grant state
FERR#	Q-05	Output	*Floating-Point Error* The assertion of FERR# indicates the occurrence of an unmasked floating-point exception resulting from the execution of a floating-point instruction.

Signal Name	Pin Location	Pin Attribute	Name and Summary
FLUSH#	AN-07	Input	*Cache Flush* In response to sampling FLUSH# asserted, the processor writes back any cache lines in the L1 data cache or L2 cache that are in the modified state, invalidates all lines in the L1 and L2 caches, and then executes a flush acknowledge special cycle. In addition, FLUSH# is sampled when RESET is negated to determine if the processor enters Tri-State Test mode.
HIT#	AK-06	Output	*Inquire Cycle Hit* The processor asserts HIT# during an inquire cycle to indicate that the cache line is valid within the processor's L1 and/or L2 caches (also known as a cache hit).
HITM#	AL-05	Output	*Inquire Cycle Hit To Modified Line* The processor asserts HITM# during an inquire cycle to indicate that the cache line exists in the processor's L1 data cache or L2 cache in the modified state. The processor performs a writeback cycle as a result of this cache hit.
HLDA	AJ-03	Output	*Hold Acknowledge* When HOLD is sampled asserted, the processor completes the current bus cycles, floats the processor bus, and asserts HLDA in an acknowledgment that these events have been completed. The following signals are floated when HLDA is asserted: A[31:3], ADS#, ADSC#, AP, BE[7:0]#, CACHE#, D[63:0], D/C#, DP[7:0], LOCK#, M/IO#, PCD, PWT, SCYC, and W/R#.
HOLD	AB-04	Input	*Bus Hold Request* The system logic can assert HOLD to gain control of the processor's bus. When HOLD is sampled asserted, the processor completes the current bus cycles, floats the processor bus, and asserts HLDA in an acknowledgment that these events have been completed.
IGNNE#	AA-35	Input	*Ignore Numeric Exception* IGNNE# is used by external logic to control the effect of an unmasked floating-point exception. Under certain circumstances, if IGNNE# is sampled asserted, the processor ignores the floating-point exception.

Signal Name	Pin Location	Pin Attribute	Name and Summary
INIT	AA-33	Input	*Initialization* The assertion of INIT causes the processor to flush its pipelines, to initialize most of its internal state, and to branch to address FFFF_FFF0h—the same instruction execution starting point used after RESET. Unlike RESET, the processor preserves the contents of its caches, the floating-point state, the MMX state, model-specific registers, the CD and NW bits of the CR0 register, and other specific internal resources.
INTR	AD-34	Input	*Maskable Interrupt* INTR is the system's maskable interrupt input to the processor. When the processor samples and recognizes INTR asserted, the processor executes a pair of interrupt acknowledge bus cycles and then jumps to the interrupt service routine specified by the interrupt number that was returned during the interrupt acknowledge sequence.
INV	U-05	Input	*Invalidation Request* During an inquire cycle, the state of INV determines whether an addressed cache line that is found in the processor's L1 and/or L2 caches transitions to the *invalid* state or the *shared* state.
KEN#	W-05	Input	*Cache Enable* If KEN# is sampled asserted, it indicates that the address presented by the processor is cacheable. Otherwise, a single-transfer cycle is executed and the processor does not cache the data. KEN# is ignored during writebacks.
LOCK#	AH-04	Output	*Bus Lock* The processor asserts LOCK# during a sequence of bus cycles to ensure that the cycles are completed without allowing other bus masters to intervene.
M/IO#	T-04	Output	*Memory or I/O* The processor drives M/IO# during a bus cycle to indicate whether it is addressing the memory or I/O space. M/IO# is used to define other bus cycles, including interrupt acknowledge and special cycles.
NA#	Y-05	Input	*Next Address* System logic asserts NA# to indicate to the processor that it is ready to accept another address pipelined into the previous bus cycle.

Signal Name	Pin Location	Pin Attribute	Name and Summary
NMI	AC-33	Input	*Non-Maskable Interrupt* When NMI is sampled asserted, the processor jumps to the interrupt service routine defined by interrupt number 02h. Unlike the INTR signal, software cannot mask the effect of NMI if it is sampled asserted by the processor.
PCD	AG-05	Output	*Page Cache Disable* The processor drives PCD to indicate the operating system's specification of cacheability for the page being addressed. System logic can use PCD to control external caching.
PCHK#	AF-04	Output	*Parity Check* The processor asserts PCHK# during read cycles if it detects an even parity error on one or more valid bytes of D[63:0] during a read cycle.
PWT	AL-03	Output	*Page Writethrough* The processor drives PWT to indicate the operating system's specification of the *writeback state* or *writethrough* state for the page being addressed. PWT, together with WB/WT#, specifies the data cache-line state during cacheable read misses and write hits to shared cache lines.
RESET	AK-20	Input	*Reset* When the processor samples RESET asserted, it immediately flushes and initializes all internal resources and its internal state including its pipelines and caches, the floating-point state, the MMX state, and all registers, and then the processor jumps to address FFFF_FFF0h to start instruction execution. FLUSH# is sampled during the falling transition of RESET to invoke the Tri-State Test mode.
RSVD	See "Pin Designations by Functional Grouping" on page 85.	–	*Reserved* Reserved signals are a special class of pins that can be treated in one of the following ways: ■ As no-connect (NC) pins, in which case these pins are left unconnected ■ As pins connected to the system logic as defined by the industry-standard Super7 and Socket 7 interface ■ Any combination of NC and Socket 7 pins
SCYC	AL-17	Output	*Split Cycle* The processor asserts SCYC during misaligned, locked transfers on the D[63:0] data bus.

Signal Name	Pin Location	Pin Attribute	Name and Summary
SMI#	AB-34	Input, Internal Pullup	*System Management Interrupt* The assertion of SMI# causes the processor to enter System Management Mode (SMM). Upon recognizing SMI#, the processor performs the following actions, in the order shown: 1. Flushes its instruction pipelines. 2. Completes all pending and in-progress bus cycles. 3. Acknowledges the interrupt by asserting SMIACT# after sampling EWBE# asserted (if EWBE# is masked off, then SMIACT# is not affected by EWBE#). 4. Saves the internal processor state in SMM memory. 5. Disables interrupts. 6. Jumps to the entry point of the SMM service routine.
SMIACT#	AG-03	Output	*System Management Interrupt Active* The processor acknowledges the assertion of SMI# with the assertion of SMIACT# to indicate that the processor has entered System Management Mode (SMM).
STPCLK#	V-34	Input, Internal Pullup	*Stop Clock* The assertion of STPCLK# causes the processor to enter the Stop Grant state, during which the processor's internal clock is stopped. From the Stop Grant state, the processor can subsequently transition to the Stop Clock state, in which the bus clock CLK is stopped. Upon recognizing STPCLK#, the processor performs the following actions, in the order shown: 1. Flushes its instruction pipelines. 2. Completes all pending and in-progress bus cycles. 3. Acknowledges the STPCLK# assertion by executing a Stop Grant special bus cycle (see Table 7 on page 34). 4. Stops its internal clock after BRDY# of the Stop Grant special bus cycle is sampled asserted and after EWBE# is sampled asserted (if EWBE# is masked off, then entry into the Stop Grant state is not affected by EWBE#). 5. Enters the Stop Clock state if the system logic stops the bus clock CLK (optional).
TCK	M-34	Input, Internal Pullup	*Test Clock* TCK is the clock for boundary-scan testing using the Test Access Port (TAP).

Signal Name	Pin Location	Pin Attribute	Name and Summary
TDI	N-35	Input, Internal Pullup	*Test Data Input* TDI is the serial test data and instruction input for boundary-scan testing using the Test Access Port (TAP).
TDO	N-33	Output	*Test Data Output* TDO is the serial test data and instruction output for boundary-scan testing using the Test Access Port (TAP).
TMS	P-34	Input, Internal Pullup	*Test Mode Select* TMS specifies the test function and sequence of state changes for boundary-scan testing using the Test Access Port (TAP).
TRST#	Q-33	Input, Internal Pullup	*Test Reset* The assertion of TRST# initializes the Test Access Port (TAP) by resetting its state machine to the Test-Logic-Reset state.
VCC2DET	AL-01	Output	*VCC2 Detect* VCC2DET is tied to V_{SS} (logic level 0) to indicate to the system logic that it must supply the specified dual-voltage requirements to the V_{CC2} and V_{CC3} pins.
VCC2H/L#	AN-05	Output	*VCC2 High/Low* VCC2H/L# is tied to V_{SS} (logic level 0) to indicate to the system logic that it must supply the specified processor core voltage to the V_{CC2} pins.
W/R#	AM-06	Output	*Write/Read* The processor drives W/R# to indicate whether it is performing a write or a read cycle on the bus. In addition, W/R# is used to define other bus cycles, including interrupt acknowledge and special cycles.
WB/WT#	AA-05	Input	*Writeback or Writethrough* WB/WT#, together with PWT, specifies the data cache-line state during cacheable read misses and write hits to shared cache lines.

Table 2. Input Pin Types

Name	Type	Note	Name	Type	Note
A20M#	Asynchronous	Note 1	IGNNE#	Asynchronous	Note 1
AHOLD	Synchronous		INIT	Asynchronous	Note 2
BF[2:0]	Synchronous	Note 4	INTR	Asynchronous	Note 1
BOFF#	Synchronous		INV	Synchronous	
BRDY#	Synchronous		KEN#	Synchronous	
BRDYC#	Synchronous		NA#	Synchronous	
CLK	Clock		NMI	Asynchronous	Note 2
EADS#	Synchronous		RESET	Asynchronous	Note 5, 6
EWBE#	Synchronous		SMI#	Asynchronous	Note 2
FLUSH#	Asynchronous	Note 2, 3	STPCLK#	Asynchronous	Note 1
HOLD	Synchronous		WB/WT#	Synchronous	

Notes:

1. *These level-sensitive signals can be asserted synchronously or asynchronously. To be sampled on a specific clock edge, setup and hold times must be met. If asserted asynchronously, they must be asserted for a minimum pulse width of two clocks.*
2. *These edge-sensitive signals can be asserted synchronously or asynchronously. To be sampled on a specific clock edge, setup and hold times must be met. If asserted asynchronously, they must have been negated at least two clocks prior to assertion and must remain asserted at least two clocks.*
3. *FLUSH# is also sampled during the falling transition of RESET and can be asserted synchronously or asynchronously. To be sampled on a specific clock edge, setup and hold times must be met the clock edge before the clock edge on which RESET is sampled negated. If asserted asynchronously, FLUSH# must meet a minimum setup and hold time of two clocks relative to the negation of RESET.*
4. *BF[2:0] are sampled during the falling transition of RESET. They must meet a minimum setup time of 1.0 ms and a minimum hold time of two clocks relative to the negation of RESET.*
5. *During the initial power-on reset of the processor, RESET must remain asserted for a minimum of 1.0 ms after CLK and V_{CC} reach specification before it is negated.*
6. *During a warm reset, while CLK and V_{CC} are within their specification, RESET must remain asserted for a minimum of 15 clocks prior to its negation.*

Table 3. Output Pin Float Conditions

Name	Floated At: (Note 1)	Note	Name	Floated At: (Note 1)	Note
A[4:3]	HLDA, AHOLD, BOFF#	Note 2,3	HLDA	Always Driven	
ADS#	HLDA, BOFF#	Note 2	LOCK#	HLDA, BOFF#	Note 2
ADSC#	HLDA, BOFF#	Note 2	M/IO#	HLDA, BOFF#	Note 2
APCHK#	Always Driven		PCD	HLDA, BOFF#	Note 2
BE[7:0]#	HLDA, BOFF#	Note 2	PCHK#	Always Driven	
BREQ	Always Driven		PWT	HLDA, BOFF#	Note 2
CACHE#	HLDA, BOFF#	Note 2	SCYC	HLDA, BOFF#	Note 2
D/C#	HLDA, BOFF#	Note 2	SMIACT#	Always Driven	
FERR#	Always Driven		VCC2DET	Always Driven	
HIT#	Always Driven		VCC2H/L#	Always Driven	
HITM#	Always Driven		W/R#	HLDA, BOFF#	Note 2

Notes:

1. *All outputs except VCC2DET, VCC2H/L#, and TDO float during Tri-State Test mode.*
2. *Floated off the clock edge that BOFF# is sampled asserted and off the clock edge that HLDA is asserted.*
3. *Floated off the clock edge that AHOLD is sampled asserted.*

Table 4. Input/Output Pin Float Conditions

Name	Floated At: (Note 1)	Note
A[31:5]	HLDA, AHOLD, BOFF#	Note 2,3
AP	HLDA, AHOLD, BOFF#	Note 2,3
D[63:0]	HLDA, BOFF#	Note 2
DP[7:0]	HLDA, BOFF#	Note 2

Notes:

1. *All outputs except VCC2DET and TDO float during Tri-State Test mode.*
2. *Floated off the clock edge that BOFF# is sampled asserted and off the clock edge that HLDA is asserted.*
3. *Floated off the clock edge that AHOLD is sampled asserted.*

Table 5. Test Pins

Name	Type	Note
TCK	Clock	
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
TRST#	Input	Asynchronous (Independent of TCK)

Table 6. Bus Cycle Definition

Bus Cycle Initiated	Generated by CPU				Generated by System Logic
	M/IO#	D/C#	W/R#	CACHE#	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	0	1	0	1	x
I/O Write	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

Note:
x means "don't care"

Table 7. Special Cycles

Special Cycle	A4	BE7#	BE6#	BE5#	BE4#	BE3#	BE2#	BE1#	BE0#	M/IO#	D/C#	W/R#	CACHE#	KEN#
Stop Grant	1	1	1	1	1	1	0	1	1	0	0	1	1	x
Flush Acknowledge (FLUSH# sampled asserted)	0	1	1	1	0	1	1	1	1	0	0	1	1	x
Writeback (WBINVD instruction)	0	1	1	1	1	0	1	1	1	0	0	1	1	x
Halt	0	1	1	1	1	1	0	1	1	0	0	1	1	x
Flush (INVD, WBINVD instruction)	0	1	1	1	1	1	1	0	1	0	0	1	1	x
Shutdown	0	1	1	1	1	1	1	1	0	0	0	1	1	x

Note:
x means "don't care"

5 Mobile AMD-K6®-III-P Processor Operation

5.1 Process Technology

The Mobile AMD-K6-III-P processor is implemented using an advanced CMOS process technology that utilizes a split core and I/O voltage supply, which allows the core of the processor to operate at a low voltage while the I/O portion operates at the industry-standard 3.3 volts. This technology enables high performance while reducing power consumption by operating the core at a low voltage and limiting power requirements to the acceptable levels for today's mobile PCs.

5.2 Clock Control

The Mobile AMD-K6-III-P processor supports five modes of clock control. The processor can transition between these modes to maximize performance, to minimize power dissipation, or to provide a balance between performance and power. (See "Power Dissipation" on page 73 for the maximum power dissipation of the Mobile AMD-K6-III-P within the normal and reduced-power states.)

The five clock-control states supported are as follows:

- **Normal State**: The processor is running in Real Mode, Virtual-8086 Mode, Protected Mode, or System Management Mode (SMM). In this state, all clocks are running—including the external bus clock CLK and the internal processor clock—and the full features and functions of the processor are available.
- **Halt State**: This low-power state is entered following the successful execution of the HLT instruction. During this state, the internal processor clock is stopped.
- **Stop Grant State**: This low-power state is entered following the recognition of the assertion of the STPCLK# signal. During this state, the internal processor clock is stopped.
- **Stop Grant Inquire State**: This state is entered from the Halt state and the Stop Grant state as the result of a system-initiated inquire cycle.
- **Stop Clock State**: This low-power state is entered from the Stop Grant state when the CLK signal is stopped.

The following sections describe each of the four low-power states. Figure 7 on page 39 illustrates the clock control state transitions.

Halt State

Enter Halt State. During the execution of the HLT instruction, the Mobile AMD-K6-III-P processor executes a Halt special cycle. After BRDY# is sampled asserted during this cycle, and then EWBE# is also sampled asserted (if not masked off), the processor enters the Halt state in which the processor disables most of its internal clock distribution. In order to support the following operations, the internal phase-lock loop (PLL) continues to run, and some internal resources are still clocked in the Halt state:

- Inquire Cycles: The processor continues to sample AHOLD, BOFF#, and HOLD in order to support inquire cycles that are initiated by the system logic. The processor transitions to the Stop Grant Inquire state during the inquire cycle. After returning to the Halt state following the inquire cycle, the processor does not execute another Halt special cycle.
- Flush Cycles: The processor continues to sample FLUSH#. If FLUSH# is sampled asserted, the processor performs the flush operation in the same manner as it is performed in the Normal state. Upon completing the flush operation, the processor executes the Halt special cycle which indicates the processor is in the Halt state.
- Time Stamp Counter (TSC): The TSC continues to count in the Halt state.
- Signal Sampling: The processor continues to sample INIT, INTR, NMI, RESET, and SMI#.

After entering the Halt state, all signals driven by the processor retain their state as they existed following the completion of the Halt special cycle.

Exit Halt State. The Mobile AMD-K6-III-P processor remains in the Halt state until it samples INIT, INTR (if interrupts are enabled), NMI, RESET, or SMI# asserted. If any of these signals is sampled asserted, the processor returns to the Normal state and performs the corresponding operation. All of the normal requirements for recognition of these input signals apply within the Halt state.

Stop Grant State

Enter Stop Grant State. After recognizing the assertion of STPCLK#, the Mobile AMD-K6-III-P processor flushes its instruction pipelines, completes all pending and in-progress bus cycles, and acknowledges the STPCLK# assertion by executing a Stop Grant special bus cycle. After BRDY# is sampled asserted during this cycle, and after EWBE# is also sampled asserted (if not masked off), the processor enters the Stop Grant state. The Stop Grant state is like the Halt state in that the processor disables most of its internal clock distribution in the Stop Grant state. In order to support the following operations, the internal PLL still runs, and some internal resources are still clocked in the Stop Grant state:

- Inquire cycles: The processor transitions to the Stop Grant Inquire state during an inquire cycle. After returning to the Stop Grant state following the inquire cycle, the processor does not execute another Stop Grant special cycle.
- Time Stamp Counter (TSC): The TSC continues to count in the Stop Grant state.
- Signal Sampling: The processor continues to sample INIT, INTR, NMI, RESET, and SMI#.

FLUSH# is not recognized in the Stop Grant state (unlike while in the Halt state).

Upon entering the Stop Grant state, all signals driven by the processor retain their state as they existed following the completion of the Stop Grant special cycle.

Exit Stop Grant State. The Mobile AMD-K6-III-P processor remains in the Stop Grant state until it samples STPCLK# negated or RESET asserted. If STPCLK# is sampled negated, the processor returns to the Normal state in less than 10 bus clock (CLK) periods. After the transition to the Normal state, the processor resumes execution at the instruction boundary on which STPCLK# was initially recognized.

If STPCLK# is recognized as negated in the Stop Grant state and subsequently sampled asserted prior to returning to the Normal state, a minimum of one instruction is executed prior to re-entering the Stop Grant state.

If INIT, INTR (if interrupts are enabled), FLUSH#, NMI, or SMI# are sampled asserted in the Stop Grant state, the processor latches the edge-sensitive signals (INIT, FLUSH#, NMI, and SMI#), but otherwise does not exit the Stop Grant state to service the interrupt. When the processor returns to the Normal state due to sampling STPCLK# negated, any pending interrupts are recognized after returning to the Normal state. To ensure their recognition, all of the normal requirements for these input signals apply within the Stop Grant state.

If RESET is sampled asserted in the Stop Grant state, the processor immediately returns to the Normal state and the reset process begins.

Stop Grant Inquire State

Enter Stop Grant Inquire State. The Stop Grant Inquire state is entered from the Stop Grant state or the Halt state when EADS# is sampled asserted during an inquire cycle initiated by the system logic. The Mobile AMD-K6-III-P processor responds to an inquire cycle in the same manner as in the Normal state by driving HIT# and HITM#. If the inquire cycle hits a modified cache line, the processor performs a writeback cycle.

Exit Stop Grant Inquire State. Following the completion of any writeback, the processor returns to the state from which it entered the Stop Grant Inquire state.

Stop Clock State

Enter Stop Clock State. If the CLK signal is stopped while the Mobile AMD-K6-III-P processor is in the Stop Grant state, the processor enters the Stop Clock state. Because all internal clocks and the PLL are not running in the Stop Clock state, the Stop Clock state represents the minimum-power state of all clock control states. The CLK signal must be held Low while it is stopped.

The Stop Clock state cannot be entered from the Halt state.

INTR is the only input signal that is allowed to change states while the processor is in the Stop Clock state. However, INTR is not sampled until the processor returns to the Stop Grant state. All other input signals must remain unchanged in the Stop Clock state.

Exit Stop Clock State. The Mobile AMD-K6-III-P processor returns to the Stop Grant state from the Stop Clock state after the CLK signal is started and the internal PLL has stabilized. PLL stabilization is achieved after the CLK signal has been running within its specification for a minimum of 1.0 ms.

The frequency of CLK when exiting the Stop Clock state can be different than the frequency of CLK when entering the Stop Clock state.

The state of the BF[2:0] signals when exiting the Stop Clock state is ignored because the BF[2:0] signals are only sampled during the falling transition of RESET.

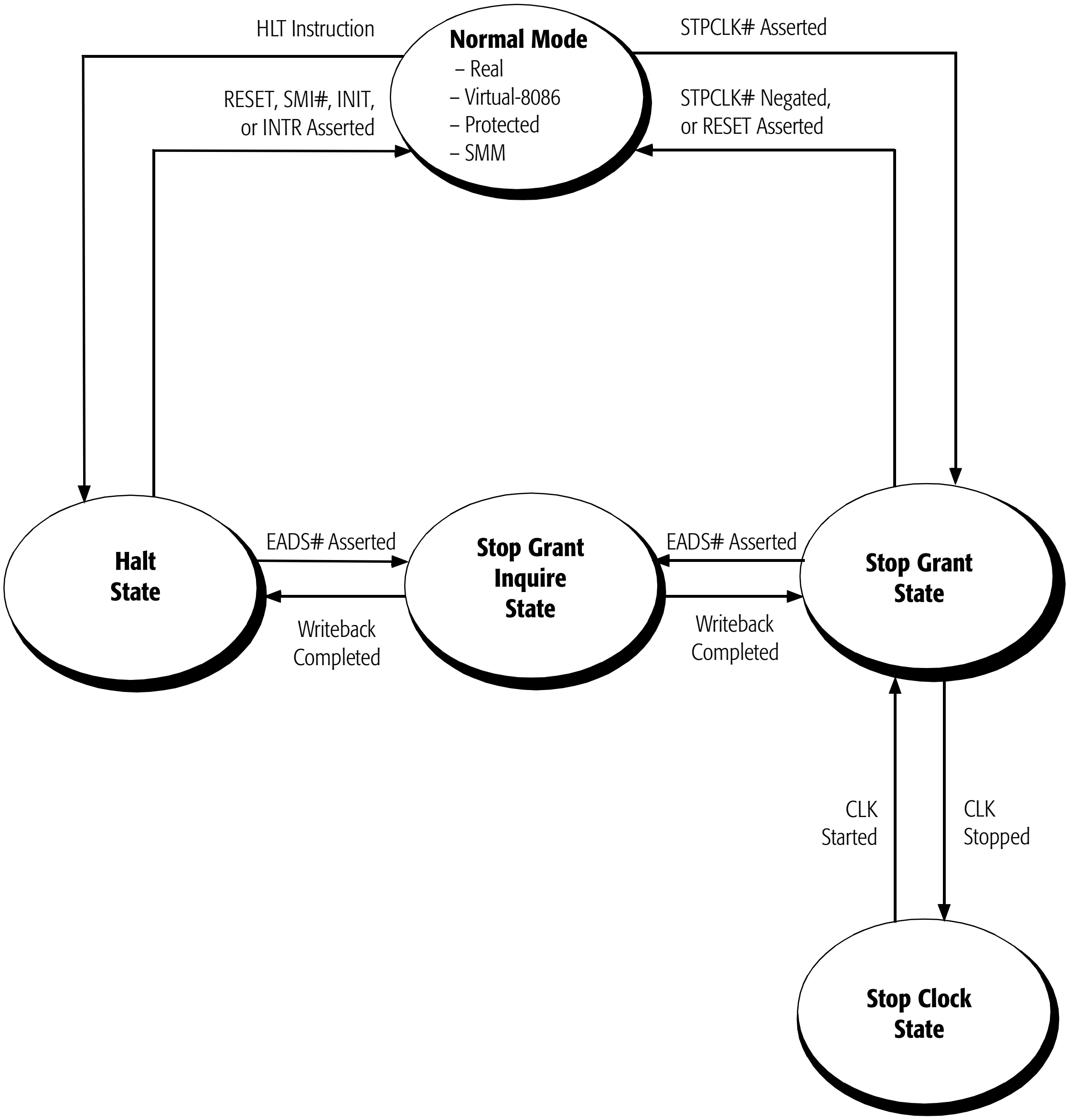


Figure 7. Clock Control State Transitions

5.3 System Management Mode (SMM)

Overview

SMM is an alternate operating mode entered by way of a system management interrupt (SMI) and handled by an interrupt service routine. SMM is designed for system control activities such as power management. These activities appear transparent to conventional operating systems like DOS and Windows. SMM is primarily targeted for use by the Basic Input Output System (BIOS) and specialized low-level device drivers. The code and data for SMM are stored in the SMM memory area, which is isolated from main memory.

The processor enters SMM by the system logic's assertion of the SMI# interrupt and the processor's acknowledgment by the assertion of SMIACT#. At this point the processor saves its state into the SMM memory state-save area and jumps to the SMM service routine. The processor returns from SMM when it executes the RSM (resume) instruction from within the SMM service routine. Subsequently, the processor restores its state from the SMM save area, negates SMIACT#, and resumes execution with the instruction following the point where it entered SMM.

The following sections summarize the SMM state-save area, entry into and exit from SMM, exceptions and interrupts in SMM, memory allocation and addressing in SMM, and the SMI# and SMIACT# signals.

SMM Operating Mode and Default Register Values

The software environment within SMM has the following characteristics:

- Addressing and operation in Real mode
- 4-Gbyte segment limits
- Default 16-bit operand, address, and stack sizes, although instruction prefixes can override these defaults
- Control transfers that do not override the default operand size truncate the EIP to 16 bits
- Far jumps or calls cannot transfer control to a segment with a base address requiring more than 20 bits, as in Real mode segment-base addressing
- A20M# is masked
- Interrupt vectors use the Real-mode interrupt vector table
- The IF flag in EFLAGS is cleared (INTR not recognized)

- The TF flag in EFLAGS is cleared
- The NMI and INIT interrupts are disabled
- Debug register DR7 is cleared (debug traps disabled)

Figure 8 shows the default map of the SMM memory area. It consists of a 64-Kbyte area, between 0003_0000h and 0003_FFFFh, of which the top 32 Kbytes (0003_8000h to 0003_FFFFh) must be populated with RAM. The default code-segment (CS) base address for the area—called the *SMM base address*—is at 0003_0000h. The top 512 bytes (0003_FE00h to 0003_FFFFh) contain a fill-down *SMM state-save area*. The default entry point for the SMM service routine is 0003_8000h.

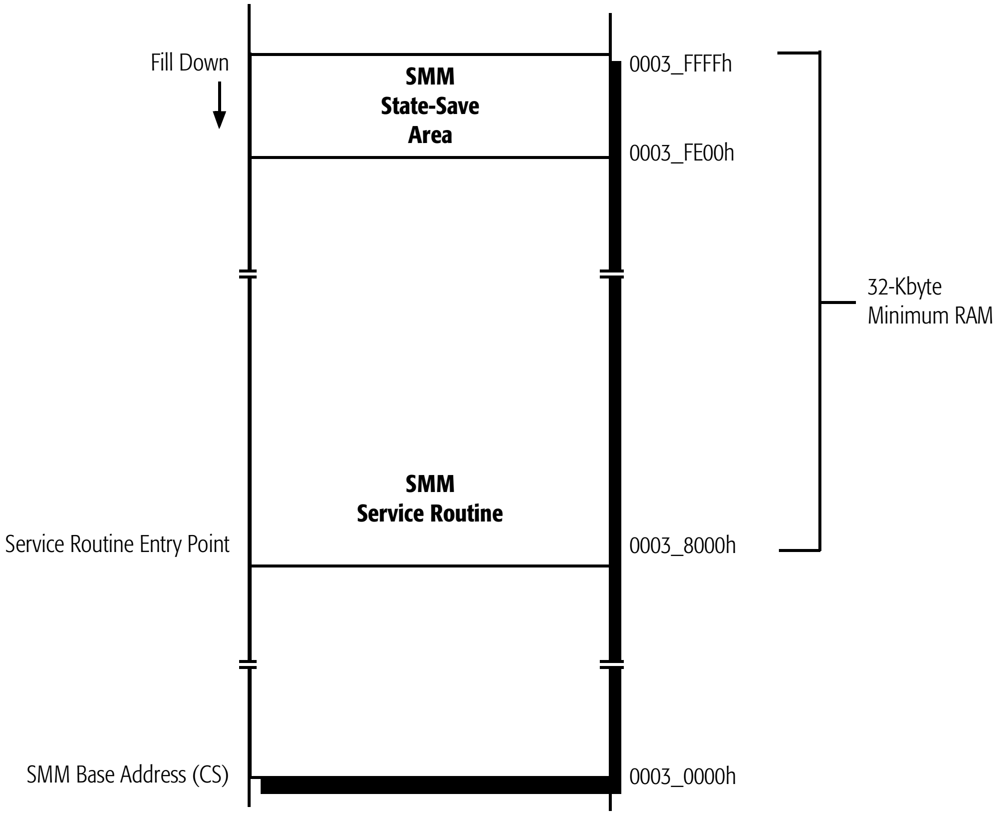


Figure 8. SMM Memory

Table 8 shows the initial state of registers when entering SMM.

Table 8. Initial State of Registers in SMM

Registers	SMM Initial State
General Purpose Registers	unmodified
EFLAGs	0000_0002h
CR0	PE, EM, TS, and PG are cleared (bits 0, 2, 3, and 31). The other bits are unmodified.
DR7	0000_0400h
GDTR, LDTR, IDTR, TSSR, DR6	unmodified
EIP	0000_8000h
CS	0003_0000h
DS, ES, FS, GS, SS	0000_0000h

SMM State-Save Area

When the processor acknowledges an SMI# interrupt by asserting SMIACT#, it saves its state in a 512-byte SMM state-save area shown in Table 9. The save begins at the top of the SMM memory area (SMM base address + FFFFh) and fills down to SMM base address + FE00h.

Table 9 shows the offsets in the SMM state-save area relative to the SMM base address. The SMM service routine can alter any of the read/write values in the state-save area.

Table 9. SMM State-Save Area Map

Address Offset	Contents Saved
FFFCh	CR0
FFF8h	CR3
FFF4h	EFLAGS
FFF0h	EIP
FFECh	EDI
FFE8h	ESI
FFE4h	EBP
FFE0h	ESP
FFDCh	EBX
FFD8h	EDX

Notes:

– No data dump at that address

** Only contains information if SMI# is asserted during a valid I/O bus cycle.*

Table 9. SMM State-Save Area Map (continued)

Address Offset	Contents Saved
FFD4h	ECX
FFD0h	EAX
FFCCh	DR6
FFC8h	DR7
FFC4h	TR
FFC0h	LDTR Base
FFBCh	GS
FFB8h	FS
FFB4h	DS
FFB0h	SS
FFACh	CS
FFA8h	ES
FFA4h	I/O Trap Dword
FFA0h	–
FF9Ch	I/O Trap EIP*
FF98h	–
FF94h	–
FF90h	IDT Base
FF8Ch	IDT Limit
FF88h	GDT Base
FF84h	GDT Limit
FF80h	TSS Attr
FF7Ch	TSS Base
FF78h	TSS Limit
FF74h	–
FF70h	LDT High
FF6Ch	LDT Low
FF68h	GS Attr
FF64h	GS Base
FF60h	GS Limit
FF5Ch	FS Attr

Notes:

– *No data dump at that address*

* *Only contains information if SMI# is asserted during a valid I/O bus cycle.*

Table 9. SMM State-Save Area Map (continued)

Address Offset	Contents Saved
FF58h	FS Base
FF54h	FS Limit
FF50h	DS Attr
FF4Ch	DS Base
FF48h	DS Limit
FF44h	SS Attr
FF40h	SS Base
FF3Ch	SS Limit
FF38h	CS Attr
FF34h	CS Base
FF30h	CS Limit
FF2Ch	ES Attr
FF28h	ES Base
FF24h	ES Limit
FF20h	–
FF1Ch	–
FF18h	–
FF14h	CR2
FF10h	CR4
FF0Ch	I/O restart ESI*
FF08h	I/O restart ECX*
FF04h	I/O restart EDI*
FF02h	HALT Restart Slot
FF00h	I/O Trap Restart Slot
FEFCh	SMM RevID
FEF8h	SMM BASE
FEF7h–FE00h	–

Notes:

– *No data dump at that address*

* *Only contains information if SMI# is asserted during a valid I/O bus cycle.*

SMM Revision Identifier

The SMM revision identifier at offset FEFCh in the SMM state-save area specifies the version of SMM and the extensions that are available on the processor. The SMM revision identifier fields are as follows:

- *Bits 31–18—Reserved*
- *Bit 17*—SMM base address relocation (1 = enabled)
- *Bit 16*—I/O trap restart (1 = enabled)
- *Bits 15–0*—SMM revision level for the Mobile AMD-K6-III-P processor = 0002h

Table 10 shows the format of the SMM Revision Identifier.

Table 10. SMM Revision Identifier

31–18	17	16	15–0
Reserved	SMM Base Relocation	I/O Trap Extension	SMM Revision Level
0	1	1	0002h

SMM Base Address

During RESET, the processor sets the base address of the code-segment (CS) for the SMM memory area—the *SMM base address*—to its default, 0003_0000h. The SMM base address at offset FEF8h in the SMM state-save area can be changed by the SMM service routine to any address that is aligned to a 32-Kbyte boundary. (Locations not aligned to a 32-Kbyte boundary cause the processor to enter the Shutdown state when executing the RSM instruction.)

In some operating environments it may be desirable to relocate the 64-Kbyte SMM memory area to a high memory area in order to provide more low memory for legacy software. During system initialization, the base of the 64-Kbyte SMM memory area is relocated by the BIOS. To relocate the SMM base address, the system enters the SMM handler at the default address. This handler changes the SMM base address location in the SMM state-save area, copies the SMM handler to the new location, and exits SMM.

The next time SMM is entered, the processor saves its state at the new base address. This new address is used for every SMM entry until the SMM base address in the SMM state-save area is changed or a hardware reset occurs.

Halt Restart Slot

During entry into SMM, the halt restart slot at offset FF02h in the SMM state-save area indicates if SMM was entered from the Halt state. Before returning from SMM, the halt restart slot (offset FF02h) can be written to by the SMM service routine to specify whether the return from SMM takes the processor back to the Halt state or to the next instruction after the HLT instruction.

Upon entry into SMM, the halt restart slot is defined as follows:

- *Bits 15–1*—Reserved
- *Bit 0*—Point of entry to SMM:

  1 = entered from Halt state

  0 = not entered from Halt state

After entry into the SMI handler and before returning from SMM, the halt restart slot can be written using the following definition:

- *Bits 15–1*—Reserved
- *Bit 0*—Point of return when exiting from SMM:

  1 = return to Halt state

  0 = return to next instruction after the HLT instruction

If the return from SMM takes the processor back to the Halt state, the HLT instruction is not re-executed, but the Halt special bus cycle is driven on the bus after the return.

I/O Trap Dword

If the assertion of SMI# is recognized during the execution of an I/O instruction, the I/O trap dword at offset FFA4h in the SMM state-save area contains information about the instruction. The fields of the I/O trap dword are configured as follows:

- *Bits 31–16*—I/O port address
- *Bits 15–4*—Reserved
- *Bit 3*—REP (repeat) string operation
  (1 = REP string, 0 = not a REP string)
- *Bit 2*—I/O string operation
  (1 = I/O string, 0 = not a I/O string)
- *Bit 1*—Valid I/O instruction (1 = valid, 0 = invalid)
- *Bit 0*—Input or output instruction (1 = INx, 0 = OUTx)

Table 11 shows the format of the I/O trap dword.

Table 11. I/O Trap Dword Configuration

31–16	15–4	3	2	1	0
I/O Port Address	Reserved	REP String Operation	I/O String Operation	Valid I/O Instruction	Input or Output

The I/O trap dword is related to the I/O trap restart slot (see "I/O Trap Restart Slot" on page 47). If bit 1 of the I/O trap dword is set by the processor, it means that SMI# was asserted during the execution of an I/O instruction. The SMI handler tests bit 1 to see if there is a valid I/O instruction trapped. If the I/O instruction is valid, the SMI handler is required to ensure the I/O trap restart slot is set properly. The I/O trap restart slot informs the CPU whether it should re-execute the I/O instruction after the RSM or execute the instruction following the trapped I/O instruction.

Note: *If SMI# is sampled asserted during an I/O bus cycle a minimum of three clock edges before BRDY# is sampled asserted, the associated I/O instruction is guaranteed to be trapped by the SMI handler.*

I/O Trap Restart Slot

The I/O trap restart slot at offset FF00h in the SMM state-save area specifies whether the trapped I/O instruction should be re-executed on return from SMM. This slot in the state-save area is called the *I/O instruction restart* function. Re-executing a trapped I/O instruction is useful, for example, if an I/O write occurs to a disk that is powered down. The system logic monitoring such an access can assert SMI#. Then the SMM service routine would query the system logic, detect a failed I/O write, take action to power-up the I/O device, enable the I/O trap restart slot feature, and return from SMM.

The fields of the I/O trap restart slot are defined as follows:

- *Bits 31–16*—Reserved
- *Bits 15–0*—I/O instruction restart on return from SMM:

  0000h = execute the next instruction after the trapped I/O instruction

  00FFh = re-execute the trapped I/O instruction

Table 12 shows the format of the I/O trap restart slot.

Table 12. I/O Trap Restart Slot

31–16	15–0
Reserved	I/O Instruction restart on return from SMM: ■ 0000h = execute the next instruction after the trapped I/O ■ 00FFh = re-execute the trapped I/O instruction

The processor initializes the I/O trap restart slot to 0000h upon entry into SMM. If SMM was entered due to a trapped I/O instruction, the processor indicates the validity of the I/O instruction by setting or clearing bit 1 of the I/O trap dword at offset FFA4h in the SMM state-save area. The SMM service routine should test bit 1 of the I/O trap dword to determine if a valid I/O instruction was being executed when entering SMM and before writing the I/O trap restart slot. If the I/O instruction is valid, the SMM service routine can safely rewrite the I/O trap restart slot with the value 00FFh, which causes the processor to re-execute the trapped I/O instruction when the RSM instruction is executed. If the I/O instruction is invalid, writing the I/O trap restart slot has undefined results.

If a second SMI# is asserted and a valid I/O instruction was trapped by the first SMM handler, the CPU services the second SMI# prior to re-executing the trapped I/O instruction. The second entry into SMM never has bit 1 of the I/O trap dword set, and the second SMM service routine must not rewrite the I/O trap restart slot.

During a simultaneous SMI# I/O instruction trap and debug breakpoint trap, the Mobile AMD-K6-III-P processor first responds to the SMI# and postpones recognizing the debug exception until after returning from SMM via the RSM instruction. If the debug registers DR3–DR0 are used while in SMM, they must be saved and restored by the SMM handler. The processor automatically saves and restores DR7–DR6. If the I/O trap restart slot in the SMM state-save area contains the value 00FFh when the RSM instruction is executed, the debug trap does not occur until after the I/O instruction is re-executed.

Exceptions, Interrupts, and Debug in SMM

During an SMI# I/O trap, the exception/interrupt priority of the Mobile AMD-K6-III-P processor changes from its normal priority. The normal priority places the debug traps at a priority higher than the sampling of the FLUSH# or SMI# signals. However, during an SMI# I/O trap, the sampling of the FLUSH# or SMI# signals takes precedence over debug traps.

The processor recognizes the assertion of NMI within SMM immediately after the completion of an IRET instruction. Once NMI is recognized within SMM, NMI recognition remains enabled until SMM is exited, at which point NMI masking is restored to the state it was in before entering SMM.

6 Signal Switching Characteristics

The Mobile AMD-K6-III-P processor signal switching characteristics are presented in Table 13 through Table 22. Valid delay, float, setup, and hold timing specifications are listed. These specifications are provided for the system designer to determine if the timings necessary for the processor to interface with the system logic are met. Table 13 and Table 14 contain the switching characteristics of the CLK input. Table 15 through Table 18 contain the timings for the normal operation signals. Table 19 and Table 20 contain the timings for RESET and the configuration signals. Table 21 and Table 22 contain the timings for the test operation signals.

All signal timings provided are:

- Measured between CLK, TCK, or RESET at 1.5 V and the corresponding signal at 1.5 V—this applies to input and output signals that are switching from Low to High, or from High to Low
- Based on input signals applied at a slew rate of 1 V/ns between 0 V and 3 V (rising) and 3 V to 0 V (falling)
- Valid within the operating ranges given in “Operating Ranges” on page 69
- Based on a load capacitance (C_L) of 0 pF

6.1 CLK Switching Characteristics

Table 13 and Table 14 contain the switching characteristics of the CLK input to the Mobile AMD-K6-III-P processor for 100-MHz and 66-MHz bus operation, respectively, as measured at the voltage levels indicated by Figure 9 on page 53.

The CLK Period Stability specifies the variance (jitter) allowed between successive periods of the CLK input measured at 1.5 V. This parameter must be considered as one of the elements of clock skew between the Mobile AMD-K6-III-P and the system logic.

6.2 Clock Switching Characteristics for 100-MHz Bus Operation

Table 13. CLK Switching Characteristics for 100-MHz Bus Operation

Symbol	Parameter Description	Preliminary Data		Figure	Comments
		Min	Max		
	Frequency	33.3 MHz	100 MHz		In Normal Mode
t_1	CLK Period	10.0 ns		9	In Normal Mode
t_2	CLK High Time	3.0 ns		9	
t_3	CLK Low Time	3.0 ns		9	
t_4	CLK Fall Time	0.15 ns	1.5 ns	9	
t_5	CLK Rise Time	0.15 ns	1.5 ns	9	
	CLK Period Stability		± 250 ps		Note

Note:

Jitter frequency power spectrum peaking must occur at frequencies greater than (Frequency of CLK)/3 or less than 500 kHz.

6.3 Clock Switching Characteristics for 66-MHz Bus Operation

Table 14. CLK Switching Characteristics for 66-MHz Bus Operation

Symbol	Parameter Description	Preliminary Data		Figure	Comments
		Min	Max		
	Frequency	33.3 MHz	66.6 MHz		In Normal Mode
t_1	CLK Period	15.0 ns	30.0 ns	9	In Normal Mode
t_2	CLK High Time	4.0 ns		9	
t_3	CLK Low Time	4.0 ns		9	
t_4	CLK Fall Time	0.15 ns	1.5 ns	9	
t_5	CLK Rise Time	0.15 ns	1.5 ns	9	
	CLK Period Stability		± 250 ps		Note

Note:

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

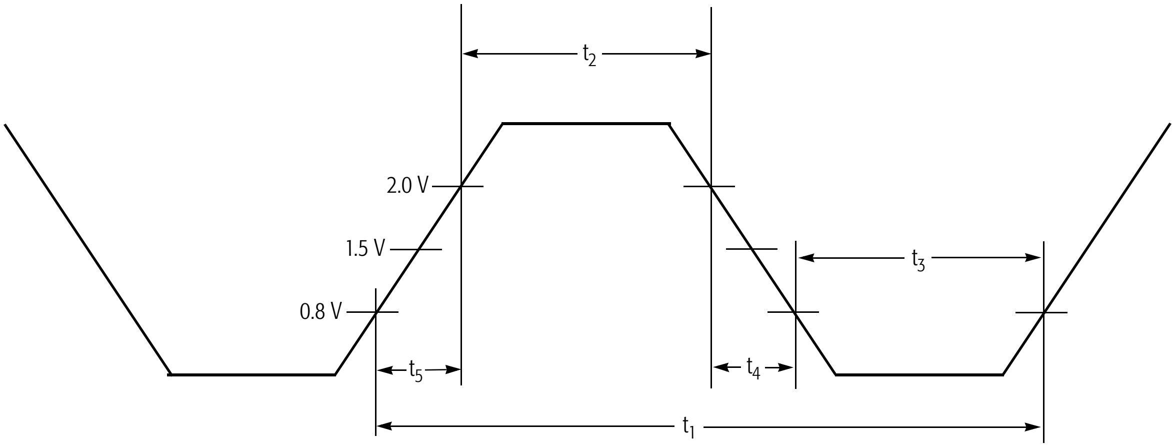


Figure 9. CLK Waveform

6.4 Valid Delay, Float, Setup, and Hold Timings

Valid delay and float timings are given for output signals during functional operation and are given relative to the rising edge of CLK. During boundary-scan testing, valid delay and float timings for output signals are with respect to the falling edge of TCK. The maximum valid delay timings are provided to allow a system designer to determine if setup times to the system logic can be met. Likewise, the minimum valid delay timings are used to analyze hold times to the system logic.

The setup and hold time requirements for the Mobile AMD-K6-III-P processor input signals must be met by the system logic to assure the proper operation of the processor. The setup and hold timings during functional and boundary-scan test mode are given relative to the rising edge of CLK and TCK, respectively.

6.5 Output Delay Timings for 100-MHz Bus Operation

Table 15. Output Delay Timings for 100-MHz Bus Operation

Symbol	Parameter Description	Preliminary Data		Figure	Comments
		Min	Max		
t_6	A[31:3] Valid Delay	1.1 ns	4.0 ns	11	
t_7	A[31:3] Float Delay		7.0 ns	12	
t_8	ADS# Valid Delay	1.0 ns	4.0 ns	11	
t_9	ADS# Float Delay		7.0 ns	12	
t_{10}	ADSC# Valid Delay	1.0 ns	4.0 ns	11	
t_{11}	ADSC# Float Delay		7.0 ns	12	
t_{12}	AP Valid Delay	1.0 ns	5.5 ns	11	
t_{13}	AP Float Delay		7.0 ns	12	
t_{14}	APCHK# Valid Delay	1.0 ns	4.5 ns	11	
t_{15}	BE[7:0]# Valid Delay	1.0 ns	4.0 ns	11	
t_{16}	BE[7:0]# Float Delay		7.0 ns	12	
t_{17}	BREQ Valid Delay	1.0 ns	4.0 ns	11	
t_{18}	CACHE# Valid Delay	1.0 ns	4.0 ns	11	
t_{19}	CACHE# Float Delay		7.0 ns	12	
t_{20}	D/C# Valid Delay	1.0 ns	4.0 ns	11	
t_{21}	D/C# Float Delay		7.0 ns	12	
t_{22}	D[63:0] Write Data Valid Delay	1.3 ns	4.5 ns	11	
t_{23}	D[63:0] Write Data Float Delay		7.0 ns	12	
t_{24}	DP[7:0] Write Data Valid Delay	1.3 ns	4.5 ns	11	
t_{25}	DP[7:0] Write Data Float Delay		7.0 ns	12	
t_{26}	FERR# Valid Delay	1.0 ns	4.5 ns	11	
t_{27}	HIT# Valid Delay	1.0 ns	4.0 ns	11	
t_{28}	HITM# Valid Delay	1.1 ns	4.0 ns	11	
t_{29}	HLDA Valid Delay	1.0 ns	4.0 ns	11	
t_{30}	LOCK# Valid Delay	1.1 ns	4.0 ns	11	
t_{31}	LOCK# Float Delay		7.0 ns	12	
t_{32}	M/IO# Valid Delay	1.0 ns	4.0 ns	11	
t_{33}	M/IO# Float Delay		7.0 ns	12	

Table 15. Output Delay Timings for 100-MHz Bus Operation (continued)

Symbol	Parameter Description	Preliminary Data		Figure	Comments
		Min	Max		
t_{34}	PCD Valid Delay	1.0 ns	4.0 ns	11	
t_{35}	PCD Float Delay		7.0 ns	12	
t_{36}	PCHK# Valid Delay	1.0 ns	4.5 ns	11	
t_{37}	PWT Valid Delay	1.0 ns	4.0 ns	11	
t_{38}	PWT Float Delay		7.0 ns	12	
t_{39}	SCYC Valid Delay	1.0 ns	4.0 ns	11	
t_{40}	SCYC Float Delay		7.0 ns	12	
t_{41}	SMIACT# Valid Delay	1.0 ns	4.0 ns	11	
t_{42}	W/R# Valid Delay	1.0 ns	4.0 ns	11	
t_{43}	W/R# Float Delay		7.0 ns	12	

6.6 Input Setup and Hold Timings for 100-MHz Bus Operation

Table 16. Input Setup and Hold Timings for 100-MHz Bus Operation

Symbol	Parameter Description	Preliminary Data		Figure	Comments
		Min	Max		
t_{44}	A[31:5] Setup Time	3.0 ns		13	
t_{45}	A[31:5] Hold Time	1.0 ns		13	
t_{46}	A20M# Setup Time	3.0 ns		13	Note 1
t_{47}	A20M# Hold Time	1.0 ns		13	Note 1
t_{48}	AHOLD Setup Time	3.5 ns		13	
t_{49}	AHOLD Hold Time	1.0 ns		13	
t_{50}	AP Setup Time	1.7 ns		13	
t_{51}	AP Hold Time	1.0 ns		13	
t_{52}	BOFF# Setup Time	3.5 ns		13	
t_{53}	BOFF# Hold Time	1.0 ns		13	
t_{54}	BRDY# Setup Time	3.0 ns		13	
t_{55}	BRDY# Hold Time	1.0 ns		13	
t_{56}	BRDYC# Setup Time	3.0 ns		13	
t_{57}	BRDYC# Hold Time	1.0 ns		13	
t_{58}	D[63:0] Read Data Setup Time	1.7 ns		13	
t_{59}	D[63:0] Read Data Hold Time	1.5 ns		13	
t_{60}	DP[7:0] Read Data Setup Time	1.7 ns		13	
t_{61}	DP[7:0] Read Data Hold Time	1.5 ns		13	
t_{62}	EADS# Setup Time	3.0 ns		13	
t_{63}	EADS# Hold Time	1.0 ns		13	
t_{64}	EWBE# Setup Time	1.7 ns		13	
t_{65}	EWBE# Hold Time	1.0 ns		13	
t_{66}	FLUSH# Setup Time	1.7 ns		13	Note 2
t_{67}	FLUSH# Hold Time	1.0 ns		13	Note 2

Notes:

1. *These level-sensitive signals can be asserted synchronously or asynchronously. To be sampled on a specific clock edge, setup and hold times must be met. If asserted asynchronously, they must be asserted for a minimum pulse width of two clocks.*
2. *These edge-sensitive signals can be asserted synchronously or asynchronously. To be sampled on a specific clock edge, setup and hold times must be met. If asserted asynchronously, they must have been negated at least two clocks prior to assertion and must remain asserted at least two clocks.*

Table 16. Input Setup and Hold Timings for 100-MHz Bus Operation (continued)

Symbol	Parameter Description	Preliminary Data		Figure	Comments
		Min	Max		
t_{68}	HOLD Setup Time	1.7 ns		13	
t_{69}	HOLD Hold Time	1.5 ns		13	
t_{70}	IGNNE# Setup Time	1.7 ns		13	Note 1
t_{71}	IGNNE# Hold Time	1.0 ns		13	Note 1
t_{72}	INIT Setup Time	1.7 ns		13	Note 2
t_{73}	INIT Hold Time	1.0 ns		13	Note 2
t_{74}	INTR Setup Time	1.7 ns		13	Note 1
t_{75}	INTR Hold Time	1.0 ns		13	Note 1
t_{76}	INV Setup Time	1.7 ns		13	
t_{77}	INV Hold Time	1.0 ns		13	
t_{78}	KEN# Setup Time	3.0 ns		13	
t_{79}	KEN# Hold Time	1.0 ns		13	
t_{80}	NA# Setup Time	1.7 ns		13	
t_{81}	NA# Hold Time	1.0 ns		13	
t_{82}	NMI Setup Time	1.7 ns		13	Note 2
t_{83}	NMI Hold Time	1.0 ns		13	Note 2
t_{84}	SMI# Setup Time	1.7 ns		13	Note 2
t_{85}	SMI# Hold Time	1.0 ns		13	Note 2
t_{86}	STPCLK# Setup Time	1.7 ns		13	Note 1
t_{87}	STPCLK# Hold Time	1.0 ns		13	Note 1
t_{88}	WB/WT# Setup Time	1.7 ns		13	
t_{89}	WB/WT# Hold Time	1.0 ns		13	

Notes:

1. *These level-sensitive signals can be asserted synchronously or asynchronously. To be sampled on a specific clock edge, setup and hold times must be met. If asserted asynchronously, they must be asserted for a minimum pulse width of two clocks.*
2. *These edge-sensitive signals can be asserted synchronously or asynchronously. To be sampled on a specific clock edge, setup and hold times must be met. If asserted asynchronously, they must have been negated at least two clocks prior to assertion and must remain asserted at least two clocks.*

6.7 Output Delay Timings for 66-MHz Bus Operation

Table 17. Output Delay Timings for 66-MHz Bus Operation

Symbol	Parameter Description	Preliminary Data		Figure	Comments
		Min	Max		
t_6	A[31:3] Valid Delay	1.1 ns	6.3 ns	11	
t_7	A[31:3] Float Delay		10.0 ns	12	
t_8	ADS# Valid Delay	1.0 ns	6.0 ns	11	
t_9	ADS# Float Delay		10.0 ns	12	
t_{10}	ADSC# Valid Delay	1.0 ns	7.0 ns	11	
t_{11}	ADSC# Float Delay		10.0 ns	12	
t_{12}	AP Valid Delay	1.0 ns	8.5 ns	11	
t_{13}	AP Float Delay		10.0 ns	12	
t_{14}	APCHK# Valid Delay	1.0 ns	8.3 ns	11	
t_{15}	BE[7:0}# Valid Delay	1.0 ns	7.0 ns	11	
t_{16}	BE[7:0}# Float Delay		10.0 ns	12	
t_{17}	BREQ Valid Delay	1.0 ns	8.0 ns	11	
t_{18}	CACHE# Valid Delay	1.0 ns	7.0 ns	11	
t_{19}	CACHE# Float Delay		10.0 ns	12	
t_{20}	D/C# Valid Delay	1.0 ns	7.0 ns	11	
t_{21}	D/C# Float Delay		10.0 ns	12	
t_{22}	D[63:0] Write Data Valid Delay	1.3 ns	7.5 ns	11	
t_{23}	D[63:0] Write Data Float Delay		10.0 ns	12	
t_{24}	DP[7:0] Write Data Valid Delay	1.3 ns	7.5 ns	11	
t_{25}	DP[7:0] Write Data Float Delay		10.0 ns	12	
t_{26}	FERR# Valid Delay	1.0 ns	8.3 ns	11	
t_{27}	HIT# Valid Delay	1.0 ns	6.8 ns	11	
t_{28}	HITM# Valid Delay	1.1 ns	6.0 ns	11	
t_{29}	HLDA Valid Delay	1.0 ns	6.8 ns	11	
t_{30}	LOCK# Valid Delay	1.1 ns	7.0 ns	11	
t_{31}	LOCK# Float Delay		10.0 ns	12	
t_{32}	M/IO# Valid Delay	1.0 ns	5.9 ns	11	
t_{33}	M/IO# Float Delay		10.0 ns	12	

Table 17. Output Delay Timings for 66-MHz Bus Operation (continued)

Symbol	Parameter Description	Preliminary Data		Figure	Comments
		Min	Max		
t_{34}	PCD Valid Delay	1.0 ns	7.0 ns	11	
t_{35}	PCD Float Delay		10.0 ns	12	
t_{36}	PCHK# Valid Delay	1.0 ns	7.0 ns	11	
t_{37}	PWT Valid Delay	1.0 ns	7.0 ns	11	
t_{38}	PWT Float Delay		10.0 ns	12	
t_{39}	SCYC Valid Delay	1.0 ns	7.0 ns	11	
t_{40}	SCYC Float Delay		10.0 ns	12	
t_{41}	SMIACT# Valid Delay	1.0 ns	7.3 ns	11	
t_{42}	W/R# Valid Delay	1.0 ns	7.0 ns	11	
t_{43}	W/R# Float Delay		10.0 ns	12	

6.8 Input Setup and Hold Timings for 66-MHz Bus Operation

Table 18. Input Setup and Hold Timings for 66-MHz Bus Operation

Symbol	Parameter Description	Preliminary Data		Figure	Comments
		Min	Max		
t_{44}	A[31:5] Setup Time	6.0 ns		13	
t_{45}	A[31:5] Hold Time	1.0 ns		13	
t_{46}	A20M# Setup Time	5.0 ns		13	Note 1
t_{47}	A20M# Hold Time	1.0 ns		13	Note 1
t_{48}	AHOLD Setup Time	5.5 ns		13	
t_{49}	AHOLD Hold Time	1.0 ns		13	
t_{50}	AP Setup Time	5.0 ns		13	
t_{51}	AP Hold Time	1.0 ns		13	
t_{52}	BOFF# Setup Time	5.5 ns		13	
t_{53}	BOFF# Hold Time	1.0 ns		13	
t_{54}	BRDY# Setup Time	5.0 ns		13	
t_{55}	BRDY# Hold Time	1.0 ns		13	
t_{56}	BRDYC# Setup Time	5.0 ns		13	
t_{57}	BRDYC# Hold Time	1.0 ns		13	
t_{58}	D[63:0] Read Data Setup Time	2.8 ns		13	
t_{59}	D[63:0] Read Data Hold Time	1.5 ns		13	
t_{60}	DP[7:0] Read Data Setup Time	2.8 ns		13	
t_{61}	DP[7:0] Read Data Hold Time	1.5 ns		13	
t_{62}	EADS# Setup Time	5.0 ns		13	
t_{63}	EADS# Hold Time	1.0 ns		13	
t_{64}	EWBE# Setup Time	5.0 ns		13	
t_{65}	EWBE# Hold Time	1.0 ns		13	
t_{66}	FLUSH# Setup Time	5.0 ns		13	Note 2
t_{67}	FLUSH# Hold Time	1.0 ns		13	Note 2

Notes:

1. *These level-sensitive signals can be asserted synchronously or asynchronously. To be sampled on a specific clock edge, setup and hold times must be met. If asserted asynchronously, they must be asserted for a minimum pulse width of two clocks.*
2. *These edge-sensitive signals can be asserted synchronously or asynchronously. To be sampled on a specific clock edge, setup and hold times must be met. If asserted asynchronously, they must have been negated at least two clocks prior to assertion and must remain asserted at least two clocks.*

Table 18. Input Setup and Hold Timings for 66-MHz Bus Operation (continued)

Symbol	Parameter Description	Preliminary Data		Figure	Comments
		Min	Max		
t_{68}	HOLD Setup Time	5.0 ns		13	
t_{69}	HOLD Hold Time	1.5 ns		13	
t_{70}	IGNNE# Setup Time	5.0 ns		13	Note 1
t_{71}	IGNNE# Hold Time	1.0 ns		13	Note 1
t_{72}	INIT Setup Time	5.0 ns		13	Note 2
t_{73}	INIT Hold Time	1.0 ns		13	Note 2
t_{74}	INTR Setup Time	5.0 ns		13	Note 1
t_{75}	INTR Hold Time	1.0 ns		13	Note 1
t_{76}	INV Setup Time	5.0 ns		13	
t_{77}	INV Hold Time	1.0 ns		13	
t_{78}	KEN# Setup Time	5.0 ns		13	
t_{79}	KEN# Hold Time	1.0 ns		13	
t_{80}	NA# Setup Time	4.5 ns		13	
t_{81}	NA# Hold Time	1.0 ns		13	
t_{82}	NMI Setup Time	5.0 ns		13	Note 2
t_{83}	NMI Hold Time	1.0 ns		13	Note 2
t_{84}	SMI# Setup Time	5.0 ns		13	Note 2
t_{85}	SMI# Hold Time	1.0 ns		13	Note 2
t_{86}	STPCLK# Setup Time	5.0 ns		13	Note 1
t_{87}	STPCLK# Hold Time	1.0 ns		13	Note 1
t_{88}	WB/WT# Setup Time	4.5 ns		13	
t_{89}	WB/WT# Hold Time	1.0 ns		13	

Notes:

1. *These level-sensitive signals can be asserted synchronously or asynchronously. To be sampled on a specific clock edge, setup and hold times must be met. If asserted asynchronously, they must be asserted for a minimum pulse width of two clocks.*
2. *These edge-sensitive signals can be asserted synchronously or asynchronously. To be sampled on a specific clock edge, setup and hold times must be met. If asserted asynchronously, they must have been negated at least two clocks prior to assertion and must remain asserted at least two clocks.*

6.9 RESET and Test Signal Timing

Table 19. RESET and Configuration Signals for 100-MHz Bus Operation

Symbol	Parameter Description	Preliminary Data		Figure	Comments
		Min	Max		
t_{90}	RESET Setup Time	1.7 ns		14	
t_{91}	RESET Hold Time	1.0 ns		14	
t_{92}	RESET Pulse Width, V_{CC} and CLK Stable	15 clocks		14	
t_{93}	RESET Active After V_{CC} and CLK Stable	1.0 ms		14	
t_{94}	BF[2:0] Setup Time	1.0 ms		14	Note 3
t_{95}	BF[2:0] Hold Time	2 clocks		14	Note 3
t_{96}	Intentionally left blank				
t_{97}	Intentionally left blank				
t_{98}	Intentionally left blank				
t_{99}	FLUSH# Setup Time	1.7 ns		14	Note 1
t_{100}	FLUSH# Hold Time	1.0 ns		14	Note 1
t_{101}	FLUSH# Setup Time	2 clocks		14	Note 2
t_{102}	FLUSH# Hold Time	2 clocks		14	Note 2

Notes:

1. *To be sampled on a specific clock edge, setup and hold times must be met the clock edge before the clock edge on which RESET is sampled negated.*
2. *If asserted asynchronously, these signals must meet a minimum setup and hold time of two clocks relative to the negation of RESET.*
3. *BF[2:0] must meet a minimum setup time of 1.0 ms and a minimum hold time of two clocks relative to the negation of RESET.*

Table 20. RESET and Configuration Signals for 66-MHz Bus Operation

Symbol	Parameter Description	Preliminary Data		Figure	Comments
		Min	Max		
t_{90}	RESET Setup Time	5.0 ns		14	
t_{91}	RESET Hold Time	1.0 ns		14	
t_{92}	RESET Pulse Width, V_{CC} and CLK Stable	15 clocks		14	
t_{93}	RESET Active After V_{CC} and CLK Stable	1.0 ms		14	
t_{94}	BF[2:0] Setup Time	1.0 ms		14	Note 3
t_{95}	BF[2:0] Hold Time	2 clocks		14	Note 3
t_{96}	Intentionally left blank				
t_{97}	Intentionally left blank				
t_{98}	Intentionally left blank				
t_{99}	FLUSH# Setup Time	5.0 ns		14	Note 1
t_{100}	FLUSH# Hold Time	1.0 ns		14	Note 1
t_{101}	FLUSH# Setup Time	2 clocks		14	Note 2
t_{102}	FLUSH# Hold Time	2 clocks		14	Note 2

Notes:

1. *To be sampled on a specific clock edge, setup and hold times must be met the clock edge before the clock edge on which RESET is sampled negated.*
2. *If asserted asynchronously, these signals must meet a minimum setup and hold time of two clocks relative to the negation of RESET.*
3. *BF[2:0] must meet a minimum setup time of 1.0 ms and a minimum hold time of two clocks relative to the negation of RESET.*

Table 21. TCK Waveform and TRST# Timing at 25 MHz

Symbol	Parameter Description	Preliminary Data		Figure	Comments
		Min	Max		
	TCK Frequency		25 MHz	15	
t_{103}	TCK Period	40.0 ns		15	
t_{104}	TCK High Time	14.0 ns		15	
t_{105}	TCK Low Time	14.0 ns		15	
t_{106}	TCK Fall Time		5.0 ns	15	Note 1, 2
t_{107}	TCK Rise Time		5.0 ns	15	Note 1, 2
t_{108}	TRST# Pulse Width	30.0 ns		16	Asynchronous

Notes:

1. *Rise/Fall times can be increased by 1.0 ns for each 10 MHz that TCK is run below its maximum frequency of 25 MHz.*
2. *Rise/Fall times are measured between 0.8 V and 2.0 V.*

Table 22. Test Signal Timing at 25 MHz

Symbol	Parameter Description	Preliminary Data		Figure	Notes
		Min	Max		
t_{109}	TDI Setup Time	5.0 ns		17	Note 2
t_{110}	TDI Hold Time	9.0 ns		17	Note 2
t_{111}	TMS Setup Time	5.0 ns		17	Note 2
t_{112}	TMS Hold Time	9.0 ns		17	Note 2
t_{113}	TDO Valid Delay	3.0 ns	13.0 ns	17	Note 1
t_{114}	TDO Float Delay		16.0 ns	17	Note 1
t_{115}	All Outputs (Non-Test) Valid Delay	3.0 ns	13.0 ns	17	Note 1
t_{116}	All Outputs (Non-Test) Float Delay		16.0 ns	17	Note 1
t_{117}	All Inputs (Non-Test) Setup Time	5.0 ns		17	Note 2
t_{118}	All Inputs (Non-Test) Hold Time	9.0 ns		17	Note 2

Notes:

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

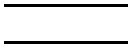
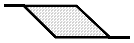
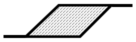
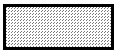
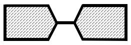
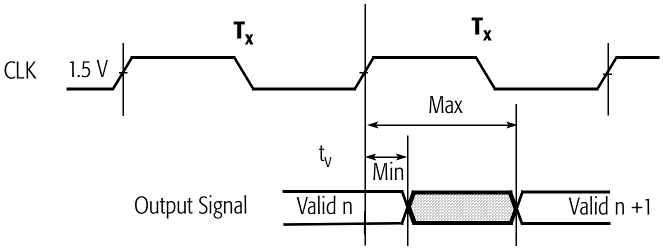
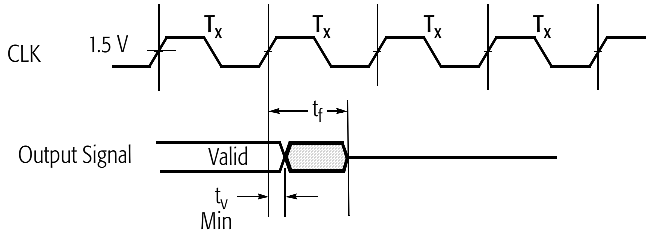
WAVEFORM	INPUTS	OUTPUTS
	Must be steady	Steady
	Can change from High to Low	Changing from High to Low
	Can change from Low to High	Changing from Low to High
	Don’t care, any change permitted	Changing, State Unknown
	(Does not apply)	Center line is high impedance state

Figure 10. Diagrams Key



v = 6, 8, 10, 12, 14, 15, 17, 18, 20, 22, 24, 26, 27, 28, 29, 30, 32, 34, 36, 37, 39, 41, 42

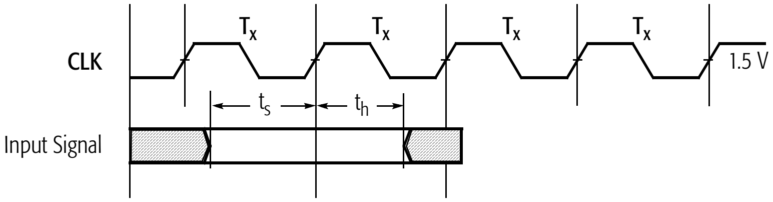
Figure 11. Output Valid Delay Timing



v = 6, 8, 10, 12, 15, 18, 20, 22, 24, 30, 32, 34, 37, 39, 42

f = 7, 9, 11, 13, 16, 19, 21, 23, 25, 31, 33, 35, 38, 40, 43

Figure 12. Maximum Float Delay Timing



s = 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88

h = 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 69, 71, 73, 75, 77, 79, 81, 83, 85, 87, 89

Figure 13. Input Setup and Hold Timing

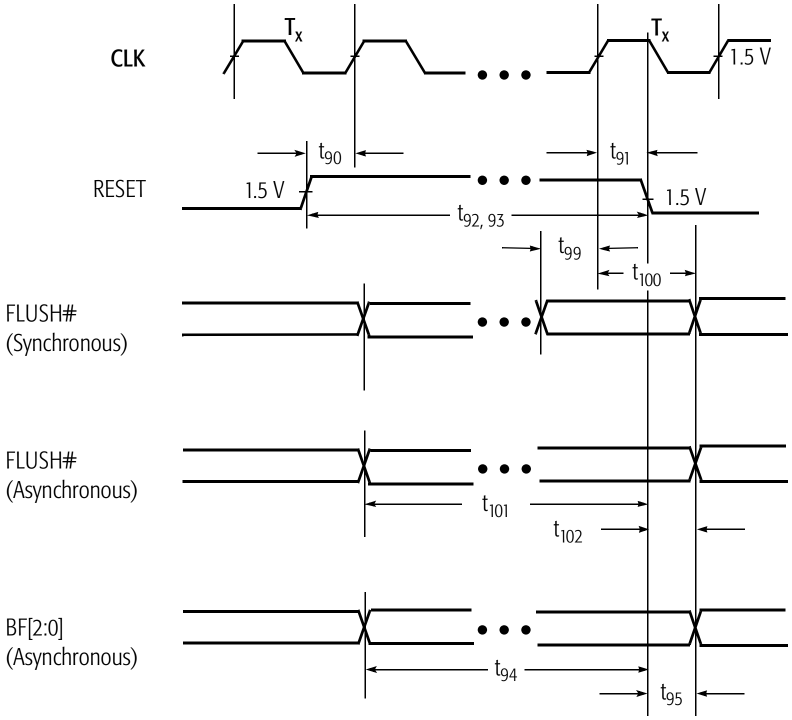


Figure 14. Reset and Configuration Timing

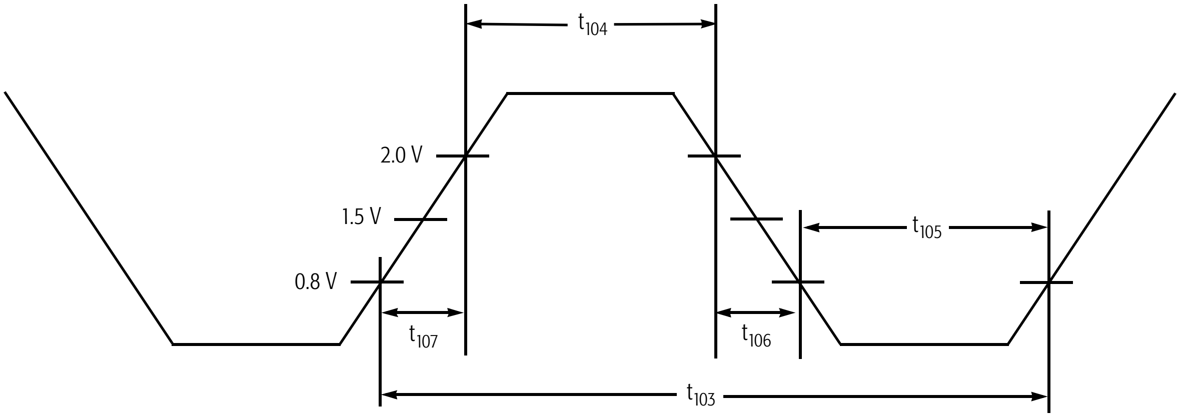


Figure 15. TCK Waveform

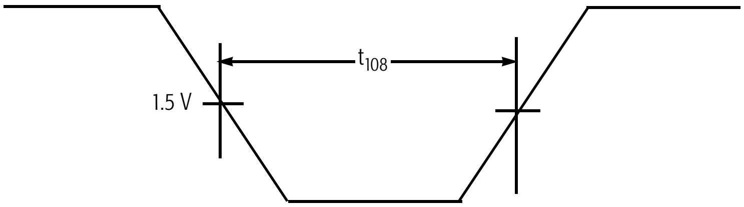


Figure 16. TRST# Timing

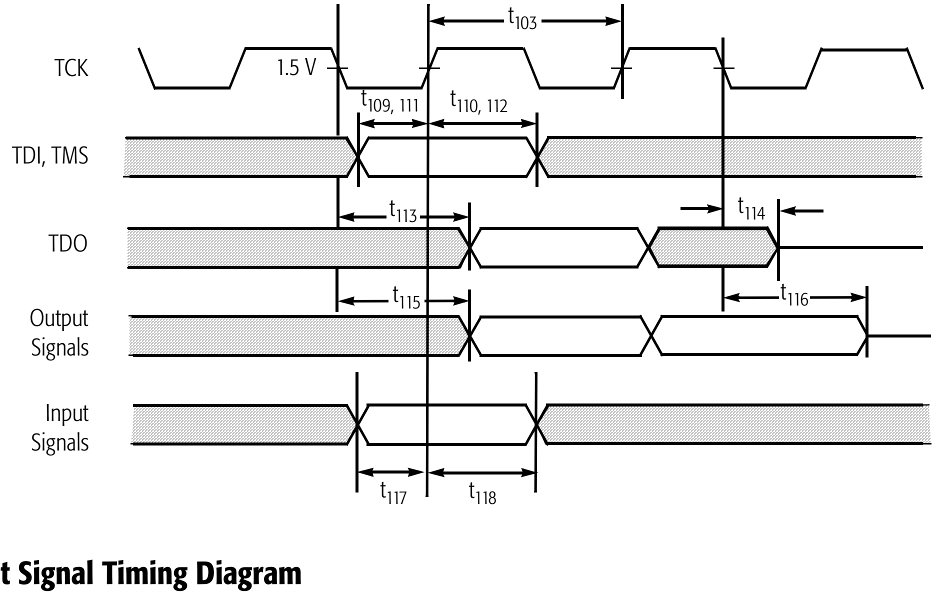


Figure 17. Test Signal Timing Diagram

7 Electrical Data

7.1 Operating Ranges

The Mobile AMD-K6-III-P processor is designed to provide functional operation if the voltage and temperature parameters are within the limits defined in Table 23.

Table 23. Operating Ranges

Parameter	Minimum	Typical	Maximum	Comments
V_{CC2}	1.9 V	2.0 V	2.1 V	Note 1, 2
V_{CC2}	2.1 V	2.2 V	2.3 V	Note 1, 3
V_{CC3}	3.135 V	3.3 V	3.6 V	Note 1
T_{CASE}	0°C		80°C	

Note:

1. *V_{CC2} and V_{CC3} are referenced from V_{SS}.*
2. *V_{CC2} specification for 2.0 V components.*
3. *V_{CC2} specification for 2.2 V components.*

7.2 Absolute Ratings

The Mobile AMD-K6-III-P processor is not designed to be operated beyond the operating ranges listed in Table 23. Exposure to conditions outside these operating ranges for extended periods of time can affect long-term reliability. Permanent damage can occur if the absolute ratings listed in Table 24 are exceeded.

Table 24. Absolute Ratings

Parameter	Minimum	Maximum	Comments
V_{CC2}	–0.5 V	2.4 V	Note 1
V_{CC2}	–0.5 V	2.6 V	Note 2
V_{CC3}	–0.5 V	3.6 V	
V_{PIN}	–0.5 V	V_{cc3} + 0.5 V and ≤ 4.0 V	Note 3
T_{CASE} (under bias)	–65°C	+110°C	
$T_{STORAGE}$	–65°C	+150°C	

Note:

1. *V_{CC2} specification for 2.0 V components.*
2. *V_{CC2} specification for 2.2 V components.*
3. *V_{PIN} (the voltage on any I/O pin) must not be greater than 0.5 V above the voltage being applied to V_{CC3}. In addition, the V_{PIN} voltage must never exceed 4.0 V.*

7.3 DC Characteristics

The DC characteristics of the Mobile AMD-K6-III-P processor are shown in Table 25.

Table 25. DC Characteristics

Symbol	Parameter Description	Preliminary Data		Comments
		Min	Max	
V_{IL}	Input Low Voltage	–0.3 V	+0.8 V	
V_{IH}	Input High Voltage	2.0 V	V_{CC3}+0.3V	Note 1
V_{OL}	Output Low Voltage		0.4 V	I_{OL} = 4.0-mA load
V_{OH}	Output High Voltage	2.4 V		I_{OH} = 3.0-mA load
I_{CC2}	2.0 V Power Supply Current		8.50 A	400 MHz, Note 2,8,10
				433 MHz, Note 2,11
				450 MHz, Note 2,10
I_{CC2}	2.2 V Power Supply Current		8.50 A	350 MHz, Note 3,10
				366 MHz, Note 3,8
				380 MHz, Note 3,9
I_{CC3}	3.3 V Power Supply Current		0.60 A	350 MHz, Note 4,10
			0.60 A	366 MHz, Note 4,8
			0.61 A	380 MHz, Note 4,9
			0.62 A	400 MHz, Note 4,8,10
			0.64 A	433 MHz, Note 4,11
			0.66 A	450 MHz, Note 4,10
I_{LI}	Input Leakage Current		±15 µA	Note 5
I_{LO}	Output Leakage Current		±15 µA	Note 5

Notes:

1. *V_{CC3} refers to the voltage being applied to V_{CC3} during functional operation.*
2. *V_{CC2} = 2.1 V – The maximum power supply current must be taken into account when designing a power supply.*
3. *V_{CC2} = 2.3 V – The maximum power supply current must be taken into account when designing a power supply.*
4. *V_{CC3} = 3.6 V – The maximum power supply current must be taken into account when designing a power supply.*
5. *Refers to inputs and I/O without an internal pullup resistor and $0 \leq V_{IN} \leq V_{CC3}$.*
6. *Refers to inputs with an internal pullup and V_{IL} = 0.4 V.*
7. *Refers to inputs with an internal pulldown and V_{IH} = 2.4 V.*
8. *This specification applies to components using a CLK frequency of 66 MHz.*
9. *This specification applies to components using a CLK frequency of 95 MHz.*
10. *This specification applies to components using a CLK frequency of 100 MHz.*
11. *This specification applies to components using a CLK frequency of 96.2 MHz.*

Table 25. DC Characteristics (continued)

Symbol	Parameter Description	Preliminary Data		Comments
		Min	Max	
I_{IL}	Input Leakage Current Bias with Pullup		–400 µA	Note 6
I_{IH}	Input Leakage Current Bias with Pulldown		200 µA	Note 7
C_{IN}	Input Capacitance		10 pF	
C_{OUT}	Output Capacitance		15 pF	
C_{OUT}	I/O Capacitance		20 pF	
C_{CLK}	CLK Capacitance		10 pF	
C_{TIN}	Test Input Capacitance (TDI, TMS, TRST#)		10 pF	
C_{TOUT}	Test Output Capacitance (TDO)		15 pF	
C_{TCK}	TCK Capacitance		10 pF	

Notes:

1. *V_{CC3} refers to the voltage being applied to V_{CC3} during functional operation.*
2. *$V_{CC2} = 2.1$ V – The maximum power supply current must be taken into account when designing a power supply.*
3. *$V_{CC2} = 2.3$ V – The maximum power supply current must be taken into account when designing a power supply.*
4. *$V_{CC3} = 3.6$ V – The maximum power supply current must be taken into account when designing a power supply.*
5. *Refers to inputs and I/O without an internal pullup resistor and $0 \leq V_{IN} \leq V_{CC3.}$*
6. *Refers to inputs with an internal pullup and $V_{IL} = 0.4$ V.*
7. *Refers to inputs with an internal pulldown and $V_{IH} = 2.4$ V.*
8. *This specification applies to components using a CLK frequency of 66 MHz.*
9. *This specification applies to components using a CLK frequency of 95 MHz.*
10. *This specification applies to components using a CLK frequency of 100 MHz.*
11. *This specification applies to components using a CLK frequency of 96.2 MHz.*

7.4 Power Dissipation

Table 26 contains the typical and maximum power dissipation of the Mobile AMD-K6-III-P processor during normal and reduced power states.

Table 26. Power Dissipation

Clock Control State	350 MHz[7]	366 MHz[5]	380 MHz[6]	400 MHz[5,7]	433 MHz[8]	450 MHz[7]	Comments
Design Power	16.00 W						Note 1
Application Power	12.60 W						Note 2
Stop Grant/Halt (Maximum)	2.56 W						Note 3
Stop Clock (Maximum)	2.25 W						Note 4

Notes:

1. *Design Power represents the maximum sustained power dissipated while executing software or instruction sequences under normal system operation with V_{CC2} = 2.0 V (for 2.0V components) or V_{CC2} = 2.2 V (for 2.2V components) and V_{CC3} = 3.3 V. Thermal solutions must use thermal feedback to limit the processor's peak power. Specified through characterization.*
2. *Application Power represents the average power dissipated while executing software or instruction sequences under normal system operation with V_{CC2} = 2.0 V (for 2.0V components) or V_{CC2} = 2.2 V (for 2.2V components) and V_{CC3} = 3.3 V.*
3. *The CLK signal and the internal PLL are still running but most internal clocking has stopped.*
4. *The CLK signal, the internal PLL, and all internal clocking has stopped.*
5. *This specification applies to components using a CLK frequency of 66 MHz.*
6. *This specification applies to components using a CLK frequency of 95 MHz.*
7. *This specification applies to components using a CLK frequency of 100 MHz.*
8. *This specification applies to components using a CLK frequency of 96.2 MHz.*

7.5 Power and Grounding

Power Connections

The Mobile AMD-K6-III-P processor is a dual voltage device. Two separate supply voltages are required: V_{CC2} and V_{CC3}. V_{CC2} provides the core voltage for the Mobile AMD-K6-III-P processor and V_{CC3} provides the I/O voltage. See “Electrical Data” on page 69 for the value and range of V_{CC2} and V_{CC3}.

There are 28 V_{CC2}, 32 V_{CC3}, and 68 V_{SS} pins on the Mobile AMD-K6-III-P processor. (See “Pin Description Diagrams” on page 83 for all power and ground pin designations.) The large number of power and ground pins are provided to ensure that the processor and package maintain a clean and stable power distribution network.

For proper operation and functionality, all V_{CC2}, V_{CC3}, and V_{SS} pins must be connected to the appropriate planes in the circuit board. The power planes have been arranged in a pattern to simplify routing and minimize crosstalk on the circuit board. The isolation region between two voltage planes must be at least 0.254mm if they are in the same layer of the circuit board. (See Figure 18 on page 75.) In order to maintain low-impedance current sink and reference, the ground plane must never be split.

Although the Mobile AMD-K6-III-P processor has two separate supply voltages, there are no special power sequencing requirements. The best procedure is to minimize the time between which V_{CC2} and V_{CC3} are either both on or both off.

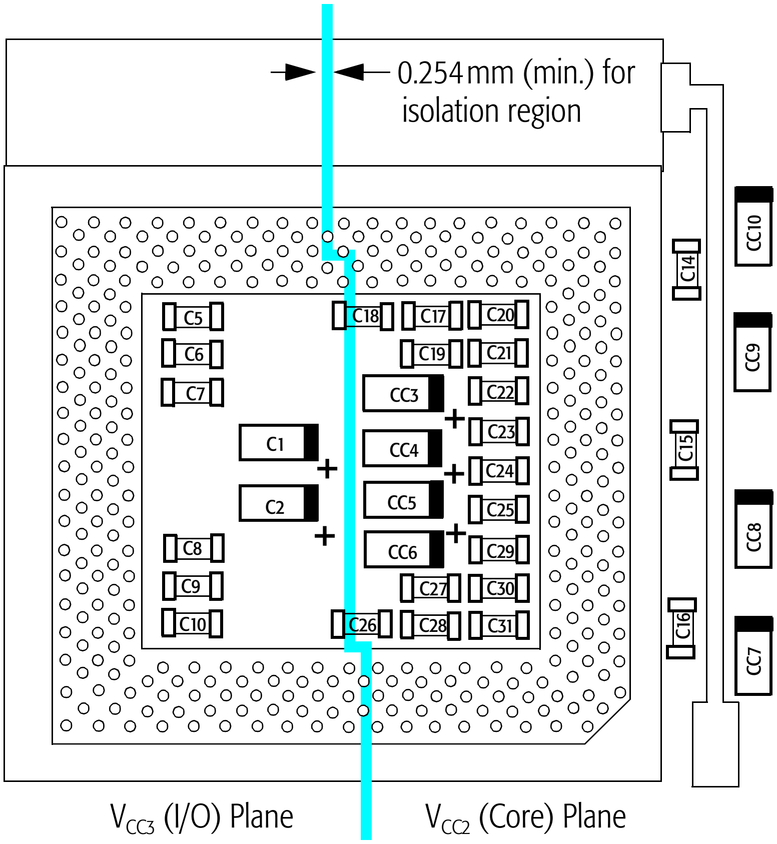


Figure 18. Suggested Component Placement

Decoupling Recommendations

In addition to the isolation region mentioned in "Power Connections" on page 74, adequate decoupling capacitance is required between the two system power planes and the ground plane to minimize ringing and to provide a low-impedance path for return currents. Suggested decoupling capacitor placement is shown in Figure 18.

Surface mounted capacitors should be used as close as possible to the processor to minimize resistance and inductance in the lead lengths while maintaining minimal height. For recommendations regarding the value, quantity, and location of the capacitors illustrated in Figure 18, see the *Mobile AMD-K6® Processor Power Supply Application Note*, order# 22495.

Pin Connection Requirements

For proper operation, the following requirements for signal pin connections must be met:

- Do not drive address and data signals into large capacitive loads at high frequencies. If necessary, use buffer chips to drive large capacitive loads.
- Leave all NC (no-connect) pins unconnected.
- Unused inputs should always be connected to an appropriate signal level.
  - Active Low inputs that are not being used should be connected to V_{CC3} through a 20k-ohm pullup resistor.
  - Active High inputs that are not being used should be connected to GND through a pulldown resistor.
- Reserved signals can be treated in one of the following ways:
  - As no-connect (NC) pins, in which case these pins are left unconnected
  - As pins connected to the system logic as defined by the industry-standard Super7 and Socket 7 interface
  - Any combination of NC and Socket 7 pins
- Keep trace lengths to a minimum.

8 Thermal Design

8.1 Package Thermal Specifications

The Mobile AMD-K6-III-P processor operating specifications call for the case temperature (T_C) to be in the range of 0°C to 80°C. The ambient temperature (T_A) is not specified as long as the case temperature is not violated. The case temperature must be measured on the top center of the package. Table 27 shows the Mobile AMD-K6-III-P processor thermal specifications.

Table 27. Package Thermal Specifications

T_C Case Temperature	Maximum Design Power					
	2.0 V and 2.2 V Components					
	350 MHz	366 MHz	380 MHz	400 MHz	433 MHz	450 MHz
0°C – 80°C	16.00 W					

Figure 19 on page 78 shows the thermal model of a processor with a passive thermal solution. The case-to-ambient temperature (T_{CA}) can be calculated from the following equation:

$$\begin{aligned} T_{CA} &= P_{MAX} \bullet \theta_{CA} \\ &= P_{MAX} \bullet (\theta_{IF} + \theta_{SA}) \end{aligned}$$

Where:

P_{MAX} = Maximum Power Consumption
θ_{CA} = Case-to-Ambient Thermal Resistance
θ_{IF} = Interface Material Thermal Resistance
θ_{SA} = Sink-to-Ambient Thermal Resistance

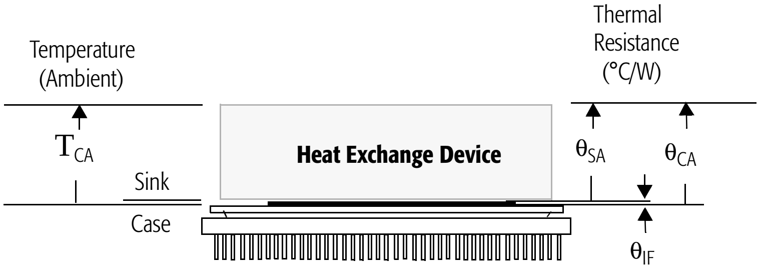


Figure 19. Thermal Model

Figure 20 illustrates the case-to-ambient temperature (T_{CA}) in relation to the power consumption (X-axis) and the thermal resistance (Y-axis). If the power consumption and case temperature are known, the thermal resistance (θ_{CA}) requirement can be calculated for a given ambient temperature (T_A) value.

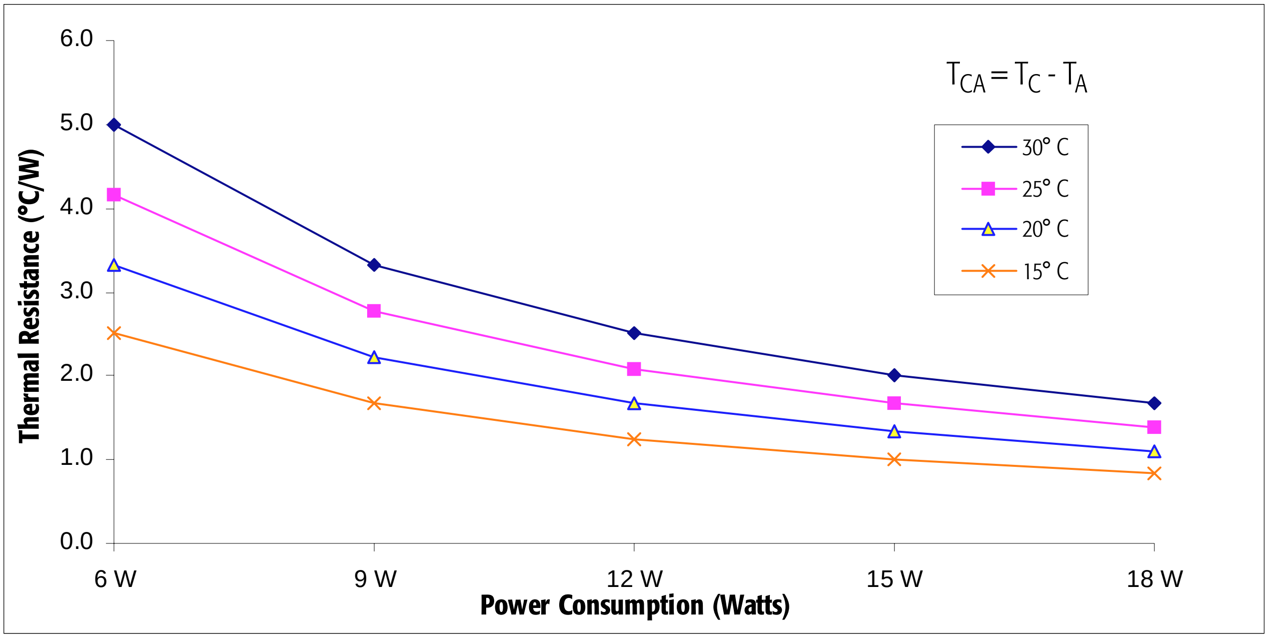


Figure 20. Power Consumption versus Thermal Resistance

The thermal resistance of a heatsink is determined by the heat dissipation surface area, the material and shape of the heatsink, and the airflow volume across the heatsink. In general, the larger the surface area the lower the thermal resistance.

The required thermal resistance of a heatsink (θ_{SA}) can be calculated using the following example:

If:

$T_C = 80°C$
$T_A = 55°C$
$P_{MAX} = 16.00W$

Then:

$$\theta_{CA} \le \left(\frac{T_C - T_A}{P_{MAX}}\right) = \frac{25°C}{16.00W} = 1.56\,(°C/W)$$

Thermal grease is recommended as interface material because it provides the lowest thermal resistance (approx. 0.20°C/W). The required thermal resistance (θ_{SA}) of the heat sink in this example is calculated as follows:

$$\theta_{SA} = \theta_{CA} - \theta_{IF} = 1.56 - 0.20 = 1.36(°C/W)$$

Heat Dissipation Path

Figure 21 illustrates the heat dissipation path of the processor. Due to the lower thermal resistance between the processor die junction and case, most of the heat generated by the processor is transferred from the top surface of the case. The small amount of heat generated from the bottom side of the processor where the processor socket blocks the convection can be safely ignored.

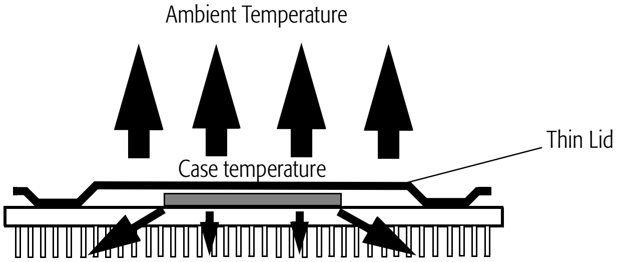


Figure 21. Processor's Heat Dissipation Path

Measuring Case Temperature

The processor case temperature is measured to ensure that the thermal solution meets the processor's operational specification. This temperature should be measured on the top center of the package where most of the heat is dissipated. Figure 22 shows the correct location for measuring the case temperature. If a heatsink is installed while measuring, the thermocouple must be installed into the heatsink via a small hole drilled through the heatsink base (for example, 1/16 of an inch). The thermocouple is then attached to the base of the heatsink and the small hole filled using thermal epoxy, allowing the tip of the thermocouple to touch the top of the processor case.

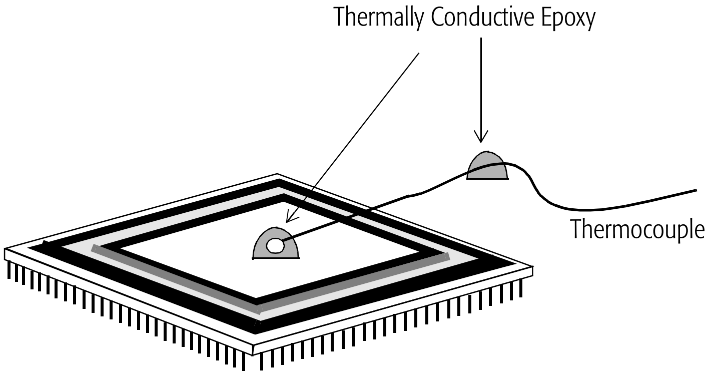


Figure 22. Measuring Case Temperature

For more information on thermal design considerations, see the *AMD-K6® Thermal Solution Design Application Note*, order# 21085.

9 Package Specifications

9.1 321-Pin Staggered CPGA Package Specification

Table 28. 321-Pin Staggered CPGA Package Specification

Symbol	Millimeters		Inches		Notes
	Min	Max	Min	Max	
A	49.28	49.78	1.940	1.960	
B	45.59	45.85	1.795	1.805	
C	31.01	32.89	1.221	1.295	
D	44.90	45.10	1.768	1.776	
E	2.91	3.63	0.115	0.143	
F	1.30	1.52	0.051	0.060	
G	3.05	3.30	0.120	0.130	
H	0.43	0.51	0.017	0.020	
M	2.29	2.79	0.090	0.110	
N	1.14	1.40	0.045	0.055	
d	1.52	2.29	0.060	0.090	
e	1.52	2.54	0.060	0.100	
f	–	0.13	–	0.005	Flatness

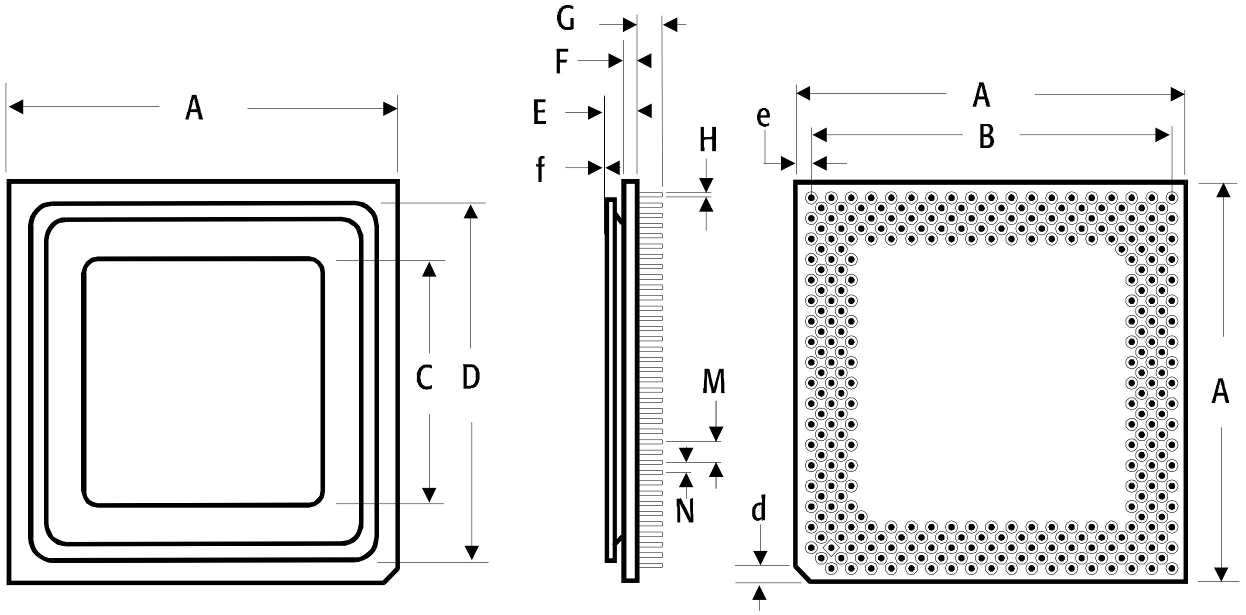


Figure 23. 321-Pin Staggered CPGA Package Specification

10 Pin Description Diagrams

●	Control/Parity Pins	○	Address Pins
⏚	V_{SS} Pins	T	Test Pins
▲	V_{CC2} Pins	⊘	NC, INC (Internal No Connect) Pins
△	V_{CC3} Pins	⊗	RSVD (Reserved) Pins
□	Data Pins	●	Chip Positioning Key Pin

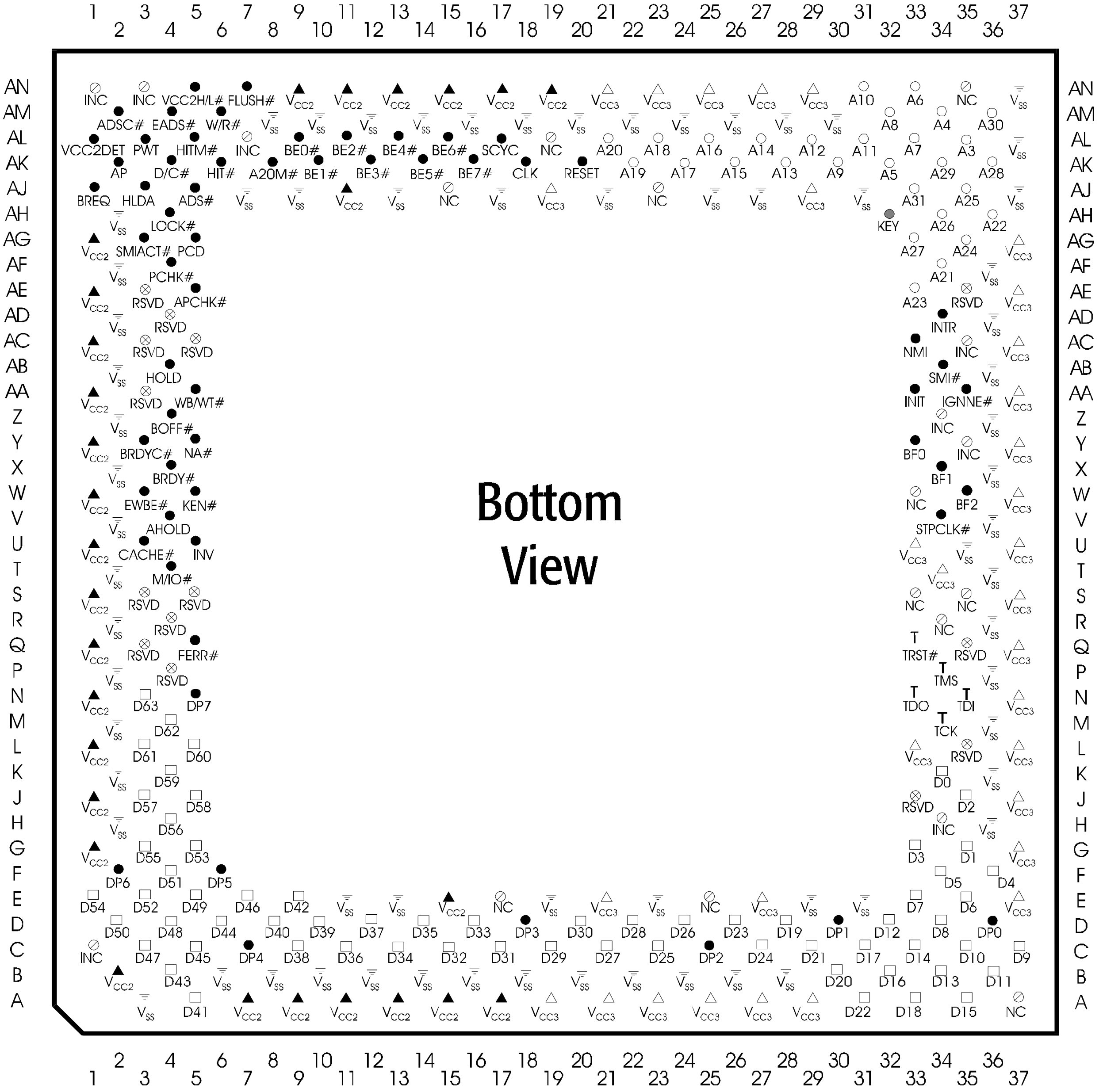

1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37
2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36
AN AM AL AK AJ AH AG AF AE AD AC AB AA Z Y X W V U T S R Q P N M L K J H G F E D C B A
AN: INC INC VCC2H/L# FLUSH# VCC2 VCC2 VCC2 VCC2 VCC2 VCC2 VCC3 VCC3 VCC3 VCC3 VCC3 A10 A6 NC VSS
AM: ADSC# EADS# W/R# VSS VSS VSS VSS VSS VSS VSS VSS VSS VSS VSS VSS A8 A4 A30
AL: VCC2DET PWT HITM# INC 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# VSS VSS VCC2 VSS NC VSS VCC3 VSS NC VSS VSS VCC3 VSS A31 A25 VSS
AH: VSS LOCK# KEY A26 A22
AG: VCC2 SMIACT# PCD A27 A24 VCC3
AF: VSS PCHK# A21 VSS
AE: VCC2 RSVD APCHK# A23 RSVD VCC3
AD: VSS RSVD INTR VSS
AC: VCC2 RSVD RSVD NMI INC VCC3
AB: VSS HOLD SMI# VSS
AA: VCC2 RSVD WB/WT# INIT IGNNE# VCC3
Z: VSS BOFF# INC VSS
Y: VCC2 BRDYC# NA# BF0 INC VCC3
X: VSS BRDY# BF1 VSS
W: VCC2 EWBE# KEN# NC BF2 VCC3
V: VSS AHOLD STPCLK# VSS
U: VCC2 CACHE# INV VCC3 VSS VCC3
T: VSS M/IO# VCC3 VSS
S: VCC2 RSVD RSVD NC NC VCC3
R: VSS RSVD NC VSS
Q: VCC2 RSVD FERR# TRST# RSVD VCC3
P: VSS RSVD TMS VSS
N: VCC2 D63 DP7 TDO TDI VCC3
M: VSS D62 TCK VSS
L: VCC2 D61 D60 VCC3 RSVD VCC3
K: VSS D59 D0 VSS
J: VCC2 D57 D58 RSVD D2 VCC3
H: VSS D56 INC VSS
G: VCC2 D55 D53 D3 D1 VCC3
F: DP6 D51 DP5 D5 D4
E: D54 D52 D49 D46 D42 VSS VSS VCC2 NC VSS VCC3 VSS NC VCC3 VSS VSS D7 D6 VCC3
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: VCC2 D43 VSS VSS VSS VSS VSS VSS VSS VSS VSS VSS VSS VSS D20 D16 D13 D11
A: VSS D41 VCC2 VCC2 VCC2 VCC2 VCC2 VCC2 VCC3 VCC3 VCC3 VCC3 VCC3 VCC3 D22 D18 D15 NC
Bottom View

Figure 24. Mobile AMD-K6®-III-P Processor Bottom-Side View

- ● Control/Parity Pins
- ⏚ V_{SS} Pins
- ▲ V_{CC2} Pins
- △ V_{CC3} Pins
- □ Data Pins
- ○ Address Pins
- T Test Pins
- ⊘ NC, INC (Internal No Connect) Pins
- ⊗ RSVD (Reserved) Pins
- ● Chip Positioning Key Pin

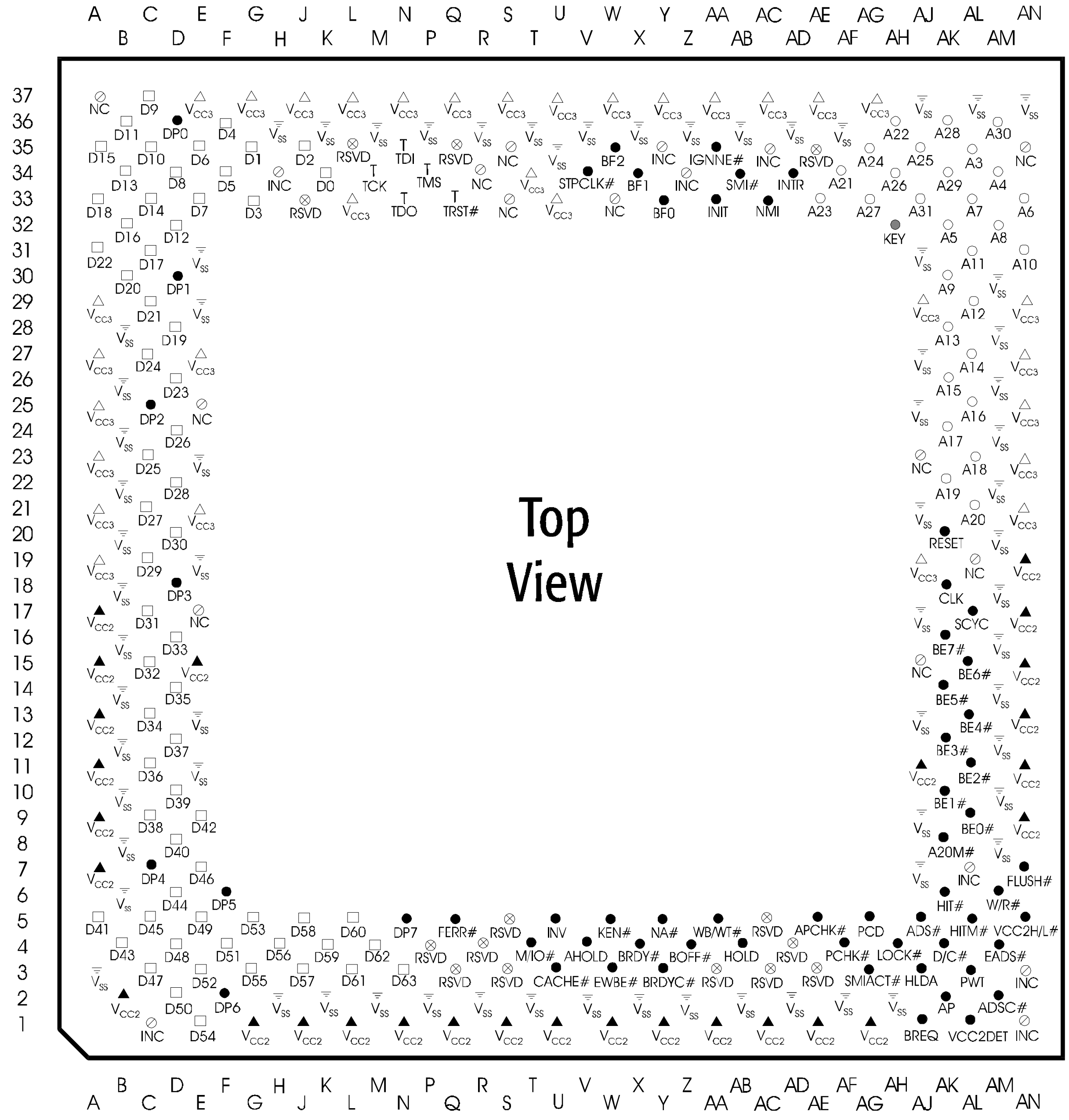

A C E G J L N Q S U W Y AA AC AE AG AJ AL AN
B D F H K M P R T V X Z AB AD AF AH AK AM
37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
NC D9 VCC3 VCC3 VCC3 VCC3 VCC3 VCC3 VCC3 VCC3 VCC3 VCC3 VCC3 VCC3 VCC3 VCC3 VSS VSS VSS
D11 DP0 D4 VSS VSS VSS VSS VSS VSS VSS VSS VSS VSS VSS VSS A22 A28 A30
D15 D10 D6 D1 D2 RSVD TDI RSVD NC VSS BF2 INC IGNNE# INC RSVD A24 A25 A3 NC
D13 D8 D5 INC D0 TCK TMS NC VCC3 STPCLK# BF1 INC SMI# INTR A21 A26 A29 A4
D18 D14 D7 D3 RSVD VCC3 TDO TRST# NC VCC3 NC BF0 INIT NMI A23 A27 A31 A7 A6
D16 D12 KEY A5 A8
D22 D17 VSS VSS A11 A10
D20 DP1 A9 VSS
VCC3 D21 VSS VCC3 A12 VCC3
VSS D19 A13 VSS
VCC3 D24 VCC3 VSS A14 VCC3
VSS D23 A15 VSS
VCC3 DP2 NC VSS A16 VCC3
VSS D26 A17 VSS
VCC3 D25 VSS NC A18 VCC3
VSS D28 A19 VSS
VCC3 D27 VCC3 VSS A20 VCC3
VSS D30 RESET VSS
VCC3 D29 VSS VCC3 NC VCC2
VSS DP3 CLK VSS
VCC2 D31 NC VSS SCYC VCC2
VSS D33 BE7# VSS
VCC2 D32 VCC2 NC BE6# VCC2
VSS D35 BE5# VSS
VCC2 D34 VSS VSS BE4# VCC2
VSS D37 BE3# VSS
VCC2 D36 VSS VCC2 BE2# VCC2
VSS D39 BE1# VSS
VCC2 D38 D42 VSS BE0# VCC2
VSS D40 A20M# VSS
VCC2 DP4 D46 VSS INC FLUSH#
VSS D44 DP5 HIT# W/R#
D41 D45 D49 D53 D58 D60 DP7 FERR# RSVD INV KEN# NA# WB/WT# RSVD APCHK# PCD ADS# HITM# VCC2H/L#
D43 D48 D51 D56 D59 D62 RSVD RSVD M/IO# AHOLD BRDY# BOFF# HOLD RSVD PCHK# LOCK# D/C# EADS#
VSS D47 D52 D55 D57 D61 D63 RSVD RSVD CACHE# EWBE# BRDYC# RSVD RSVD RSVD SMIACT# HLDA PWT INC
VCC2 D50 DP6 VSS VSS VSS VSS VSS VSS VSS VSS VSS VSS VSS VSS AP ADSC#
INC D54 VCC2 VCC2 VCC2 VCC2 VCC2 VCC2 VCC2 VCC2 VCC2 VCC2 VCC2 VCC2 VCC2 BREQ VCC2DET INC
Top View

Figure 25. Mobile AMD-K6®-III-P Processor Top-Side View

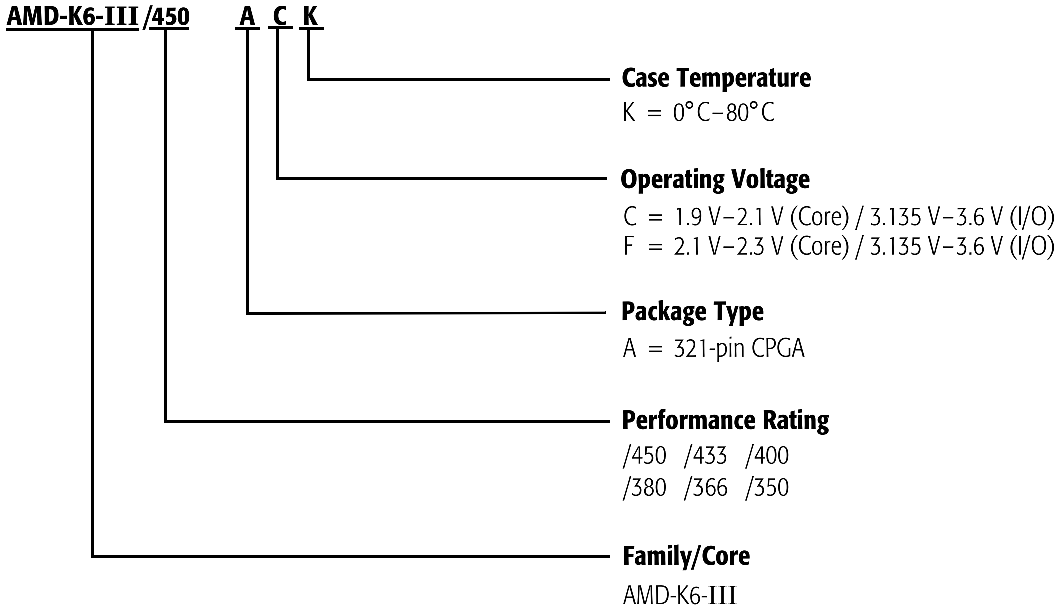
10.1 Pin Designations by Functional Grouping

Address		Data		Control		Test		NC	V_{CC2}	V_{CC3}	V_{SS}	
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	A20M#	AK-08	TCK	M-34	A-37	A-07	A-19	A-03	AM-20
A4	AM-34	D1	G-35	ADS#	AJ-05	TDI	N-35	E-17	A-09	A-21	B-06	AM-22
A5	AK-32	D2	J-35	ADSC#	AM-02	TDO	N-33	E-25	A-11	A-23	B-08	AM-24
A6	AN-33	D3	G-33	AHOLD	V-04	TMS	P-34	R-34	A-13	A-25	B-10	AM-26
A7	AL-33	D4	F-36	APCHK#	AE-05	TRST#	Q-33	S-33	A-15	A-27	B-12	AM-28
A8	AM-32	D5	F-34	BE0#	AL-09			S-35	A-17	A-29	B-14	AM-30
A9	AK-30	D6	E-35	BE1#	AK-10			W-33	B-02	E-21	B-16	AN-37
A10	AN-31	D7	E-33	BE2#	AL-11	**Parity**		AJ-15	E-15	E-27	B-18	
A11	AL-31	D8	D-34	BE3#	AK-12			AJ-23	G-01	E-37	B-20	
A12	AL-29	D9	C-37	BE4#	AL-13			AL-19	J-01	G-37	B-22	
A13	AK-28	D10	C-35	BE5#	AK-14	AP	AK-02	AN-35	L-01	J-37	B-24	
A14	AL-27	D11	B-36	BE6#	AL-15	DP0	D-36		N-01	L-33	B-26	
A15	AK-26	D12	D-32	BE7#	AK-16	DP1	D-30		Q-01	L-37	B-28	
A16	AL-25	D13	B-34	BF0	Y-33	DP2	C-25	**INC**	S-01	N-37	E-11	
A17	AK-24	D14	C-33	BF1	X-34	DP3	D-18		U-01	Q-37	E-13	
A18	AL-23	D15	A-35	BF2	W-35	DP4	C-07		W-01	S-37	E-19	
A19	AK-22	D16	B-32	BOFF#	Z-04	DP5	F-06	C-01	Y-01	T-34	E-23	
A20	AL-21	D17	C-31	BRDY#	X-04	DP6	F-02	H-34	AA-01	U-33	E-29	
A21	AF-34	D18	A-33	BRDYC#	Y-03	DP7	N-05	Y-35	AC-01	U-37	E-31	
A22	AH-36	D19	D-28	BREQ	AJ-01			Z-34	AE-01	W-37	H-02	
A23	AE-33	D20	B-30	CACHE#	U-03			AC-35	AG-01	Y-37	H-36	
A24	AG-35	D21	C-29	CLK	AK-18			AL-07	AJ-11	AA-37	K-02	
A25	AJ-35	D22	A-31	D/C#	AK-04			AN-01	AN-09	AC-37	K-36	
A26	AH-34	D23	D-26	EADS#	AM-04			AN-03	AN-11	AE-37	M-02	
A27	AG-33	D24	C-27	EWBE#	W-03				AN-13	AG-37	M-36	
A28	AK-36	D25	C-23	FERR#	Q-05				AN-15	AJ-19	P-02	
A29	AK-34	D26	D-24	FLUSH#	AN-07			**RSVD**	AN-17	AJ-29	P-36	
A30	AM-36	D27	C-21	HIT#	AK-06				AN-19	AN-21	R-02	
A31	AJ-33	D28	D-22	HITM#	AL-05			J-33		AN-23	R-36	
		D29	C-19	HLDA	AJ-03			L-35		AN-25	T-02	
		D30	D-20	HOLD	AB-04			P-04		AN-27	T-36	
		D31	C-17	IGNNE#	AA-35			Q-03		AN-29	U-35	
		D32	C-15	INIT	AA-33			Q-35			V-02	
		D33	D-16	INTR	AD-34			R-04			V-36	
		D34	C-13	INV	U-05			S-03			X-02	
		D35	D-14	KEN#	W-05			S-05			X-36	
		D36	C-11	LOCK#	AH-04			AA-03			Z-02	
		D37	D-12	M/IO#	T-04			AC-03			Z-36	
		D38	C-09	NA#	Y-05			AC-05			AB-02	
		D39	D-10	NMI	AC-33			AD-04			AB-36	
		D40	D-08	PCD	AG-05			AE-03			AD-02	
		D41	A-05	PCHK#	AF-04			AE-35			AD-36	
		D42	E-09	PWT	AL-03						AF-02	
		D43	B-04	RESET	AK-20						AF-36	
		D44	D-06	SCYC	AL-17			**KEY**			AH-02	
		D45	C-05	SMI#	AB-34						AJ-07	
		D46	E-07	SMIACT#	AG-03						AJ-09	
		D47	C-03	STPCLK#	V-34			AH-32			AJ-13	
		D48	D-04	VCC2DET	AL-01						AJ-17	
		D49	E-05	VCC2H/L#	AN-05						AJ-21	
		D50	D-02	W/R#	AM-06						AJ-25	
		D51	F-04	WB/WT#	AA-05						AJ-27	
		D52	E-03								AJ-31	
		D53	G-05								AJ-37	
		D54	E-01								AL-37	
		D55	G-03								AM-08	
		D56	H-04								AM-10	
		D57	J-03								AM-12	
		D58	J-05								AM-14	
		D59	K-04								AM-16	
		D60	L-05								AM-18	
		D61	L-03									
		D62	M-04									
		D63	N-03									

11 Ordering Information

Standard Products

AMD standard mobile products are available in several operating ranges. The ordering part number (OPN) is formed by a combination of the elements below.



AMD-K6-III /450 A C K

Case Temperature
K = 0°C–80°C

Operating Voltage
C = 1.9 V–2.1 V (Core) / 3.135 V–3.6 V (I/O)
F = 2.1 V–2.3 V (Core) / 3.135 V–3.6 V (I/O)

Package Type
A = 321-pin CPGA

Performance Rating
/450 /433 /400
/380 /366 /350

Family/Core
AMD-K6-III

Table 29. Valid Ordering Part Number Combinations

OPN	Package Type	Operating Voltage	Case Temperature
AMD-K6-III/450ACK	321-pin CPGA	1.9V–2.1V (Core) 3.135V–3.6V (I/O)	0°C–80°C
AMD-K6-III/433ACK	321-pin CPGA	1.9V–2.1V (Core) 3.135V–3.6V (I/O)	0°C–80°C
AMD-K6-III/400ACK	321-pin CPGA	1.9V–2.1V (Core) 3.135V–3.6V (I/O)	0°C–80°C
AMD-K6-III/380AFK	321-pin CPGA	2.1V–2.3V (Core) 3.135V–3.6V (I/O)	0°C–80°C

Note:

This table 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.

Table 29. Valid Ordering Part Number Combinations (continued)

OPN	Package Type	Operating Voltage	Case Temperature
AMD-K6-III/366AFK	321-pin CPGA	2.1V–2.3V (Core) 3.135V–3.6V (I/O)	0°C–80°C
AMD-K6-III/350AFK	321-pin CPGA	2.1V–2.3V (Core) 3.135V–3.6V (I/O)	0°C–80°C

Note:

This table 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.

Index