AMD-K6® Processor Technical Brief

The Superior Engine for Windows® Computing.

AMD's sixth-generation AMD-K6® processor brings advanced, six-issue RISC86® superscalar performance to mainstream PCs running both 16-bit and 32-bit code, enabling leading-edge performance on both the Microsoft® Windows® 95 and Windows NT™ operating systems, as well the installed base of x86 software. The AMD-K6 processor's Socket 7-compatible bus interface also allows PC manufacturers and resellers to leverage today's mature, cost-effective infrastructure (motherboards, chipsets, power supplies, and thermal designs) to quickly bring superior price/performance PC systems to market.

AMD-K6® Processor Technical Features and Innovations

- Sixth-generation performance competitive with the Pentium® II processor
- High performance on both the Windows 95 and Windows NT operating systems
- Advanced, six-issue RISC86® superscalar microarchitecture
- Seven parallel execution units
- Multiple sophisticated x86-to-RISC86 instruction decoders
- Advanced two-level branch prediction
  - Speculative execution
  - Full out-of-order execution
  - Register renaming and data forwarding
- Large 64-Kbyte on-chip L1 caches
  - 32-Kbyte instruction cache plus predecode cache
  - 32-Kbyte writeback dual-ported data cache
  - MESI protocol for cache coherency

AMD K6™
MMX™ ENHANCED PROCESSOR

- High-performance IEEE 754-compatible floating-point unit (FPU)
- High-performance, industry-standard MMX™ instructions
- Fully compatible system management mode (SMM)
- Socket 7 compatibility enabling PC manufacturers and resellers to leverage low-cost infrastructure and system designs
- Available in Ceramic Pin Grid Array (CPGA) package (Socket 7 compatible) using innovative C4 flip-chip technology
- Manufactured using AMD's state-of-the-art 0.35-micron, five-layer-metal silicon process at AMD's Fab 25 wafer fabrication facility

Fully Compatible with Leading Operating Systems and All x86 Software

The AMD-K6 processor architecture is fully x86 binary code compatible and executes high-performance, industry-standard MMX™ instructions. In designing the AMD-K6 processor, AMD confirmed robust compatibility by completing thorough evaluations with industry-standard tools and third-party compatibility verification. AMD's extensive experience with four generations of x86 processors has been integrated into the AMD-K6 processor to provide complete compatibility with Windows 95, Windows 3.x, Windows NT, MS-DOS, OS/2, Novell® NetWare®, Unix, Solaris, Vines, and other leading operating systems and applications.

Innovative RISC86® Microarchitecture

The AMD-K6 processor's RISC86 microarchitecture features a decoupled decode/execution superscalar design that provides enhanced sixth-generation performance and full x86 binary software compatibility. State-of-the-art design techniques include multiple x86 instruction decode, single-clock internal RISC operations, out-of-order execution, data forwarding, speculative execution, and register renaming. The AMD-K6 processor contains parallel decoders, a centralized RISC86 operation scheduler, and seven execution units that support superscalar operation of x86 instructions. These elements are packed into a highly efficient six-stage pipeline.

AMD's innovative RISC86 microarchitecture implements the x86 instruction set by internally decoding x86 instructions into RISC86 operations that directly support the x86 instruction set while adhering to the RISC performance principles of fixed-length encoding, regularized instruction fields, and a large register set. The RISC86

microarchitecture enables higher processor core performance and promotes straightforward extensibility in future designs. Rather than directly executing complex x86 instructions, which have lengths of 1 to 15 bytes, the AMD-K6 MMX™ enhanced processor executes the simpler, fixed-length RISC86 opcodes, while maintaining instruction coding efficiencies found in x86 programs.

The AMD-K6 processor's advanced branch prediction logic implements an 8,192-entry branch history table, a branch target cache, and a return address stack. These design techniques combine to deliver a prediction rate better than 95 percent.

Decoders. The x86 instruction decoding begins before the on-chip 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 x86 instructions, in the instruction cache, to be used later by the decoders. The decoders translate up to two x86 instructions per clock into RISC86 operations. These instructions are categorized into three decode types:

- ***Short decoders*** - Decodes the most commonly used x86 instructions
- ***Long decoders*** - Decodes the semi-common and commonly used instructions
- ***Vector decoders*** - Decodes uncommon, complex x86 instructions.

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. The 24-operation buffer size is optimized for the efficient use of the processor's six-stage RISC86 pipeline and seven parallel execution units. The scheduler accepts up to four RISC86 operations at a time from the decoders. The ICU can simultaneously issue up to six RISC86 operations per clock to the execution units.

Registers. When managing the 24 RISC86 operations, the scheduler uses 48 physical registers contained within the RISC86 microarchitecture. These registers are located in a general register file and are grouped as 24 general registers, plus 24 renaming registers.

Branch Logic. The AMD-K6 processor uses dynamic branch logic to minimize delays due to the branch instructions common in x86 software. The processor's sophisticated dynamic branch logic consists of a branch history/prediction table, a data branch target cache, and a return address stack. The processor implements a two-level branch prediction scheme based on an 8,192-entry branch history table, which stores prediction information used

to predict conditional branches. Since the branch history table does not store predicted target addresses, special address Arithmetic Logic Units (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 supplies the first 16 bytes of target instructions to the decoders when the branches are predicted.

Cache, Instruction Prefetch, and Predecode Bits

The AMD-K6 processor's writeback L1 cache features a separate 32-Kbyte instruction cache and a 32-Kbyte data cache with two-way set associativity. The cache lines are prefetched from main memory using an efficient pipelined burst transaction. As the instruction cache is filled, each instruction byte is analyzed for instruction boundaries using predecoding logic. This technique enables the decoders to efficiently decode multiple instructions in a single pipeline stage.

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

Cache Misses. If an instruction or data cache line required for execution does not reside in the processor's L1 cache, the processor performs a burst cache-line fill from memory. To maximize efficiency of this operation, the processor identifies which of the four quadwords in the cache-line contains the required data or instruction. That quadword is the first to be returned to the L1 cache, thus enabling the processor to continue execution as soon as possible. This technique of varying the burst order improves performance by minimizing execution latency after an L1 cache miss.

Prefetching. The AMD-K6 performs cache prefetching for sector replacements only. As a result, the required cache line is filled first, followed by a prefetch of the second cache line. 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.

Predecode Bits. Decoding of x86 instructions is particularly difficult because these variable-length instructions can be from 1 to 15 bytes long. Predecode logic supplies the predecode bits associated with each instruction

byte. Among other things, the predecode bits indicate the number of bytes to the start of the next x86 instruction. These bits are stored in an extended instruction cache beside each x86 instruction byte. The predecode bits are passed with the instruction bytes to the decoders where they assist with parallel x86 instruction decoding, thus improving the decoding bandwidth.

Instruction Fetch and Decode

Instruction Fetch. The AMD-K6 processor can fetch up to 16 bytes per clock out of the instruction cache or branch target cache. The fetched information goes into a 16-byte instruction buffer that feeds directly into the decoders. Fetching can occur along a single execution stream with up to seven outstanding branches taken. Instruction fetch logic can retrieve any 16 contiguous bytes of information within a 32-byte boundary. No additional penalty occurs 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.

Instruction Decode. The decode logic is designed to decode multiple x86 instructions per clock cycle. 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-format internal instructions, and most execute in a single clock. RISC86 operations combine to perform every function of the x86 instruction set. Some x86 instructions are decoded into as few as zero RISC86 operations or one RISC86 operation. More complex x86 instructions are decoded into several RISC86 operations.

The AMD-K6 processor uses a combination of decoders to convert x86 instructions into RISC86 operations. The hardware includes four decoders:

- ***Two parallel short decoders*** - These translate the most commonly used x86 instructions into zero, one, or two RISC86 operations each. They are also designed to decode up to two x86 instructions per clock.
- ***Long decoder*** - This handles commonly used x86 instructions that can be represented in four or fewer RISC86 operations.
- ***Vectoring decoder*** - This handles all other translations in concert with RISC86 operation sequences fetched from an on-chip ROM.

All of the common, and a few of the uncommon, floating-point instructions are hardware decoded as short

decodes. This decode generates a RISC86 floating-point operation and, optionally, an associated floating-point or store operation. Floating-point or ESC (Escape) instruction decode is only allowed in the first short decoder, but non-ESC instructions (excluding MMX™ instructions) can be decoded simultaneously by the second short decoder.

All MMX™ instructions are hardware decoded as short decodes. This MMX™ instruction decode generates a RISC86 MMX™ operation and, optionally, an associated MMX™ load or store operation. MMX™ instruction decode is only allowed in the first short decode, but instructions other than MMX™ and ESC instructions can be decoded simultaneously by the second short decoder.

Centralized Scheduler

The scheduler is the heart of the AMD-K6 processor. It contains the logic needed to manage out-of-order execution, data forwarding, register renaming, simultaneous issue and retirement of multiple RISC86 operations, and speculative execution. The scheduler's RISC86 operation buffer can hold up to 24 operations. The scheduler can simultaneously issue a RISC86 operation to any available execution unit (store, load, branch, integer, integer/multimedia, or floating point). In total, the scheduler can issue up to six and retire up to four RISC86 operations per clock.

The scheduler and its operation buffer can examine an x86 instruction window equal to 12 x86 instructions at one time. This advantage stems from the fact that the scheduler operates on the RISC86 operations in parallel and allows the AMD-K6 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.

Execution Units

The AMD-K6 processor contains seven independent execution units, each capable of handling the RISC86 operations.

- ***Load Unit*** - performs data memory reads with a two-stage pipeline; data is available from this unit after two clocks.
- ***Store Unit*** - performs data writes and register calculations with a two-stage pipeline; data memory and register writes from stores are available after one clock.
- ***Integer X Unit*** - operates on ALU operations, multiplies, divides, shifts, and rotates.
- ***Multimedia Unit*** - executes all MMX™ instructions.
- ***Integer Y Unit*** - operates on the basic word and double-word ALU operations.
- ***Floating-Point Unit*** - executes all floating-point instructions.
- ***Branch Unit*** - resolves conditional branches after they have been evaluated.

Branch-Prediction Logic

The AMD-K6 processor's sophisticated branch logic is designed to minimize or hide the impact of changes in program flow. Branches in x86 code fit two categories: unconditional branches (which always change program flow) and conditional branches (which may or may not divert program flow). When a conditional branch is not taken, the processor continues decoding and executing the next instructions in memory. Typical applications have up to 10 percent unconditional branches and another 10-20 percent conditional branches. The AMD-K6 branch logic has been designed to handle this type of program behavior and its effects on instruction execution (i.e., stalls due to delayed instruction fetching and draining of the pipeline).

Branch History Table. The AMD-K6 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 AMD-K6 processor's built-in dynamic branch-prediction mechanism. A two-level adaptive history algorithm is implemented in an 8,192-entry branch history table, which stores executed branch information, predicts individual branches, and predicts the behavior of groups of branches. To accommodate this large branch history table, the AMD-K6 processor does not store predicted target addresses; instead, the branch target addresses are calculated on the fly using ALUs during the decode stage.

Branch Target Cache. To avoid a one-clock fetch penalty with a branch prediction, a built-in branch target cache supplies the first 16 bytes of instructions directly to the instruction buffer. 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 percent.

Return Address Stack. The return address stack is designed to optimize CALL and RET pairs. To save space, software is typically compiled with subroutines that are frequently called from various places in a program. Entry into the subroutine occurs with the execution of a CALL instruction. When the processor encounters a RET instruction, 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. This unit enables efficient speculative execution, allowing the processor to execute instructions beyond conditional branches before knowing whether the branch prediction was correct. The AMD-K6 processor does not permanently update the x86 registers or memory locations until all speculatively executed conditional branch instructions are resolved. The AMD-K6 processor can support up to seven outstanding branches.

The AMD-K6® Enhanced Processor

The Superior Engine for Windows Computing. The AMD-K6 processor is the latest generation in AMD's long line of x86 processors. In fact, AMD has designed, manufactured, and delivered more than 50 million Windows-compatible processors in the last five years.

The AMD-K6 processor combines state-of-the-art RISC86 superscalar architecture, full x86 and Windows OS compatibility, and industry-standard MMX™ instructions to deliver leading-edge performance on both 16-bit and 32-bit code. Feature for feature, clock for clock, the AMD-K6 processor is the smart choice for today's and tomorrow's PC designs.

For more information on the AMD-K6 processor line, call (408) 749-5703 or (800) 222-9323 (U.S. only).