Advanced Micro Devices, Inc.
x86-64™ Technology White Paper

ADVANCED MICRO DEVICES, INC.
One AMD Place
Sunnyvale, CA 94088

Contact:

Sean Cleveland
Public Relations
(408) 749-5980
sean.cleveland@amd.com

Abstract

The need for 64-bit computing is driven by applications that address large amounts of data and memory, such as high-performance servers, database management systems, CAD tools, and digital content creation tools. Although this increased demand on the high-end along with technical advances is making 64-bit a reality, it is anticipated that the industry as a whole will not fully embrace the 64-bit world entirely for many more years.

Existing 32-bit environments and applications will continue to serve a majority of users for quite sometime, although they too will require continual improvements in processor and system performance to continue. Many x86 workstation and server users are now facing the dilemma of how to transition to 64-bit computing, yet maintain their existing knowledge base and investment in the existing code base and tools. The challenge for processor manufacturers is to find a way to offer customers all the advantages of 64-bit processing in a market friendly fashion while making the conversion from 32-bit efficient and inexpensive. Unfortunately, the 64-bit solutions proposed by some processor manufacturers leave customers facing a potentially disruptive and ultimately expensive transition to the new architectures.

AMD's strategy of extending the x86 architecture for 64-bit computing is a straightforward alternative to total conversion using incompatible instruction sets. AMD processors including the x86-64™ technology will permit platform suppliers, developers, corporate MIS departments and consumers to transition to 64-bit environments gradually, while continuing to run 32-bit applications without incurring performance penalties. By providing a smoother migration to 64-bit computing, AMD's strategy is designed to save its customers billions of dollars in software re-development and deployment costs.

Technology at a Cross-Roads

Not too many years ago, industry pundits predicted that microprocessors based on RISC (Reduced Instruction Set Computer) computing, an architecture that reduces chip complexity by using simpler instructions, would forever out-perform current x86 processors based on CISC (Complex Instruction Set Computer) architectures, which use microcode to execute very comprehensive instructions.

Yet today, CISC processors essentially match RISC chips in integer performance, and are quickly catching up on floating point performance. As can be seen in the chart below, x86 performance has nearly erased a 50% SPECint benchmarking deficit when compared to RISC performance over the past four years. SPECint is a set of eight compute-intensive integer/non-floating point benchmarks used to measure typical processor performance.
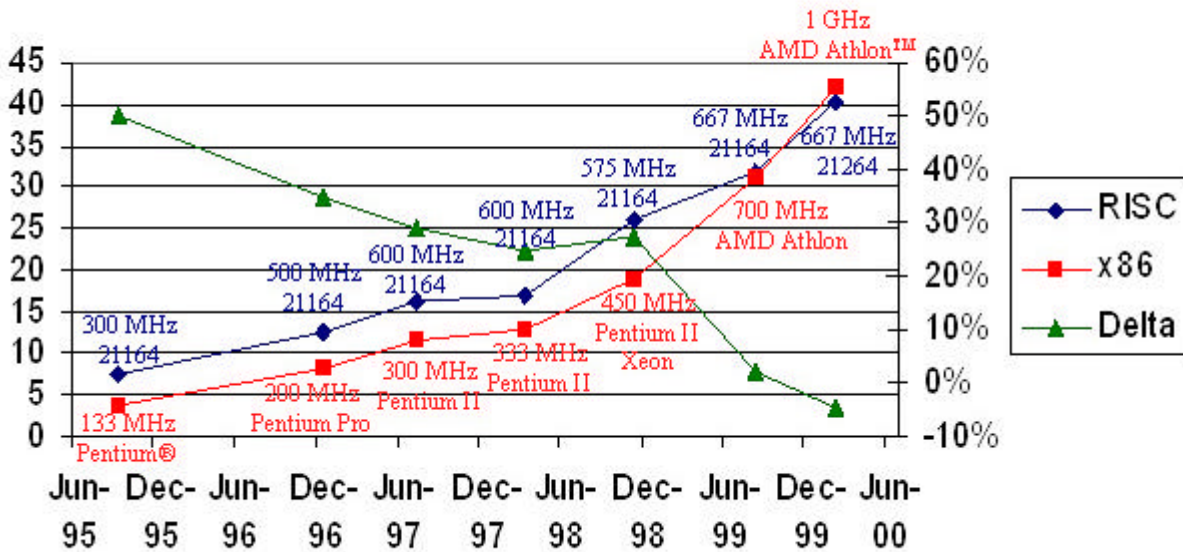
Source: Microprocessor Report and Standard Performance Evaluation Corporation.

Future gains in performance are anticipated to have less to do with instruction sets than with implementation techniques. Therefore, any decision adopting 64-bit RISC or VLIW (Very Long Instruction Word) based processors over 64-bit CISC based processors will likely offer little if any advantage in overall processor performance.

Better Solution: AMD x86-64 Architecture

While 64-bit computing may offer vital features for some applications, the transition from 32- to 64-bit raises notable issues for several audiences:

- Processor manufacturers must decide how to provide 64-bit functionality while preserving x86 compatibility and performance for their existing 32-bit installed base.

- Platform suppliers must consider the impact on motherboard design, operating system availability and stability; and the costs of providing technical support for two (32-bit and 64-bit) systems.

- Operating system and software providers must decide whether to incur the costs of developing 32-bit and 64-bit applications in parallel, or whether to abandon their 32-bit development efforts and begin programming for 64-bit environments.

- MIS managers must plan for a major IT transition, replacing existing 32-bit applications and re-training in-house development and support staff, or risk being "left behind" with incompatible 32-bit technologies.

- End-users must choose between upgrading their operating environments and applications to 64-bit, or retaining their 32-bit applications at current performance and functionality levels.

To date, processor manufacturers have attempted to solve some of these transition issues with less than satisfactory results. Software emulation provides 32-bit compatibility but lacks native performance. And requiring the user to purchase and install a separate, 32-bit co-processor entails additional investments in money and administration.

New 64-bit VLIW-based (Very Long Instruction Word) architectures, such as Intel's IA-64, force users to undergo a laborious transition process to a new instruction set to achieve full native performance. This is because IA-64 offers no native x86 compatibility, which means existing 32-bits applications are not anticipated to run with leading edge performance on IA-64 technology based processors. Also, the adoption of an entirely new instruction set architecture like IA-64 requires the development of an entirely new tool chain including compilers, debuggers, assemblers, and profilers, whereas x86-64 tools can be modified from existing code.

AMD's 64-bit strategy is one that allows the latest in processor innovation to be brought to the existing installed base of 32-bit applications and operating systems, while establishing an installed base of systems that are 64-bit capable.

It is also the only solution that allows companies and individuals to enable 64-bit computing at their own pace, as hardware and software support becomes available. This allows the industry to advance software development on all fronts, with x86 compatibility, while providing the capability of delivering 64-bit performance benefits. Any other 64-bit solution does not fully preserve the industries investment in software.

AMD takes a more practical and less disruptive approach to the challenge of 64-bit computing. Key extensions to the reliable, proven, high-performance x86 instruction set not only preserve full compatibility between 32- and 64 bit environments, they also enable 32-bit applications to move up the performance ladder right along with other advances in technology.

AMD does not intend to force its platform suppliers, software developers, and enterprise customers to make the tough choice between protecting their investments in 32-bit technology and moving into the 64-bit world of the future. Instead, AMD's strategy of forward compatibility and future performance allows customers to continue using 32-bit applications and then seamlessly transition to 64-bit when they are ready, and hardware and software support becomes available.

This strategy enables AMD and its platform partners to leverage the latest in 64-bit microprocessor innovations without disrupting their current 32-bit installed base. For operating system and application providers, development can continue uninterrupted and unburdened by transition issues and support for the 64-bit extensions in the x86-64 technology can be provided at their own pace. In addition, the AMD x86-64 architecture permits existing 16- and 32-bit x86 code to take full advantage of all the performance advances of new processor designs without the overhead of software or hardware emulation.

AMD x86-64 Architecture™	Other Vendor's 64-Bit Solutions
Fully x86 compatible.	Instruction sets are NOT x86 compatible.
No performance sacrifice for either 32-bit or 64-bit computing	Compromises 32-bit performance. Smaller, less sophisticated, x86 engine causes speed penalty for legacy code. Future development is focused on 64-bit only performance.
Not forced to move to a new architecture. Seamless transition based on user's timeframe.	Forces the costly transition of many 32-bit applications that do not require 64-bits.
Continued use of existing 32-bit applications, tools, and knowledge base.	Doubles investment: 2 instruction sets, 2 operating environments, 2 application binaries, 2 development and support teams.
Full support for 16-, 32-, and 64-bit applications running concurrently.	Support for 16- and 32-bit apps only through emulation software or hardware.
32-bit code runs unchanged. Designed to be easy to port applications that might benefit from 64-bit address space.	Must port 32-bit applications for full speed execution. During transition years, must manage 2 code bases.

Extending x86 for the 64-bit World

AMD's 64-bit strategy is a straightforward extension to the current x86 instruction set. The methodology is similar to that used to extend the 16-bit architecture to 32-bits.

The x86-64 architecture extends the standard x86 architecture by adding a new mode called *long mode*. Long mode is enabled by a global control bit called LMA (for Long Mode Active). When LMA is disabled, the processor operates as a standard x86 processor, and is compatible with all existing 16- and 32-bit operating systems and applications. When Long Mode is activated (LMA=1) the 64-bit processor extensions are enabled. This allows the system to auto configure according to the capabilities of the machine and the processor.

64-Bit OS (LMA)	64-Bit App (CSD L Bit)	Data Size (CSD D Bit)	CPU Mode
0	X	0	**Standard 16-Bit Mode**
0	X	1	**Standard 32-Bit Mode**
1	0	0	**Compatibility 16-Bit Mode**
1	0	1	**Compatibility 32-Bit Mode**
1	1	0	**64-bit Mode**
1	1	1	**Reserved**

Long Mode consists of two sub modes: *64-bit mode* and *compatibility mode*. The new modes are encoded using two flags in the code segment descriptor. The first flag is the existing "D" bit that controls the size of operands. A second bit called the "L bit", is a previously unused bit (bit 53) in the CS descriptor, is used for determining if specific applications are 64-bit enabled or are run in compatibility mode.

If long mode is active, CS.L = 1, and CS.D = 0, the processor is then set to run in 64-bit mode. With this encoding, default operand size is 32-bits and default address size is 64-bits. Using instruction prefixes, the default operand size can be overridden to 64-bits or 16-bits, and the default address size can be overridden to 32-bits.

If long mode is active and CS.L=0 then the processor is set to run in compatibility mode. Compatibility mode maintains binary compatibility with applications written in 16- and 32-bit x86 modes. An operating system running natively in Long Mode can run existing 16- and 32-bit applications by simply clearing the L bit in the code segment descriptor of the applications.

The AMD x86-64™ Architecture

The AMD x86-64 architecture extends the legacy x86 architecture by introducing two major features: a 64-bit extension called *long mode*, and register extensions.

Mode		Operating System Required	Application Recompile Required	Defaults[1]			
				Address Size (bits)	Operand Size (bits)	Register Extensions[2]	GPR Width (bits)
Long Mode[3]	64-Bit Mode	New 64-bit OS	yes	64	32	yes	64
	Compatibility Mode		no	32		no	32
				16			
Legacy Mode[4]		Legacy 32-bit or 16-bit OS	no	32	32	no	32
				16	16		

1. Defaults can be overridden in most modes using an instruction prefix or system control bit.
2. Register extensions includes eight new GPRs and eight new XMM registers (also called SSE registers).
3. Long mode supports only x86 protected mode. It does not support x86 real mode or virtual-8086 mode. Also, it does not support task switching.
4. Legacy mode supports x86 real mode, virtual-8086 mode, and protected mode.

Long Mode

Long mode consists of two sub-modes: *64-bit mode*, and *compatibility mode*. Compatibility mode supports binary compatibility with existing 16-bit and 32-bit applications under a 64-bit operating system. In addition to long mode, the architecture also supports a pure x86 *legacy mode*, which preserves binary compatibility not only with existing 16-bit and 32-bit applications but also with existing 16-bit and 32-bit operating systems.

64-Bit Mode

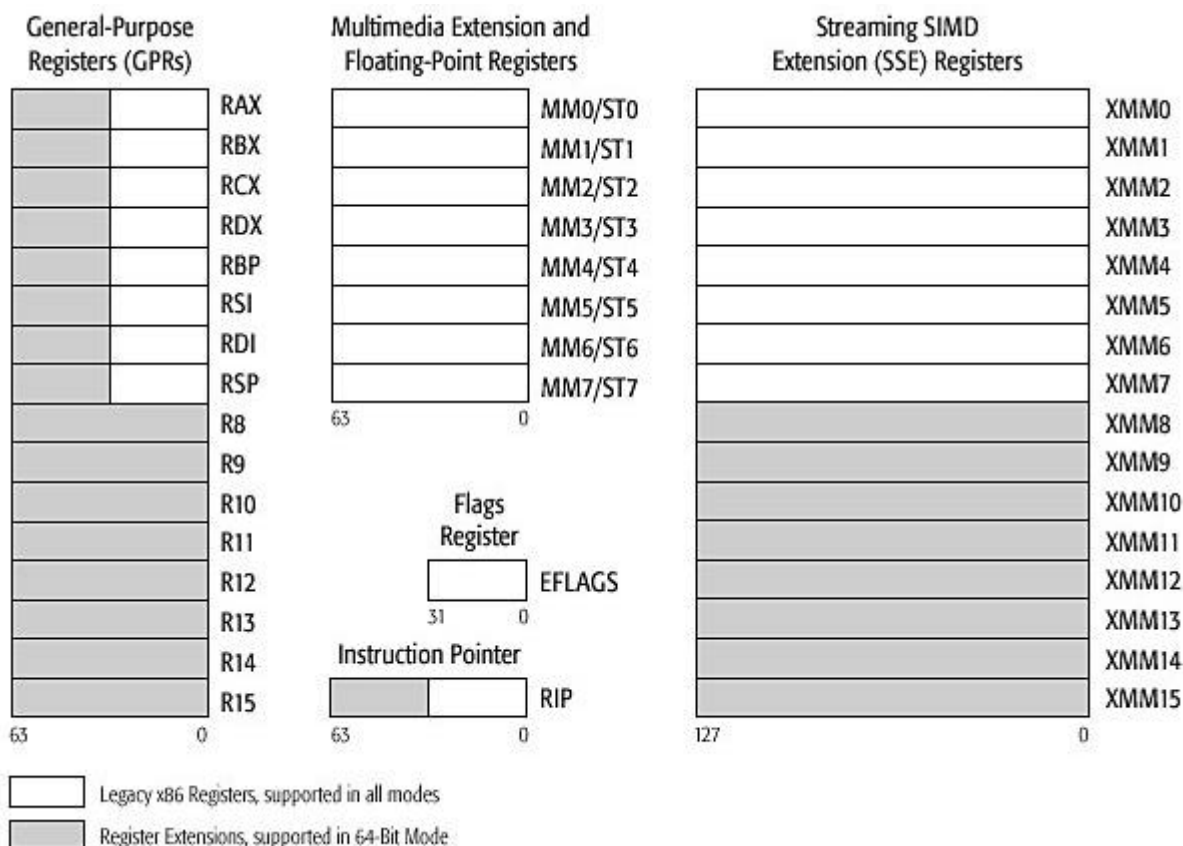64-bit mode supports the following new features:

- 64-bit virtual addresses (implementations can have less).
- Register extensions through a new prefix (REX):
    - Adds eight GPRs (R8–R15).
    - Widens GPRs to 64 bits.
    - Adds eight 128-bit Streaming SIMD Extension (SSE) registers (XMM8–XMM15).
- 64-bit instruction pointer (RIP).

- New RIP-relative data addressing mode.
- Flat address space with single code, data, and stack space.

The default address size is 64-bits, and the default operand size is 32-bits. The defaults can be overridden on an instruction-by-instruction basis using prefixes. A new REX prefix is introduced for specifying 64-bit operand size and the new registers. The mode is enabled by the operating system on an individual code-segment basis.

The new register extensions added via the new REX prefix add eight 64-bit GPRs (R8–R15), eight 128-bit Streaming SIMD Extensions (SSE) registers (XMM8–XMM15), and widens all GPRs and the instruction pointer to 64 bits. The REX prefix also provides a new byte-register capability that makes the least-significant byte of any GPR available for byte operations. This results in a uniform set of byte, word, dword, and qword registers better suited for a compiler's register allocation.

The instruction pointer is also widened to 64 bits.

General-Purpose Registers (GPRs)	Multimedia Extension and Floating-Point Registers	Streaming SIMD Extension (SSE) Registers
RAX	MM0/ST0	XMM0
RBX	MM1/ST1	XMM1
RCX	MM2/ST2	XMM2
RDX	MM3/ST3	XMM3
RBP	MM4/ST4	XMM4
RSI	MM5/ST5	XMM5
RDI	MM6/ST6	XMM6
RSP	MM7/ST7	XMM7
R8	63 0	XMM8
R9		XMM9
R10	Flags Register	XMM10
R11		XMM11
R12	EFLAGS	XMM12
R13	31 0	XMM13
R14	Instruction Pointer	XMM14
R15	RIP	XMM15
63 0	63 0	127 0

Legacy x86 Registers, supported in all modes

Register Extensions, supported in 64-Bit Mode

Because 64-bit mode supports a 64-bit virtual-address space, it requires a 64-bit operating system and tool chain. A few instruction opcodes and prefix bytes are redefined to allow the register extensions and 64-bit addressing.

Register Extensions

To define the addressing logic for the registers, the AMD x86-64 architecture simply extends the addressing scheme currently used for 16- and 32-bit instructions. For example, as shown in the diagram below:
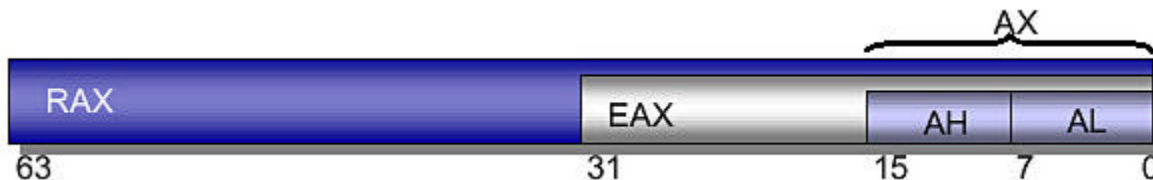
- For 16-bit operations, the two bytes of Register A are addresses as AX
- For 32-bitoperations, the four bytes of Register A are addressed as EAX
- For 64-bitoperations, the eight bytes of Register A are addressed as RAX

In 64-bit Mode, General Purpose Registers (GPRs) are extended to 64-bits. The 64-bit registers are called RAX, RBX, RCX, RDX, RDI, RSI, RBP, RSP, RIP and RFLAGS. The new 64-bit registers overlay and extend the existing registers. In addition, eight new 64-bit GPRs are added for a total of 16 GPRS. The new GPRs are called R8 through R15.

The register extensions also add eight new streaming SIMD registers for a total of 16 SIMD registers. The new SIMD registers are called XMM8 through XMM15.

Segment registers (ES, DS, FS, GS and SS) are ignored in 64-bit Mode. Code segments still exist in 64-bit Mode. The CS is needed to encapsulate the default mode of the processor (16-, 32- or 64-bit mode) as well as the execution privilege level. As noted above, the D bit and L bit are used to specify the default address and operand sizes. The DPL is used for execution privilege checks. Base and limit fields are ignored.

When performing 32-bit operations and the destination register is a GPR, the 32-bit value will be zero-extended into the full 64-bit GPR. 8-bit and 16-bit operations on GPRs preserve all unwritten upper bits. This preserves the 16- and 32-bit semantics for partial-width results. The final step is to simply define a set of instruction prefixes that specify a 64-bit operand size and allow access to the new registers. This is similar to the method used to extend the x86 architecture for other functionality, such as AMD's 3DNow!™ technology. With this strategy, AMD makes it possible for platform suppliers, developers, and other users to use existing toolsets, applications, and knowledge, while providing a smooth migration to 64-bit enabled applications as hardware and software support becomes available.

Compatibility Mode

Compatibility mode allows operating systems to implement binary compatibility with existing 16-bit and 32-bit x86 applications. It allows these applications to run, without recompilation, under a 64-bit operating system in long mode. In compatibility mode, applications can only

access the first 4GBytes of virtual-address space. Standard x86 instruction prefixes toggle between 16-bit and 32-bit address and operand sizes.

As with 64-bit mode, compatibility mode is enabled by the operating system on an individual code-segment basis. Unlike 64-bit mode, however, x86 segmentation functions normally, using 16-bit or 32-bit protected-mode semantics. From the application's viewpoint, compatibility mode looks like a legacy x86 protected-mode environment. From the operating system's viewpoint, address translation, interrupt and exception handling, and system data structures use the 64-bit long mode mechanisms.

Legacy Mode
When LMA=0, the processor is said to be running in legacy mode. None of the 64-bit features are available and the processor operates as a standard x86 processor. Legacy mode is completely compatible with existing 32-bit implementations of the x86 architecture. This includes support for current technologies like segmented memory, and 32-bit GPRs and instruction pointer.

AMD 64-Bit Roadmap

Already performance-leaders for high-volume markets, AMD processors set a clear path for 64-bit processor architectures. Key data and address paths are already 64-bits wide. The x86-64 architecture is designed to have minimal impact on processor die size and no impact on the progression of processor clock speed. Future improvements in the processor core will accelerate both 32-bit and 64-bit applications. Thus, high-performance systems using 64-bit capable processors from AMD are planned to also be among the highest performing 32-bit systems ever built.

AMD is currently collaborating with its key platform suppliers, operating systems providers, and other technology partners for the x86-64 architecture. AMD plans to include 64-bit technology in its next-generation of performance-leading processors code-named "Hammer" to establish an initial base of 64-bit platforms by the end of 2001.

AMD: Advancing the Potential of x86

By extending the x86 core rather than replacing it with a new, entirely different instruction set, AMD makes the transition to a 64-bit world easier, faster, and less expensive for its system partners, developers, and end-user customers. The burden of transitioning to a new architecture is greatly reduced, without limiting the forward compatibility and future performance of existing applications. For platform suppliers and operating systems providers, AMD's x86-64 strategy helps preserve investments in current technology and engineering expertise. Further more, because end-user customers are not faced with a major computing transition, they are anticipated to less likely consider changing their preferred hardware or OS vendor. Offering a solution that provides both compatibility and future performance is a win-win for both suppliers and customers.

For application developers and MIS managers, AMD's strategy allows development to continue on a path of high performance and optimal functionality, regardless of whether the underlying architecture is 32- or 64-bit. Familiar tools, applications, and management protocols are retained until *they* choose to migrate and hardware and software support becomes available. When they do make the transition, they can do so with a minimum investment of time and resources. AMD's x86-64 strategy brings the latest processor innovation to existing 32-bit environments while establishing a clear path for record-breaking levels of performance, scalability, and compatibility.

Final Comments

History is filled with examples of processor revolutions that never were; transitions to new instruction sets so expensive and time consuming that advances in existing architectures passed them by. Advanced Micro Devices will continue to lead the way by advancing the performance and scalability of x86-compatible processors.

AMD Overview

AMD (NYSE: AMD) is a global supplier of integrated circuits for the personal and networked computer and communications markets. AMD produces processors, flash memories, and products for communications and networking applications. The world's second-leading supplier of Windows® compatible processors, AMD has shipped more than 120 million x86 microprocessors, including more than 90 million Windows compatible CPUs. Founded in 1969 and based in Sunnyvale, California, AMD has sales and marketing offices worldwide and manufacturing facilities in Sunnyvale; Austin, Texas; Dresden, Germany; Bangkok, Thailand; Penang, Malaysia; Singapore; and Aizu-Wakamatsu, Japan. AMD had revenues of $2.9 billion in 1999.

Cautionary Statement

This document contains forward-looking statements, which are made pursuant to the safe harbor provisions of the U.S. Private Securities Litigation Reform Act of 1995. Forward-looking statements are generally preceded by words such as "plans," "expects," "believes," "anticipates" or "intends." Investors are cautioned that all forward-looking statements in this document involve risks and uncertainty that could cause actual results to differ materially from current expectations. Forward looking statements in this document include the risks that developers may not support the x86-64 technology and design tools for the technology in a timely manner or at all; that AMD will not successfully implement the technology in its products on a timely basis; and that AMD may not effectively penetrate the enterprise market. We urge investors to review in detail the risks and uncertainties in the Company's filings with the United States Securities Exchange Commission.