



The AMD64™ ISA Value Proposition

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

www.amd.com

Executive Summary

64-bit microprocessors have been in existence since 1992 when DEC introduced the first Alpha processor. Since then Sun, IBM, HP, and Intel all have introduced 64-bit processors with corresponding 64-bit instruction sets. AMD is now entering the world of 64-bit computing with the AMD64 Instruction Set Architecture (ISA). Unlike other 64-bit instruction sets, AMD64 ISA is based on the industry standard x86 instruction set architecture. By building on the x86 architecture, AMD is bringing to market 32-bit x86 compatibility along with the advantages of 64-bit computing.

This paper presents a brief overview of the AMD64 ISA. This is followed by a description of the benefits of the instruction set. The most significant benefits of AMD64 ISA are:

- An extended flat address space which allows programs to address memory beyond 4 GB
- Native 32-bit x86 compatibility which allows 64-bit and 32-bit applications to run on a 64-bit operating system
- Support for 16 GPRs and 16 XMM registers - double the number of registers present in the 32-bit x86 instruction set
- The ease of porting software from 32-bit to 64-bit systems

AMD64 Architecture Overview

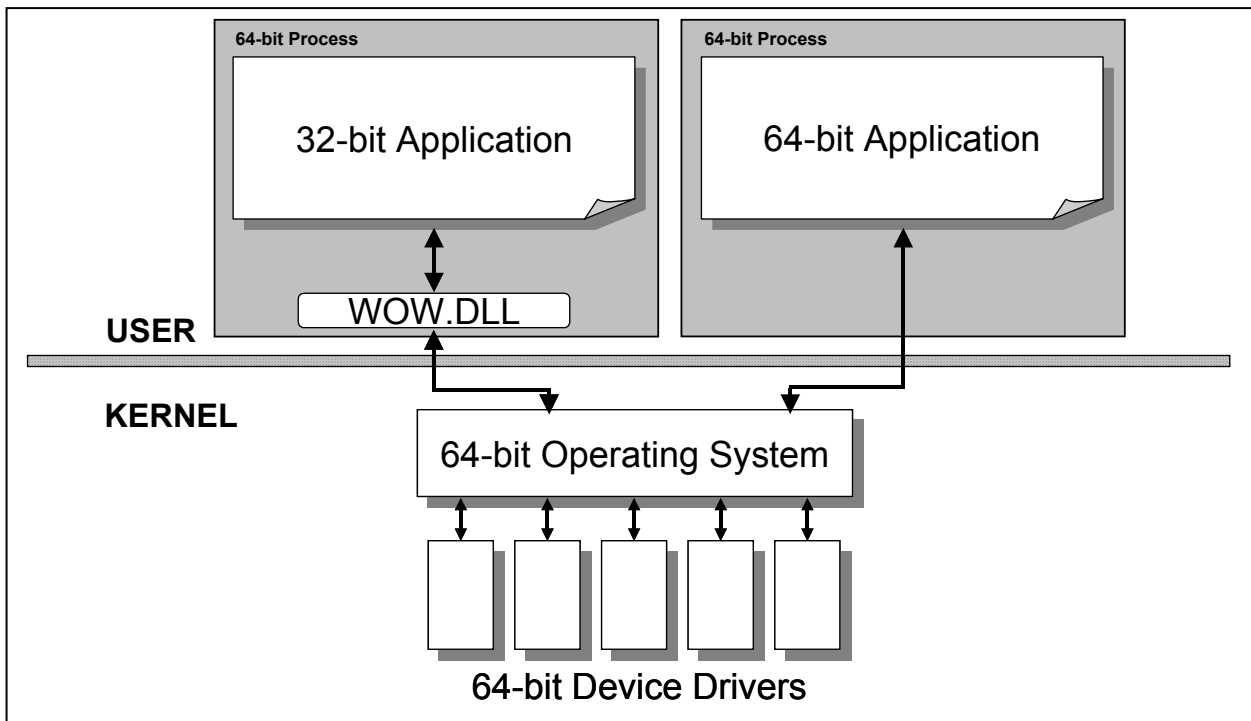
AMD has implemented the AMD64 ISA as an extension to the x86 instruction set architecture. A new execution mode has been added to the processor referred to as “long mode”. Long mode supports running both 32-bit and 64-bit applications under a 64-bit operating system. Long mode has two submodes: 64-bit mode and compatibility mode. 64-bit mode allows the processor to execute 64-bit programs under a 64-bit OS while compatibility mode allows 32-bit programs to execute natively, at full speed in hardware, under a 64-bit operating system. Legacy Mode brings forward all the existing 16-bit and 32-bit x86 operating modes. The AMD64 operating modes are shown in *Table 1* below.

Table 1

Operating Mode		OS Required	Application Recompile Required	Defaults		Register Extensions	Typical GPR Width
				Address Size (bits)	Operand Size (bits)		
Long Mode	64-bit Mode	New 64-bit OS	yes	64	32	yes	64
	Compatibility Mode		no	32		no	32
				16	16		16
Legacy Mode	Protected Mode	Legacy 32-bit OS	no	32	32	no	32
	Virtual-8086 Mode			16	16		
	Real Mode	Legacy 16-bit OS		16	16		16

Most modern operating systems have built-in support for Compatibility Mode. Microsoft® Windows® supports this through the Windows on Windows subsystem (WoW). The WOW.DLL converts function arguments from 32-bit to 64-bit and return values from 64-bit to 32-bit. This conversion is very low overhead as most of the calculations involve adding leading zeros. Address translation is from flat 32-bit to flat 64-bit. *Diagram 1* illustrates how 32-bit and 64-bit applications interact with the Windows operating system.

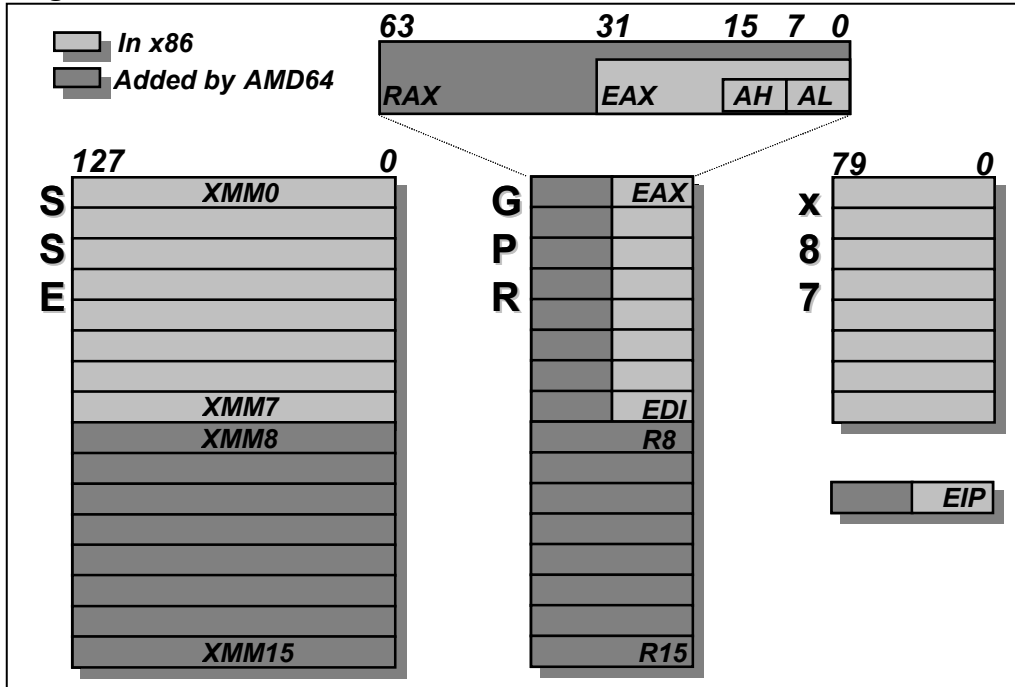
Diagram 1



The 64-bit programming model is actually an enhancement of the 32-bit x86 model designed to support 64-bit programming. The eight GPRs (General Purpose Registers) have been widened to 64-bits to support 64-bit addressing and 64-bit integer math. The EIP has also been widened to support execution of code above the lower 4 GB range. Additionally, eight General Purpose Registers, referred to as the extended GPRs, and eight XMM registers, referred to as the extended XMM registers, have been added. These additional registers are available to programs in 64-bit mode only.

A diagram of the AMD64 ISA register file is shown in *Diagram 2*.

Diagram 2



All x86 instructions have been brought forward into the AMD64 ISA. These instructions have been extended to support 64-bit data types where applicable. Only a handful of other instructions have been added, mostly to support sign extension to 64-bits.

An additional addressing mode has been added called RIP-relative addressing. This addressing mode supports addressing relative to the 64-bit instruction pointer (also called the program counter). The effective address is formed by adding the displacement, a signed 32-bit integer, to the 64-bit RIP of the next instruction. In the legacy x86 architecture, addressing relative to the instruction pointer (IP or EIP) is available only in control transfer instructions. In the 64-bit mode, any instruction that uses ModRM addressing can use RIP-relative addressing.

The binary instruction encodings in the AMD64 ISA are very similar to x86. The major change is an additional override byte, referred to as the REX (Register Extension) prefix, has been added. This single-byte prefix allows the use of an extended GPR register, an extended XMM register, the use of a 64-bit operand (which includes a 64-bit GPR), or accessing one of the extended control and debug registers. At most, a single REX prefix is necessary per instruction.

Additionally, some of the binary instruction encodings have been eliminated to support future instruction set expansion. The x86 instruction set actually supports multiple ways to encode several instructions, an unnecessary redundancy. Some of these instruction encodings have been removed in 64-bit mode to support future enhancements of the instruction set.

The paging mechanism has been extended to provide 64-bit addressing. The paging model is a four-level page table that is a simple extension of the x86 PAE-formatted entries.

Benefits of the AMD64 ISA

Extended memory Addressing

The major reason for going to a 64-bit programming model is to enable programs to access memory outside of the 4GB boundary of 32-bit systems. There are several reasons why this is important.

Computer programs and their data are initially stored on the system's hard disk. The program (or part of a very large program) is loaded into the system's memory by the operating system and the program then loads its data as needed. The operating system will swap in and out data and program code between memory and disk depending on how much memory is required by the program versus how much physical memory is available. Ideally, the program, data, operating system kernel, and libraries will be loaded into physical memory as memory accesses are orders of magnitude faster than even today's fastest disk drives.

Current 32-bit systems support a 4 GB virtual address space that must be shared between the application, library routines, and the operating system kernel. Many applications such as Databases, Digital Content Creation (DCC), Mechanical CAD (MCAD), Electronic Design Automation (EDA), Oil Exploration, and Customer Relationship Management (CRM) are outgrowing the 4 GB limit. They need to manage very large data sets which require access to more virtual address space. 32-bit x86 processors can support larger virtual address space using mechanisms such as Address Windowing Extensions (AWE) in Windows. Programming AWE is very difficult and the performance is lackluster compared to flat 64-bit addressing. The solution is to move to AMD64 which implements a flat 64-bit programming model. The AMD Opteron™ processor supports 48-bits of virtual address in 64-bit mode that will allow applications to address up to 256 Terrabytes of memory. This will remove today's 4GB limit and is designed to allow applications access to an abundant virtual address space. Today's largest transaction-oriented databases are roughly 1TB in size while the existing largest data warehouses are pushing the 10TB limit. It will be several years before databases outgrow the 256 TB limit.

As prices for DRAM continue to drop it is now cost effective to build systems with greater than 4GB of memory. With the price of memory currently below \$250 per Gigabyte of DRAM, it costs less than \$4,000 for 16GB of memory. These systems must be able to address physical memory beyond the 4GB limit. Existing 32-bit processors support this through a mechanism called Physical Address Extensions (PAE). Each page of physical memory present in the system must be mapped through a page table entry. Each page table entry takes up a certain amount of memory space in the operating system kernel. As the size of physical memory increases, the size of the page table increases. Both Linux and Windows have a practical limit of 16GB of physical memory present in a system before coming close to overflowing the one GB allocated to

the kernel space with page table entries. Windows Advanced Server can support up to 64GB of physical memory when applications are restricted to 2GB virtual address space.

As memory prices continue to fall, it will become increasingly cost effective to build systems with greater than 16GB of memory necessitating 64-bit addressing. AMD Opteron supports 40 bits of physical addressing which enables up to 1TB of physical memory.

Bringing 64-bit Computing into the Mainstream

Currently, 64-bit computing is only supported in mid-range and high-end servers and workstations. RISC-based 64-bit servers systems start at roughly \$25K. Intel IA-64 systems start at \$10K and move up from there. This translates to roughly 5% of all servers sold today.

All AMD Opteron and AMD Athlon™ 64 processors will support the AMD64 ISA. These processors are planned to power mobile and desktop computers as well as workstations and servers. This will allow AMD to bring the economies of scale of 32-bit x86 computing to 64-bit computing.

AMD's software and hardware partners perceive this as a huge advantage. This allows them a much bigger market for their 64-bit applications. By lowering the cost of the hardware, it allows the total system cost to the end user to be lower. In addition, the performance of x86-based systems has already begun to outstrip that of proprietary RISC systems. The end result of this is higher performance at a lower price, which is extremely compelling to end users in today's financially constrained environment.

32-bit Compatibility

The major factor facilitating the amazing success of the x86 architecture is that succeeding generations of hardware and operating systems have allowed backwards compatibility of applications. This allows end users to maintain their older applications and migrate to applications that take advantage of the new architecture. The move from 32-bit to 64-bit computing will be similar to the move from 16-bit to 32-bit computing. The 386 processor was introduced in 1986 and was the first x86 processor to support 32-bit computing. Software support for 32-bit computing did not arrive for seven years; it wasn't until 1993 that Windows NT® 3.1 was introduced and 1995 when Windows 95 was introduced. Both these operating systems supported 16-bit compatibility where 16-bit applications ran at full speed in hardware alongside 32-bit applications. ISVs ported their applications to 32-bit as they saw fit. The AMD64 ISA enables this type of adoption pattern for 64-bit computing.

The AMD64 ISA supports all the existing 16-bit and 32-bit x86 operating modes. These are referred to as Legacy Mode as shown in *Table 1*. This allows all processors that implement AMD64 to run legacy 16-bit and 32-bit applications and operating systems unmodified. This presents a very compelling upgrade path for end users where they can purchase an AMD64-based processor system and initially use it as a 32-bit machine, preserving their investment in 32-bit applications and operating systems. They can then upgrade to 64-bit applications and operating systems as necessary at a later

time. This provides a form of investment protection where they have the option later on to upgrade to 64-bit computing.

The AMD64 ISA also supports 32-bit and 64-bit applications coexisting under a 64-bit operating system. More importantly, 32-bit applications will run at full speed in hardware with no emulation. This has huge advantages for software vendors, OEMs, and end users allowing software vendors to initially port only those parts of their software that can take advantage of 64-bit computing and leave other components as 32-bit applications. Porting less code translates into less porting time, less testing, and less performance tuning leading to lower costs and faster time to market for the ISV and OEM. The end user does not have to replace all applications, only those that can take advantage of 64-bit computing. For example, consider a system that is running both a database and an application server. The database application can be ported to 64-bit and the application server can remain as 32-bit. Backward compatibility permits applications to be ported as needed.

The ability to run 32-bit and 64-bit applications concurrently is extremely valuable for software engineers. While developing 64-bit applications, they can use their 32-bit development tools on the same machine they are testing on. Programmers have their favorite editors and other tools that currently are 32-bit applications and perhaps will never be ported to 64-bits. 32-bit compatibility makes it possible for programmers to develop and test their code on the same machine without a reboot, allowing faster development cycles. Certain tools, specifically debuggers, must be native 64-bit applications because of their interaction with the operating system kernel.

Ease of Porting Software from x86 to AMD64 ISA

AMD64 ISA is actually an extension to the 32-bit x86 instruction set. Programmers are familiar with x86 and there is a minimal learning curve. All the programming techniques that programmers have mastered learning x86 are applicable to AMD64 code development. Software developers have spent years learning the x86 instruction set. All of this effort can be leveraged because the AMD64 ISA is so similar to x86. There is already a highly trained workforce needing only to learn a few simple concepts to be up and running programming the instruction set facilitating quicker and more efficient software development cycles.

AMD64 software development tools such as compilers and linkers can be brought to a very robust state quickly due to the similarity of the back-end code generation of AMD64 to x86. In fact, AMD64 is a more elegant target for compilers and linkers due to the addition of the RIP-relative addressing mode, the use of SSE and SSE2 for floating point, and the 64-bit GPR's which enable 64-bit integer math in a straightforward fashion.

As an example of the robustness of the Linux programming tools, IBM ported several million lines of DB2® to SuSE AMD64 Linux in two days. This is significantly faster than the weeks or months that it has taken them to port to other 64-bit platforms. This is a testimony to the robustness of not only the AMD64 tools, but also the AMD Opteron processor and platforms.

Ease of Migration from Proprietary RISC Systems to AMD64

Proprietary UNIX implementations such as AIX and Solaris have been losing market share to both Linux and Windows. The trend will continue as the performance of x86 machines continues to outstrip that of RISC implementations. Windows machines will take market share in such areas as web servers, file servers, and mail servers. Linux also is very strong in these areas. Linux has a strong advantage in that it is similar from a programming perspective to UNIX. This allows an easy migration of software by ISV's and IHV's. Many available enterprise applications are several million lines of code – porting these applications to Linux is quite straightforward because of the similar libraries and threading model. Because the proprietary RISC systems from IBM, HP, and Sun are all 64-bit systems the most direct path from UNIX to Linux is to 64-bit Linux rather than 32-bit Linux. All of these applications need an industry standard 64-bit architecture with which to port.

32-bit Applications Can Run More Efficiently

One possible migration path that an end user may take to 64-bit computing is to migrate their operating systems to AMD64 while running their existing 32-bit applications. There are several advantages to this:

- Operating system robustness can be qualified while holding the application stack constant.
- 32-bit applications can have access to the full 4GB address space. In a 64-bit operating system the kernel will be moved outside the 4GB range allowing the application to use the entire 32-bit address space for code and data.
- More applications can run on a single machine due to more memory becoming available. The more physical memory on the system, the more of each application can be resident in memory, allowing faster context switch time.
- Applications can take advantage of a higher performing OS and drivers. AMD64 ISA supports additional general purpose registers and XMM registers. Additional registers will give the operating system better performance.

Additional Registers

One of the major weaknesses of the x86 instruction set is the lack of general purpose and floating point registers limiting the ability of compilers to generate efficient code. The AMD64 ISA supports 16 GPR's and 16 XMM registers - double the number of registers present in the 32-bit x86 instruction set. This allows the design of higher-performance Application Binary Interfaces (ABI's) in 64-bit operating systems. These ABI's can offer the following advantages:

- Use of SSE and SSE2 operations for single and double-precision arithmetic. This allows elimination of the x87 floating point stack in favor of flat floating point register file which allows compilers to use modern scheduling techniques to generate efficient code.

- Fewer fills and spills of data to and from memory. Because there are more registers, persistent data can be held in registers rather than being written to memory and then read back. This generates more compact code as well as minimizing cache and memory traffic.
- More efficient calling conventions. Up to four function call arguments can be passed in registers rather than pushing and popping arguments onto the stack. This generates more compact code as well as minimizing cache and memory traffic.

64-bit Integer Math

The AMD64 General Purpose Registers (GPR's) are 64-bits wide as shown above in *Diagram 2*. These registers support 64-bit integer math operations such as additions and multiplications. The native width for a pointer in AMD64 is 64-bits which require 64-bit arithmetic for pointer calculations. Many applications will use 32-bit integers as data, both for ease of migration to 64-bits and also to minimize the code size of integer literals. Two applications that can take advantage of native 64-bit arithmetic are:

Encryption algorithms such as SSL require 64-bit multiplies. Performing a single 64-bit integer multiply using 32-bit arithmetic requires several multiplies and additions to compute the final result. Doing the same 64-bit integer multiply using AMD64 takes a single instruction.

Algorithms that perform integer bit packing such as Huffman encoding, which is used in video compression, can be performed more efficiently using 64-bit registers. More data can be packed into a single register than when using a 32-bit register.

Closing Thoughts

AMD64 ISA is a revolutionary design because it extends the already successful industry standard x86 instruction set. The advantages inherent in the design are all based on the similarity of AMD64 to x86. By creating a derivative rather than an entirely new instruction set, AMD plans to bring all the advantages of 64-bit computing along with 32-bit compatibility and ease of porting.

AMD Overview

AMD is a global supplier of integrated circuits for the personal and networked computer and communications markets with manufacturing facilities in the United States, Europe, and Asia. AMD produces microprocessors, flash memory devices, and support circuitry for communications and networking applications. Founded in 1969 and based in Sunnyvale, California, AMD had revenues of \$3.9 billion in 2001. (NYSE: AMD).

© 2003 Advanced Micro Devices, Inc. All rights reserved

AMD, the AMD Arrow logo, AMD Athlon, AMD Opteron and combinations thereof are trademarks of Advanced Micro Devices, Inc. Microsoft and Windows are registered trademarks of Microsoft Corporation in the U.S. and other jurisdictions. Other product names used in this document are for identification purposes only and may be trademarks of their respective companies.