64-bit computing

From Wikipedia, the free encyclopedia
(Redirected from 64-bit)

Bit		
1 4 8 12 16 18 24 31 32 36 48 60 **64** 128 256 512		
Application		
16	32	64
Floating point precision		
x½	x1	x2 x4
Floating point decimal precision		
	32	64 128

In computer architecture, **64-bit computing** is the use of processors that have datapath widths, integer size, and memory addresses widths of 64 bits (eight octets). Also, 64-bit CPU and ALU architectures are those that are based on registers, address buses, or data buses of that size. From the software perspective, 64-bit computing means the use of code with 64-bit virtual memory addresses.

The term *64-bit* is a descriptor given to a generation of computers in which 64-bit processors are the norm. 64 bits is a word size that defines certain classes of computer architecture, buses, memory and CPUs, and by extension the software that runs on them. 64-bit CPUs have existed in supercomputers since the 1970s (Cray-1, 1975) and in RISC-based workstations and servers since the early 1990s, notably the DEC Alpha, Sun UltraSPARC, Fujitsu SPARC64, and IBM RS64 and POWER3 and later POWER microprocessors. In 2003 they were introduced to the (previously 32-bit) mainstream personal computer arena in the form of the x86-64 and 64-bit PowerPC processor architectures and later in 2012[1] even in processors that were before mainly used in smartphones, tablet computers, and embedded systems with the introduction of the AArch64 processor architectures in ARMv8 and on September 10, 2013 with the introduction of the 64-bit ARMv8 Apple A7 SOC, powering the iPhone 5s.

A 64-bit register can store 2^{64} (over 18 quintillion or 1.8×10^{19}) different values. Hence, a processor with 64-bit memory addresses can directly access 2^{64} bytes (=16 exbibytes) of byte-addressable memory.

Without further qualification, a *64-bit computer architecture* generally has integer and addressing registers that are 64 bits wide, allowing direct support for 64-bit data types and addresses. However, a CPU might have external data buses or address buses with different sizes from the registers, even larger (the 32-bit Pentium had a 64-bit data bus, for instance). The term may also refer to the size of low-level data types, such as 64-bit floating-point numbers.

Contents

- 1 Architectural implications
- 2 History
- 3 Limitations of practical processors
- 4 64-bit processor timeline
- 5 64-bit operating system timeline
- 6 64-bit application
  - 6.1 32-bit vs 64-bit
  - 6.2 Pros and cons
  - 6.3 Software availability
- 7 64-bit data models
- 8 Current 64-bit microprocessor architectures
- 9 See also
- 10 Notes
- 11 References
- 12 External links

Architectural implications

Processor registers are typically divided into several groups: *integer*, *floating-point*, *SIMD*, *control*, and often special registers for address arithmetic which may have various uses and names such as *address*, *index* or *base registers*. However, in modern designs, these functions are often performed by more general purpose *integer* registers. In most processors, only integer and/or address-registers can be used to address data in memory; the other types of registers cannot. The size of these registers therefore normally limits the amount of directly addressable memory, even if there are registers, such as floating-point registers, that are wider.

Most high performance 32-bit and 64-bit processors (some notable exceptions are older or embedded ARM and 32-bit MIPS CPUs) have integrated floating point hardware, which is often, but not always, based on 64-bit units of data. For example, although the x86/x87 architecture has instructions capable of loading and storing 64-bit (and 32-bit) floating-point values in memory, the internal floating point data and register format is 80 bits wide, while the general-purpose registers are 32 bits wide. In contrast, the 64-bit Alpha family uses a 64-bit floating-point data and register format (as well as 64-bit integer registers).

History

Most CPUs are designed so that the contents of a single integer register can store the address (location) of any data in the computer's virtual memory. Therefore, the total number of addresses in the virtual memory – the total amount of data the computer can keep in its working area – is determined by the width of these registers. Beginning in the 1960s with the IBM System/360 (which was an exception, in that it used the low order 24 bits of a word for addresses, resulting in a 16 MB [16×1024^2 bytes] address space size), then (amongst many others) the DEC VAX minicomputer in the 1970s, and then with the Intel 80386 in the mid-1980s, a *de facto* consensus developed that 32 bits was a convenient register size. A 32-bit address register meant that 2^{32} addresses, or 4 GB of RAM, could be referenced. At the time these architectures were devised, 4 GB of memory was so far beyond the typical quantities (4 MB) in installations that this was considered to be enough "headroom" for addressing. 4.29 billion addresses were considered an appropriate size to work with for another important reason: 4.29 billion integers are enough to assign unique references to most entities in applications like databases.

Some supercomputer architectures of the 1970s and 1980s used registers up to 64 bits wide. In the mid-1980s, Intel i860[2] development began culminating in a (too late[3] for Windows NT) 1989 release. However, 32 bits remained the norm until the early 1990s, when the continual reductions in the cost of memory led to installations with quantities of RAM approaching 4 GB, and the use of virtual memory spaces exceeding the 4 GB ceiling became desirable for handling certain types of problems. In response, MIPS and DEC developed 64-bit microprocessor architectures, initially for high-end workstation and server machines. By the mid-1990s, HAL Computer Systems, Sun Microsystems, IBM, Silicon Graphics, and Hewlett Packard had developed 64-bit architectures for their workstation and server systems. A notable exception to this trend were mainframes from IBM, which then used 32-bit data and 31-bit address sizes; the IBM mainframes did not include 64-bit processors until 2000. During the 1990s, several low-cost 64-bit microprocessors were used in consumer electronics and embedded applications. Notably, the Nintendo 64[4] and the PlayStation 2 had 64-bit microprocessors before their introduction in personal computers. High-end printers and network equipment, as well as industrial computers, also used 64-bit microprocessors, such as the Quantum Effect Devices R5000. 64-bit computing started to drift down to the personal computer desktop from 2003 onwards, when some models in Apple's Macintosh lines switched to PowerPC 970 processors (termed "G5" by Apple), and AMD released its first 64-bit x86-64 processor.

Limitations of practical processors

In principle, a 64-bit microprocessor can address 16 exabytes of memory. In practice, it is less than that.

For example, the AMD64 architecture as of 2011 allowed 52 bits for physical memory and 48 bits for virtual memory.[5] These limits allow memory sizes of 4 PiB (4×1024^5 bytes) and 256 TiB (256×1024^4 bytes), respectively. A PC cannot currently contain 4 petabytes of memory (due to the physical size of the memory chips) but AMD envisioned large servers, shared memory clusters, and other uses of physical address space that might approach this in the foreseeable future, and the 52-bit physical address provides ample room for expansion while not incurring the cost of implementing 64-bit physical addresses. Similarly, the 48-bit virtual address space was designed to provide more than 65,000 times the 32-bit limit of 4 GB (4×1024^3 bytes), allowing room for later expansion without incurring the overhead of translating full 64-bit addresses.

64-bit processor timeline

1961
: IBM delivers the IBM 7030 Stretch supercomputer, which uses 64-bit data words and 32- or 64-bit instruction words.

1974
: Control Data Corporation launches the CDC Star-100 vector supercomputer, which uses a 64-bit word architecture (previous CDC systems were based on a 60-bit architecture).
: International Computers Limited launches the ICL 2900 Series with 32-bit, 64-bit, and 128-bit two's complement integers; 64-bit and 128-bit floating point; 32-bit, 64-bit and 128-bit packed decimal and a 128-bit accumulator register. The architecture has survived through a succession of ICL and Fujitsu machines. The latest is the Fujitsu Supernova, which emulates the original environment on 64-bit Intel processors.

1976
: Cray Research delivers the first Cray-1 supercomputer, which is based on a 64-bit word architecture and will form the basis for later Cray vector supercomputers.

1983
: Elxsi launches the Elxsi 6400 parallel minisupercomputer. The Elxsi architecture has 64-bit data registers but a 32-bit address space.

1989
: Intel introduces the Intel i860 RISC processor. Marketed as a "64-Bit Microprocessor", it had essentially a 32-bit architecture, enhanced with a 3D Graphics Unit capable of 64-bit integer operations.[6]

1991
: MIPS Technologies produces the first 64-bit microprocessor, the R4000, which implements the MIPS III ISA, the third revision of their MIPS architecture.[7] The CPU is used in SGI graphics workstations starting with the IRIS Crimson. Kendall Square Research deliver their first KSR1 supercomputer, based on a proprietary 64-bit RISC processor architecture running OSF/1.

1992
: Digital Equipment Corporation (DEC) introduces the pure 64-bit Alpha architecture which was born from the PRISM project.[8]

1993
: Atari introduces the Atari Jaguar video game console, which includes some 64-bit wide data paths in its architecture.[9]

1994
: Intel announces plans for the 64-bit IA-64 architecture (jointly developed with Hewlett-Packard) as a successor to its 32-bit IA-32 processors. A 1998 to 1999 launch date is targeted.

1995
: Sun launches a 64-bit SPARC processor, the UltraSPARC.[10] Fujitsu-owned HAL Computer Systems launches workstations based on a 64-bit CPU, HAL's independently designed first-generation SPARC64. IBM releases the A10 and A30 microprocessors, 64-bit PowerPC AS processors.[11] IBM also releases a 64-bit AS/400 system upgrade, which can convert the operating system, database and applications.

1996
: Nintendo introduces the Nintendo 64 video game console, built around a low-cost variant of the MIPS

R4000. HP releases an implementation of the 64-bit 2.0 version of their PA-RISC processor architecture, the PA-8000.[12] IBM released the high-end, 4-way SMP, multi-chip version called Muskie, A25 or A30 in AS/400 systems.

1997

IBM releases the RS64 line of 64-bit PowerPC/PowerPC AS processors.

1998

IBM releases the POWER3 line of full-64-bit PowerPC/POWER processors.[13]

1999

Intel releases the instruction set for the IA-64 architecture. AMD publicly discloses its set of 64-bit extensions to IA-32, called x86-64 (later branded AMD64).

2000

IBM ships its first 64-bit z/Architecture mainframe, the zSeries z900. z/Architecture is a 64-bit version of the 32-bit ESA/390 architecture, a descendant of the 32-bit System/360 architecture.

2001

Intel ships its IA-64 processor line, after repeated delays in getting to market. Now branded Itanium and targeting high-end servers, sales fail to meet expectations.

2003

AMD introduces its Opteron and Athlon 64 processor lines, based on its AMD64 architecture which is the first x86-based 64-bit processor architecture. Apple also ships the 64-bit "G5" PowerPC 970 CPU produced by IBM. Intel maintains that its Itanium chips would remain its only 64-bit processors.

2004

Intel, reacting to the market success of AMD, admits it has been developing a clone of the AMD64 extensions named IA-32e (later renamed EM64T, then yet again renamed to Intel 64). Intel ships updated versions of its Xeon and Pentium 4 processor families supporting the new 64-bit instruction set.

VIA Technologies announces the Isaiah 64-bit processor.[14]

2006

Sony, IBM, and Toshiba begin manufacturing of the 64-bit Cell processor for use in the PlayStation 3, servers, workstations, and other appliances.

2011

ARM Holdings announces ARMv8, the first 64-bit version of the ARM architecture.[15]

2012

ARM announced their Cortex-A53 and Cortex-A57 cores on 30 October 2012.[1][16]

2013

Apple announces the iPhone 5s, claiming it to be "the first 64−bit smartphone in the world"; it uses their A7 ARMv8-based system-on-a-chip.

64-bit operating system timeline

1985

Cray releases UNICOS, the first 64-bit implementation of the Unix operating system.[17]

1993

DEC releases the 64-bit DEC OSF/1 AXP Unix-like operating system (later renamed Tru64 UNIX) for its systems based on the Alpha architecture.

1994

Support for the MIPS R8000 processor is added by Silicon Graphics to the IRIX operating system in release 6.0.

1995

DEC releases OpenVMS 7.0, the first full 64-bit version of OpenVMS for Alpha. First 64-bit Linux distribution for the Alpha architecture is released.[18]

1996

Support for the MIPS R4x00 processors in 64-bit mode is added by Silicon Graphics to the IRIX operating system in release 6.2.

1998

Sun releases Solaris 7, with full 64-bit UltraSPARC support.

2000

IBM releases z/OS, a 64-bit operating system descended from MVS, for the new zSeries 64-bit mainframes; 64-bit Linux on zSeries follows the CPU release almost immediately.

2001

Linux becomes the first OS kernel to fully support x86-64 (on a simulator, as no x86-64 processors had been released yet).[19]

2001

Microsoft releases Windows XP 64-Bit Edition for the Itanium's IA-64 architecture, although it was able to run 32-bit applications through an execution layer.

2003

Apple releases its Mac OS X 10.3 "Panther" operating system which adds support for native 64-bit integer arithmetic on PowerPC 970 processors.[20] Several Linux distributions release with support for AMD64. Microsoft announces plans to create a version of its Windows operating system to support the AMD64 architecture, with backwards compatibility with 32-bit applications. FreeBSD releases with support for AMD64.

2005

On January 31, Sun releases Solaris 10 with support for AMD64 and EM64T processors. On April 29, Apple releases Mac OS X 10.4 "Tiger" which provides limited support for 64-bit command-line applications on machines with PowerPC 970 processors; later versions for Intel-based Macs supported 64-bit command-line applications on Macs with EM64T processors. On April 30, Microsoft releases Windows XP Professional x64 Edition for AMD64 and EM64T processors.

2006

Microsoft releases Windows Vista, including a 64-bit version for AMD64/EM64T processors that retains 32-bit compatibility. In the 64-bit version, all Windows applications and components are 64-bit, although many also have their 32-bit versions included for compatibility with plugins.

2007

Apple releases Mac OS X 10.5 "Leopard", which fully supports 64-bit applications on machines with PowerPC 970 or EM64T processors.

2009

Microsoft releases Windows 7, which, like Windows Vista, includes a full 64-bit version for AMD64/Intel 64 processors; most new computers are loaded by default with a 64-bit version. It also releases Windows Server 2008 R2, which is the first 64-bit only operating system released by Microsoft. Apple releases Mac OS X 10.6, "Snow Leopard," which ships with a 64-bit kernel for AMD64/Intel64 processors, although only certain recent models of Apple computers will run the 64-bit kernel by default. Most applications bundled with Mac OS X 10.6 are now also 64-bit.[20]

2011

Apple releases Mac OS X 10.7, "Lion," which runs the 64-bit kernel by default on supported machines. Older machines that are unable to run the 64-bit kernel run the 32-bit kernel, but, as with earlier releases, can still run 64-bit applications; Lion does not support machines with 32-bit processors. Nearly all applications bundled with Mac OS X 10.7 are now also 64-bit, including iTunes.

2013

Apple releases iOS 7, which, on machines with AArch64 processors, has a 64-bit kernel that supports 64-bit applications.

64-bit application

32-bit vs 64-bit

A change from a 32-bit to a 64-bit architecture is a fundamental alteration, as most operating systems must be extensively modified to take advantage of the new architecture, because that software has to manage the actual memory addressing hardware.[21] Other software must also be ported to use the new capabilities; older 32-bit software may be supported through either a *hardware compatibility mode* in which the new processors support the older 32-bit version of the instruction set as well as the 64-bit version, through software emulation, or by the actual implementation of a 32-bit processor core within the 64-bit processor, as with the Itanium processors from Intel, which include an IA-32 processor core to run 32-bit x86 applications. The operating systems for those 64-bit architectures generally support both 32-bit and 64-bit applications.[22]

One significant exception to this is the AS/400, whose software runs on a virtual Instruction Set Architecture (ISA) called TIMI (Technology Independent Machine Interface), which is translated to native machine code by low-level software before being executed. The translation software is all that has to be rewritten to move the entire OS and all software to a new platform, such as when IBM transitioned their line from the older 32/48-bit "IMPI" instruction set to the 64-bit PowerPC-AS instruction set, codenamed "Amazon" (the IMPI instruction set was quite different from the 32-bit PowerPC instruction set, so this was an even bigger transition than from a 32-bit version of an instruction set to a 64-bit version of the same instruction set).

On 64-bit hardware with x86-64 architecture (AMD64), most 32-bit operating systems and applications can run without compatibility issues. While the larger address space of 64-bit architectures makes working with large data sets in applications such as digital video, scientific computing, and large databases easier, there has been considerable debate on whether they or their 32-bit compatibility modes will be faster than comparably priced 32-bit systems for other tasks.

A compiled Java program can run on a 32- or 64-bit Java virtual machine without modification. The lengths and precision of all the built-in types are specified by the standard and are not dependent on the underlying architecture. Java programs that run on a 64-bit Java virtual machine have access to a larger address space.[23]

Speed is not the only factor to consider in a comparison of 32-bit and 64-bit processors. Applications such as multi-tasking, stress testing, and clustering—for high-performance computing (HPC)—may be more suited to a 64-bit architecture when deployed appropriately. 64-bit clusters have been widely deployed in large organizations, such as IBM, HP, and Microsoft, for this reason.

Summary:

- A 64-bit processor performs best with 64-bit software.
- A 64-bit processor has backward compatibility and will handle most 32-bit software.
- A 32-bit processor is not compatible with 64-bit software.

Pros and cons

A common misconception is that 64-bit architectures are no better than 32-bit architectures unless the computer has more than 4 GB of random access memory.[24] This is not entirely true:

- Some operating systems and certain hardware configurations limit the physical memory space to 3 GB on IA-32 systems, due to much of the 3–4 GB region being reserved for hardware addressing; see 3 GB barrier; 64-bit architectures can address far more than 4 GB. However, IA-32 processors from the Pentium II onwards allow for a 36-bit *physical* memory address space, using Physical Address Extension (PAE), which gives a 64 GB physical address range, of which up to 62 GB may be used by main memory; operating systems that support PAE may not be limited to 4GB of physical memory, even on IA-32 processors. However, drivers and other kernel mode software, particularly older versions, may not be compatible with PAE.

- Some operating systems reserve portions of process address space for OS use, effectively reducing the total address space available for mapping memory for user programs. For instance, 32-bit Windows reserves 1 or 2 GB (depending on the settings) of the total address space for the kernel, which leaves only 3 or 2 GB (respectively) of the address space available for user mode. This limit is much higher on 64-bit operating systems.

- Memory-mapped files are becoming more difficult to implement in 32-bit architectures as files of over 4 GB become more common; such large files cannot be memory-mapped easily to 32-bit architectures—

only part of the file can be mapped into the address space at a time, and to access such a file by memory mapping, the parts mapped must be swapped into and out of the address space as needed. This is a problem, as memory mapping, if properly implemented by the OS, is one of the most efficient disk-to-memory methods.

- Some 64-bit programs, such as encoders, decoders and encryption software, can benefit greatly from 64-bit registers, while the performance of other programs, such as 3D graphics-oriented ones, remains unaffected when switching from a 32-bit to a 64-bit environment.

- Some 64-bit architectures, such as x86-64, support more general-purpose registers than their 32-bit counterparts (although this is not due specifically to the word length). This leads to a significant speed increase for tight loops since the processor does not have to fetch data from the cache or main memory if the data can fit in the available registers.

  Example in C:

```c
int a, b, c, d, e;
for (a=0; a<100; a++)
{
  b = a;
  c = b;
  d = c;
  e = d;
}
```

  If a processor only has the ability to keep two or three values or variables in registers it would need to move some values between memory and registers to be able to process variables d and e as well; this is a process that takes many CPU cycles. A processor that is capable of holding all values and variables in registers can loop through them without needing to move data between registers and memory for each iteration. This behavior can easily be compared with virtual memory, although any effects are contingent upon the compiler.

The main disadvantage of 64-bit architectures is that, relative to 32-bit architectures, the same data occupies more space in memory (due to longer pointers and possibly other types, and alignment padding). This increases the memory requirements of a given process and can have implications for efficient processor cache utilization. Maintaining a partial 32-bit model is one way to handle this, and is in general reasonably effective. For example, the z/OS operating system takes this approach, requiring program code to reside in 31-bit address spaces (the high order bit is not used in address calculation on the underlying hardware platform) while data objects can optionally reside in 64-bit regions.

As of June 2011, most proprietary x86 software is compiled into 32-bit code, with less being also compiled into 64-bit code (although the trend is rapidly equalizing[*citation needed*]), so most of that software does not take advantage of the larger 64-bit address space or wider 64-bit registers and data paths on x64 processors, or the additional general-purpose registers. However, users of most RISC platforms, and users of free or open source operating systems (where the source code is available for recompiling with a 64-bit compiler) have been able to use exclusive 64-bit computing environments for years. Not all such applications require a large address space or manipulate 64-bit data items, so these applications do not benefit from these features. The main advantage of 64-bit versions of such applications is the ability to access more registers in the x86-64 architecture.

Software availability

x86-based 64-bit systems sometimes lack equivalents of software that is written for 32-bit architectures. The most severe problem in Microsoft Windows is incompatible device drivers for obsolete hardware. Most 32-bit application software can run on a 64-bit operating system in a compatibility mode, also known as an emulation mode, e.g. Microsoft WoW64 Technology for IA-64 and AMD64. The 64-bit Windows Native Mode[25] driver environment runs atop 64-bit NTDLL.DLL, which cannot call 32-bit Win32 subsystem code (often devices whose actual hardware function is emulated in user mode software, like Winprinters). Because 64-bit drivers for most devices were not available until early 2007 (Vista x64), using a 64-bit version of Windows was considered a challenge. However, the trend has since moved towards 64-bit computing, particularly as memory prices dropped and the use of more than 4 GB of RAM increased. Most manufacturers started to provide both 32-bit and 64-bit drivers for new devices, so unavailability of 64-bit drivers ceased to be a problem. 64-bit drivers were not provided for many older devices, which could consequently not be used in 64-bit systems.

Driver compatibility was less of a problem with open-source drivers, as 32-bit ones could be modified for 64-bit use. Support for hardware made before early 2007 was problematic for open-source platforms[*citation needed*], due to the relatively small number of users.

Mac OS X Tiger and Mac OS X Leopard only had a 32-bit kernel, but that kernel can run 64-bit user-mode code on 64-bit-capable processors. Mac OS X Snow Leopard had both 32-bit and 64-bit kernels, and, on most Macs, used the 32-bit kernel even on 64-bit processors; this allowed those Macs to support 64-bit processes while still supporting 32-bit device drivers – although not 64-bit drivers and performance advantages that would come with them. Mac OS X Lion ran with a 64-bit kernel on more Macs, and OS X Mountain Lion only has a 64-bit kernel. On systems with 64-bit processors, both the 32- and 64-bit OS X kernels can run 32-bit user-mode code, and all versions of OS X include 32-bit versions of libraries that 32-bit applications would use, so 32-bit user-mode software for OS X will run on those systems.

Linux and most other Unix-like operating systems, and the C and C++ toolchains for them, have supported 64-bit processors for many years, releasing 64-bit versions of their operating systems before official Microsoft releases. Many applications and libraries for those platforms are open source, written in C and C++, so that if they are 64-bit-safe they can be compiled into 64-bit versions. This source-based distribution model with an emphasis on frequent releases makes availability of application software for those operating systems less of an issue.

64-bit data models

In 32-bit programs, pointers and data types such as integers generally have the same length; this is not necessarily true on 64-bit machines.[26][27][28] Mixing data types in programming languages such as C and its descendants such as C++ and Objective-C may thus function on 32-bit implementations but not on 64-bit implementations.

In many programming environments for C and C-derived languages on 64-bit machines, "int" variables are still 32 bits wide, but long integers and pointers are 64 bits wide. These are described as having an **LP64** data model. Another alternative is the **ILP64** data model in which all three data types are 64 bits wide, and even **SILP64** where "short" integers are also 64 bits wide.[*citation needed*] However, in most cases the modifications required are relatively minor and straightforward, and many well-written programs can simply be recompiled for the new environment without changes. Another alternative is the **LLP64** model, which maintains compatibility with 32-bit code by leaving both int and long as 32-bit. "LL" refers to the "long long integer" type, which is at least 64 bits on all platforms, including 32-bit environments.

64-bit data models

Data model	short (integer)	int	long (integer)	long long	pointers/ size_t	Sample operating systems
LLP64/ IL32P64	16	32	32	64	64	Microsoft Windows (X64/IA-64)
LP64/ I32LP64	16	32	64	64	64	Most Unix and Unix-like systems, e.g. Solaris, Linux, BSD, and OS X; z/OS
ILP64	16	64	64	64	64	HAL Computer Systems port of Solaris to SPARC64
SILP64	64	64	64	64	64	UNICOS

Many 64-bit compilers today use the **LP64** model (including Solaris, AIX, HP-UX, Linux, OS X, BSD, and IBM z/OS native compilers). Microsoft's Visual C++ compiler uses the **LLP64** model. The disadvantage of the LP64 model is that storing a long into an int may overflow. On the other hand, casting a pointer to a long will work. In the LLP model, the reverse is true. These are not problems which affect fully standard-compliant code, but code is often written with implicit assumptions about the widths of integer types.

Note that a programming model is a choice made on a per-compiler basis, and several can coexist on the same OS. However, the programming model chosen as the primary model for the OS API typically dominates.

Another consideration is the data model used for drivers. Drivers make up the majority of the operating system code in most modern operating systems[*citation needed*] (although many may not be loaded when the operating system is running). Many drivers use pointers heavily to manipulate data, and in some cases have to load pointers of a certain size into the hardware they support for DMA. As an example, a driver for a 32-bit PCI device asking the device to DMA data into upper areas of a 64-bit machine's memory could not satisfy requests from the operating system to load data from the device to memory above the 4 gigabyte barrier, because the pointers for those addresses would not fit into the DMA registers of the device. This problem is solved by having the OS take the memory restrictions of the device into account when generating requests to drivers for DMA, or by using an IOMMU.

Current 64-bit microprocessor architectures

64-bit microprocessor architectures for which processors are currently being manufactured (as of January 2011) include:

- The 64-bit extension created by AMD to Intel's x86 architecture (later licensed by Intel); commonly known as "x86-64", "AMD64", or "x64":
  - AMD's AMD64 extensions (used in Athlon 64, Opteron, Sempron, Turion 64, Phenom, Athlon II and Phenom II processors)
  - Intel's Intel 64 extensions (used in newer Celeron, Pentium, and Xeon processors, in Intel Core 2/i3/i5/i7 processors, and in some Atom processors)
  - VIA Technologies' 64-bit extensions, used in the VIA Nano processors
- The 64-bit version of the Power Architecture:
  - IBM's POWER6 and POWER7 processors
  - IBM's PowerPC 970 processor
  - The Cell Broadband Engine used in the PlayStation 3, designed by IBM, Toshiba and Sony, combines a 64-bit Power architecture processor with seven or eight Synergistic Processing Elements.
  - IBM's "Xenon" processor used in the Microsoft Xbox 360 comprises three 64-bit PowerPC cores.
- SPARC V9 architecture:
  - Sun's UltraSPARC processors
  - Fujitsu's SPARC64 processors
- IBM's z/Architecture, a 64-bit version of the ESA/390 architecture, used in IBM's eServer zSeries and System z mainframes
- Intel's IA-64 architecture (used in Itanium processors)
- MIPS Technologies' MIPS64 architecture
- ARM Holdings' AArch64 architecture

Most 64-bit processor architectures that are derived from 32-bit processor architectures can execute code for the 32-bit version of the architecture natively without any performance penalty.[*citation needed*] This kind of support is commonly called *bi-arch support* or more generally *multi-arch support*.

See also

- Computer memory

Notes

1. ^ ***a*** ***b*** "ARM Launches Cortex-A50 Series, the World's Most Energy-Efficient 64-bit Processors" (http://www.arm.com/about/newsroom/arm-launches-cortex-a50-series-the-worlds-most-energy-efficient-64-bit-processors.php) (Press release). ARM Holdings. Retrieved 2012-10-31.
2. **^** Grimes, Jack; Kohn, Les; Bharadhwaj, Rajeev (July/August 1989). "The Intel i860 64-Bit Processor: A General-Purpose CPU with 3D Graphics Capabilities" (http://www.computer.org/portal/web/csdl/doi/10.1109/38.31467). *IEEE Computer Graphics and Applications* **9** (4): 85–94. doi:10.1109/38.31467 (http://dx.doi.org/10.1109%2F38.31467). Retrieved 2010-11-19.
3. **^** Zachary, *Showstopper!*
4. **^** "NEC Offers Two High Cost Performance 64-bit RISC Microprocessors" (http://www.nec.co.jp/press/en/9801/2002.htm) (Press release). NEC. 1998-01-20. Retrieved 2011-01-09. "Versions of the VR4300 processor are widely used in consumer and office automation applications, including the popular Nintendo 64TM video game and advanced laser printers such as the recently announced, award-winning Hewlett-Packard LaserJet 4000 printer family."
5. **^** AMD64 Programmer's Manual Volume 2: System Programming, order number 24593, revision 3.14, September 2007, Advanced Micro Devices
6. **^** "i860 64-Bit Microprocessor" (http://smithsonianchips.si.edu/intel/i860.htm). Intel. 1989. Retrieved 30 November 2010.
7. **^** Joe Heinrich: "MIPS R4000 Microprocessor User's Manual, Second Edition", 1994, MIPS Technologies, Inc.
8. **^** Richard L. Sites: "Alpha AXP Architecture", Digital Technical Journal, Volume 4, Number 4, 1992, Digital Equipment Corporation.
9. **^** *Atari Jaguar History (http://www.atariage.com/Jaguar/history.html)*. AtariAge. Retrieved 9 August 2010.
10. **^** Linley Gwennap: "UltraSparc Unleashes SPARC Performance", Microprocessor Report, Volume 8, Number 13, 3 October 1994, MicroDesign Resources.
11. **^** J. W. Bishop, et al.: "PowerPC AS A10 64-bit RISC microprocessor", IBM Journal of Research and Development, Volume 40, Number 4, July 1996, IBM Corporation.
12. **^** Linley Gwennap: "PA-8000 Combines Complexity and Speed", Microprocessor Report, Volume 8, Number 15, 14 November 1994, MicroDesign Resources
13. **^** F. P. O'Connell and S. W. White: "POWER3: The next generation of PowerPC processors", IBM Journal of Research and Development, Volume 44, Number 6, November 2000, IBM Corporation.
14. **^** "VIA Unveils Details of Next-Generation Isaiah Processor Core" (http://www.via.com.tw/en/resources/pressroom/2004_archive/pr041005_fpf-isaiah.jsp). VIA Technologies, Inc. Retrieved 2007-07-18.
15. **^** "ARMv8 Technology Preview" (http://www.arm.com/files/downloads/ARMv8_Architecture.pdf). October 31, 2011. Retrieved November 15, 2012.
16. **^** ARM Keynote: ARM Cortex-A53 and ARM Cortex-A57 64bit ARMv8 processors launched (http://armdevices.net/2012/10/31/arm-keynote-arm-cortex-a53-and-arm-cortex-a57-64bit-armv8-processors-launched/) on armdevices.net
17. **^** Stefan Berka. "Unicos Operating System" (http://www.operating-system.org/betriebssystem/_english/bs-unicos.htm). www.operating-system.org. Archived (http://web.archive.org/web/20101126033526/http://operating-system.org/betriebssystem/_english/bs-unicos.htm) from the original on 26 November 2010. Retrieved 2010-11-19.
18. **^** My Life and Free Software (http://www.linuxjournal.com/article/4047)
19. **^** Andi Kleen (2001-06-26). "Porting Linux to x86-64" (http://www.x86-64.org/pipermail/announce/2001-June/000020.html). "Status: The kernel, compiler, tool chain work. The kernel boots and work on simulator and is used for porting of userland and running programs"
20. ^ ***a*** ***b*** John Siracusa. "Mac OS X 10.6 Snow Leopard: the Ars Technica review" (http://arstechnica.com/apple/reviews/2009/08/mac-os-x-10-6.ars/5). Ars Technica. p. 5. Archived (http://web.archive.org/web/20091009161632/http://arstechnica.com/apple/reviews/2009/08/mac-os-x-10-6.ars/5) from the original on 9 October 2009. Retrieved 2009-09-06.
21. **^** Mashey, John (October 2006). "The Long Road to 64 Bits" (http://queue.acm.org/detail.cfm?id=1165766). *ACM Queue* **4** (8): 85–94. Retrieved 2011-02-19.
22. **^** "Windows 7: 64 bit vs 32 bit?" (http://www.w7forums.com/windows-7-64-bit-vs-32-bit-t484.html). W7 Forums. Archived (http://web.archive.org/web/20090405053428/http://www.w7forums.com/windows-7-64-bit-vs-32-bit-t484.html) from the original on 5 April 2009. Retrieved 2009-04-05.
23. **^** "Frequently Asked Questions About the Java HotSpot VM" (http://java.sun.com/docs/hotspot/HotSpotFAQ.html#64bit_compilers). Sun Microsystems, Inc. Archived (http://web.archive.org/web/20070510204014/http://java.sun.com/docs/hotspot/HotSpotFAQ.html) from the original

on 10 May 2007. Retrieved 2007-05-03.
24. ^ "A description of the differences between 32-bit versions of Windows Vista and 64-bit versions of Windows Vista" (http://support.microsoft.com/kb/946765). Retrieved 2011-10-14.
25. ^ "Inside Native Applications" (http://technet.microsoft.com/en-us/sysinternals/bb897447.aspx). Technet.microsoft.com. 2006-11-01. Archived (http://web.archive.org/web/20101023130328/http://technet.microsoft.com/en-us/sysinternals/bb897447.aspx) from the original on 23 October 2010. Retrieved 2010-11-19.
26. ^ "Exploring 64-bit development on POWER5: How portable is your code, really?" (http://www.ibm.com/developerworks/power/library/pa-openpower1/#N100C7) by Peter Seebach 2006
27. ^ "The Ten Commandments for C Programmers" (http://www.lysator.liu.se/c/ten-commandments.html) by Henry Spencer
28. ^ "The Story of Thud and Blunder" (http://www.datacenterworks.com/stories/thud.html). Datacenterworks.com. Retrieved 2010-11-19.

References

- This article is based on material taken from the Free On-line Dictionary of Computing prior to 1 November 2008 and incorporated under the "relicensing" terms of the GFDL, version 1.3 or later.

External links

- 64-bit Transition Guide, Mac Developer Library (http://developer.apple.com/library/mac/documentation/Darwin/Conceptual/64bitPorting/transition/transition.html)
- Andrey Karpov. A Collection of Examples of 64-bit Errors in Real Programs (http://software.intel.com/en-us/articles/collection-of-examples-of-64-bit-errors-in-real-programs/)
- Mark J. Kilgard. Is your X code ready for 64-bit? (http://web.archive.org/web/20010603003655/http://reality.sgi.com/mjk/64bit/64bit.html)
- Lessons on development of 64-bit C/C++ applications (http://software.intel.com/en-us/articles/lessons-on-development-of-64-bit-cc-applications/)
- 64-Bit Programming Models: Why LP64? (http://www.unix.org/version2/whatsnew/lp64_wp.html)
- AMD64 (EM64T) architecture (http://www.codeproject.com/KB/system/AMD64_EM64T_architectur.aspx)

Retrieved from "http://en.wikipedia.org/w/index.php?title=64-bit_computing&oldid=574195923"
Categories: Data unit
