64-bit computing

In computer architecture, **64-bit** integers, memory addresses, or other data units are those that are 64 bits (8 octets) wide. Also, 64-bit CPU and ALU architectures are those that are based on registers, address buses, or data buses of that size. 64-bit microcomputers are computers in which 64-bit microprocessors are the norm. From the software perspective, 64-bit computing means the use of code with 64-bit virtual memory addresses. However, not all 64-bit instruction sets support full 64-bit virtual memory addresses; x86-64 and ARMv8, for example, support only 48 bits of virtual address, with the remaining 16 bits of the virtual address required to be all 0's or all 1's, and several 64-bit instruction sets support fewer than 64 bits of physical memory address.

The term *64-bit* describes 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 been used in supercomputers since the 1970s (Cray-1, 1975) and in reduced instruction set computing (RISC) based workstations and servers since the early 1990s, notably the MIPS R4000, R8000, and R10000, the DEC Alpha, the Sun UltraSPARC, and the IBM RS64 and POWER3 and later POWER microprocessors. In 2003, 64-bit CPUs were introduced to the (formerly 32-bit) mainstream personal computer market in the form of x86-64 processors and the PowerPC G5, and were introduced in 2012[1] into the ARM architecture targeting smartphones and tablet computers, first sold on September 20, 2013, in the iPhone 5S powered by the ARMv8-A Apple A7 system on a chip (SoC).

A 64-bit register can hold any of 2^{64} (over 18 quintillion or 1.8×10^{19}) different values. The range of integer values that can be stored in 64 bits depends on the integer representation used. With the two most common representations, the range is 0 through 18,446,744,073,709,551,615 ($2^{64} - 1$) for representation as an (unsigned) binary number, and −9,223,372,036,854,775,808 (-2^{63}) through 9,223,372,036,854,775,807 ($2^{63} - 1$) for representation as two's complement. Hence, a processor with 64-bit memory addresses can directly access 2^{64} bytes (=16 exabytes) of byte-addressable memory.

With no further qualification, a *64-bit computer architecture* generally has integer and addressing processor 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).[2] The term may also refer to the size of low-level data types, such as 64-bit floating-point numbers.

Contents

Architectural implications

History

Limits of processors

64-bit data timeline

64-bit address timeline

64-bit operating system timeline

64-bit applications

32-bit vs 64-bit

Pros and cons
Software availability
64-bit data models
Current 64-bit architectures
See also
Notes
References
External links

Architectural implications

Processor registers are typically divided into several groups: *integer*, *floating-point*, *single-instruction-multiple-data* (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 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 architecture (ARM) and 32-bit MIPS architecture (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 able to load and store 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, and 64-bit integer registers.

History

Many computer instruction sets are designed so that a single integer register can store the memory address to any location in the computer's physical or virtual memory. Therefore, the total number of addresses to memory is often determined by the width of these registers. The IBM System/360 of the 1960s was an early 32-bit computer; it had 32-bit integer registers, although it only used the low order 24 bits of a word for addresses, resulting in a 16 MiB [16×1024^2 bytes] address space. 32-bit superminicomputers, such as the DEC VAX, became common in the 1970s, and 32-bit microprocessors, such as the Motorola 68000 family and the 32-bit members of the x86 family starting with the Intel 80386, appeared in the mid-1980s, making 32 bits something of a *de facto* consensus as a convenient register size.

A 32-bit address register meant that 2^{32} addresses, or 4 GiB of random-access memory (RAM), could be referenced. When these architectures were devised, 4 GB of memory was so far beyond the typical amounts (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, such as the Cray-1,[3] used registers up to 64 bits wide, and supported 64-bit integer arithmetic, although they did not support 64-bit addressing. In the mid-1980s, Intel i860[4] development began culminating in a (too late[5] for Windows NT) 1989 release; the i860 had 32-bit integer registers and 32-bit addressing, so it was not a fully 64-bit processor, although its graphics

unit supported 64-bit integer arithmetic.[6] However, 32 bits remained the norm until the early 1990s, when the continual reductions in the cost of memory led to installations with amounts 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[7] and the PlayStation 2 had 64-bit microprocessors before their introduction in personal computers. High-end printers, network equipment, and industrial computers, also used 64-bit microprocessors, such as the Quantum Effect Devices R5000. 64-bit computing started to trickle down to the personal computer desktop from 2003 onward, 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.

Limits of processors

In principle, a 64-bit microprocessor can address 16 EiBs ($16 \times 1024^6 = 2^{64}$ = 18,446,744,073,709,551,616 bytes, or about 18.4 exabytes) of memory. However, not all instruction sets, and not all processors implementing those instruction sets, support a full 64-bit virtual or physical address space.

The x86-64 architecture (as of 2016) allows 48 bits for virtual memory and, for any given processor, up to 52 bits for physical memory.[8][9] These limits allow memory sizes of 256 TiB (256×1024^4 bytes) and 4 PiB (4×1024^5 bytes), respectively. A PC cannot currently contain 4 pebibytes 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. Thus the 52-bit physical address provides ample room for expansion while not incurring the cost of implementing full 64-bit physical addresses. Similarly, the 48-bit virtual address space was designed to provide more than 65,000 (2^{16}) times the 32-bit limit of 4 GiB (4×1024^3 bytes), allowing room for later expansion and incurring no overhead of translating full 64-bit addresses.

The Power ISA v3.0 allows 64 bits for an effective address, mapped to a segmented address with between 65 and 78 bits allowed, for virtual memory, and, for any given processor, up to 60 bits for physical memory.[10]

The Oracle SPARC Architecture 2015 allows 64 bits for virtual memory and, for any given processor, between 40 and 56 bits for physical memory.[11]

The ARM AArch64 Virtual Memory System Architecture allows 48 bits for virtual memory and, for any given processor, from 32 to 48 bits for physical memory.[12]

64-bit data 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 (prior 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 reduced instruction set computing (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.[13]

1993

Atari introduces the Atari Jaguar video game console, which includes some 64-bit wide data paths in its architecture.[14]

64-bit address timeline

1991

MIPS Computer Systems produces the first 64-bit microprocessor, the R4000, which implements the MIPS III architecture, the third revision of its MIPS architecture.[15] 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.[16]

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 was targeted.

1995

Sun launches a 64-bit SPARC processor, the UltraSPARC.[17] 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, the first 64-bit PowerPC AS processors.[18] 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 the first implementation of its 64-bit PA-RISC 2.0 architecture, the PA-8000.[19]

1998

IBM releases the POWER3 line of full-64-bit PowerPC/POWER processors.[20]

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.[21]

2006

Sony, IBM, and Toshiba begin manufacturing the 64-bit Cell processor for use in the PlayStation 3, servers, workstations, and other appliances. Intel released Core 2 Duo as the first mainstream x86-64 processor for its mobile, desktop, and workstation line. Prior 64-bit extension processor lines were not widely available in the consumer retail market (most of 64-bit Pentium 4/D were OEM), 64-bit Pentium 4, Pentium D, and Celeron were not into mass production until late 2006 due to poor yield issue (most of good yield wafers were targeted at server and mainframe while mainstream still remain 130 nm 32-bit processor line until 2006) and soon became low end after Core 2 debuted. AMD released their first 64-bit mobile processor and manufactured in 90 nm.

2011

ARM Holdings announces ARMv8-A, the first 64-bit version of the ARM architecture.[22]

2012

ARM Holdings announced their Cortex-A53 and Cortex-A57 cores, their first cores based on their 64-bit architecture, on 30 October 2012.[1][23]

2013

Apple announces the iPhone 5S, the first 64-bit smartphone, which uses their A7 ARMv8-A-based system-on-a-chip.

2014

Google announces the Nexus 9, the first Android device to run on a 64-bit Tegra K1 processor.

64-bit operating system timeline

1985

Cray releases UNICOS, the first 64-bit implementation of the Unix operating system.[24]

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 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.[25]

1996

Support for the 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 z Systems 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).[26]

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.[27] Several Linux distributions release with support for AMD64. FreeBSD releases with support for AMD64.

2005

On January 4, Microsoft discontinues Windows XP 64-Bit Edition, as no PCs with IA-64 processors had been available since the previous September, and announces that it is developing x86-64 versions of Windows to replace it.[28] 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 and Windows Server 2003 x64 Edition for AMD64 and EM64T processors.[29]

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 plug-ins.

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. Microsoft also releases Windows Server 2008 R2, which is the first 64-bit only server operating system. 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.[27]

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.

2014

Google releases Android Lollipop, the first version of the Android operating system with support for 64-bit processors.

2017

Apple releases iOS 11, supporting only machines with AArch64 processors. It has a 64-bit kernel that only supports 64-bit applications. 32-bit applications are no longer compatible.

2019

Apple releases macOS 10.15 "Catalina", dropping support for Intel 32-bit applications.

64-bit applications

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.[30] Other software must also be ported to use the new abilities; older 32-bit software may be supported either by virtue of the 64-bit instruction set being a superset of the 32-bit instruction set, so that processors that support the 64-bit instruction set can also run code for the 32-bit instruction set, or through software emulation, or by the actual implementation of a 32-bit processor core within the 64-bit processor, as with some Itanium processors from Intel, which included 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.[31]

One significant exception to this is the AS/400, software for which is compiled into a virtual instruction set architecture (ISA) called *Technology Independent Machine Interface* (TIMI); TIMI code is then translated to native machine code by low-level software before being executed. The translation software is all that must be rewritten to move the full OS and all software to a new platform, as when IBM transitioned the native instruction set for AS/400 from the older 32/48-bit *IMPI* to the newer 64-bit *PowerPC-AS*, codenamed *Amazon*. The IMPI instruction set was quite different from even 32-bit PowerPC, so this transition was even bigger than moving a given instruction set from 32 to 64 bits.

On 64-bit hardware with x86-64 architecture (AMD64), most 32-bit operating systems and applications can run with no 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 with no modification. The lengths and precision of all the built-in types, such as `char`, `short`, `int`, `long`, `float`, and `double`, and the types that can be used as array indices, 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.[32]

Speed is not the only factor to consider in comparing 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. For this reason, 64-bit clusters have been widely deployed in large organizations, such as IBM, HP, and Microsoft.

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 incompatible 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 GiB of random-access memory.[33] This is not entirely true:

- Some operating systems and certain hardware configurations limit the physical memory space to 3 GiB on IA-32 systems, due to much of the 3–4 GiB region being reserved for hardware addressing; see 3 GB barrier; 64-bit architectures can address far more than 4 GiB. However, IA-32 processors from the Pentium Pro onward allow a 36-bit *physical* memory address space, using Physical Address Extension (PAE), which gives a 64 GiB physical address range, of which up to 62 GiB may be used by main memory; operating systems that support PAE may not be limited to 4 GiB of physical memory, even on IA-32 processors. However, drivers and other kernel mode software, more so older versions, may be incompatible with PAE; this has been cited as the reason for 32-bit versions of Microsoft Windows being limited to 4 GiB of physical RAM[34] (although the validity of this explanation has been disputed[35]).
- 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 GiB (depending on the settings) of the total address space for the kernel, which leaves only 3 or 2 GiB (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 GiB become more common; such large files cannot be memory-mapped easily to 32-bit architectures, as 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:

```
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 also; this is a process that takes many CPU cycles. A processor that is able to hold all values and variables in registers can loop through them with no need to move data between registers and memory for each iteration. This behavior can easily be compared with virtual memory, although any effects are contingent on 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 use. 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. Not all such applications require a large address space or manipulate 64-bit data items, so these applications do not benefit from these features.

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 termed an emulation mode, e.g., Microsoft WoW64 Technology for IA-64 and AMD64. The 64-bit Windows Native Mode[36] 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 unavailable until early 2007 (Vista x64), using a 64-bit version of Windows was considered a challenge. However, the trend has since moved toward 64-bit computing, more so 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, due to the relatively small number of users.

64-bit versions of Windows cannot run 16-bit software. However, most 32-bit applications will work well. 64-bit users are forced to install a virtual machine of a 16- or 32-bit operating system to run 16-bit applications.[37]

Mac OS X 10.4 "Tiger" and Mac OS X 10.5 "Leopard" only had a 32-bit kernel, but they can run 64-bit user-mode code on 64-bit processors. Mac OS X 10.6 "Snow Leopard" had both 32- 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 can come with them. Mac OS X 10.7 "Lion" ran with a 64-bit kernel on more Macs, and OS X 10.8 "Mountain Lion" and later macOS releases only have a 64-bit kernel. On systems with 64-bit processors, both the 32- and 64-bit macOS kernels can run 32-bit user-mode code, and all versions of macOS include 32-bit versions of libraries that 32-bit applications would use, so 32-bit user-mode software for macOS 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. Many applications and libraries for those platforms are open-source software, 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.[38][39][40] Mixing data types in programming languages such as C and its descendants such as C++ and Objective-C may thus work 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.[41][42] Other models are the *ILP64* data model in which all three data types are 64 bits wide,[43][42] and even the *SILP64* model where *short* integers are also 64 bits wide.[44][45] 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 with no changes. Another alternative is the *LLP64* model, which maintains compatibility with 32-bit code by leaving both `int` and `long` as 32-bit.[46][42] *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	16	32	32	64	64	Microsoft Windows (x86-64 and IA-64) using Visual C++; and MinGW
LP64	16	32	64	64	64	Most Unix and Unix-like systems, e.g., Solaris, Linux, BSD, macOS. Windows when using Cygwin; z/OS
ILP64	16	64	64	64	64	HAL Computer Systems port of Solaris to the SPARC64
SILP64	64	64	64	64	64	*Classic* UNICOS[44][45] (versus UNICOS/mp, etc.)

Many 64-bit platforms today use an *LP64* model (including Solaris, AIX, HP-UX, Linux, macOS, BSD, and IBM z/OS). Microsoft Windows uses an *LLP64* model. The disadvantage of the LP64 model is that storing a `long` into an `int` may overflow. On the other hand, converting a pointer to a `long` will "work" in LP64. In the LLP64 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 data types. C code should prefer `(u)intptr_t` instead of `long` when casting pointers into integer objects.

A programming model is a choice made to suit a given compiler, and several can coexist on the same OS. However, the programming model chosen as the primary model for the OS application programming interface (API) typically dominates.

Another consideration is the data model used for device drivers. Drivers make up the majority of the operating system code in most modern operating systems (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 direct memory access (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 gibibyte 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 input–output memory management unit (IOMMU).

Current 64-bit architectures

As of May 2018, 64-bit architectures for which processors are being manufactured include:

- The 64-bit extension created by AMD to Intel's x86 architecture (later licensed by Intel); commonly termed *x86-64*, *AMD64*, or *x64*:
  - AMD's AMD64 extensions (used in Athlon 64, Opteron, Sempron, Turion 64, Phenom, Athlon II, Phenom II, FX, Ryzen, and Epyc processors)

    - Intel's Intel 64 extensions, used in Intel Core 2-i3-i5-i7-i9, some Atom, and newer Celeron, Pentium, and Xeon processors
        - Intel's K1OM architecture, a variant of *Intel 64* with no CMOV, MMX, and SSE instructions, used in Xeon Phi coprocessors, binary incompatible with x86-64 programs
    - VIA Technologies' 64-bit extensions, used in the VIA Nano processors
- IBM's PowerPC/Power ISA:
    - IBM's POWER4, POWER5, POWER6, POWER7, POWER8, POWER9, and IBM A2 processors
- SPARC V9 architecture:
    - Oracle's M8 and S7 processors
    - Fujitsu's SPARC64 XII and SPARC64 XIfx processors
- IBM's z/Architecture, a 64-bit version of the ESA/390 architecture, used in IBM's eServer zSeries and System z mainframes:
    - IBM z13 and z14
    - Hitachi AP8000E
- HP-Intel's IA-64 architecture:
    - Intel's Itanium processors
- MIPS Technologies' MIPS64 architecture
- ARM Holdings' AArch64 architecture
- Elbrus architecture:
    - Elbrus-8S
- NEC SX architecture
    - SX-Aurora TSUBASA
- RISC-V

Most architectures of 64 bits that are derived from the same architecture of 32 bits can execute code written for the 32-bit versions natively, with no performance penalty. This kind of support is commonly called *bi-arch support* or more generally *multi-arch support*.

See also

- Computer memory

Notes

- 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.

References

1. "ARM Launches Cortex-A50 Series, the World's Most Energy-Efficient 64-bit Processors" (https://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. *Pentium Processor User's Manual Volume 1: Pentium Processor Data Book* (http://bitsavers.org/components/intel/pentium/1993_Intel_Pentium_Processor_Users_Manual_Volume_1.pdf) (PDF). Intel. 1993.
3. "Cray-1 Computer System Hardware Reference Manual" (http://bitsavers.trailing-edge.com/pdf/cray/CRAY-1/2240004C_CRAY-1_Hardware_Reference_Nov77.pdf) (PDF). Cray Research. 1977. Retrieved October 8, 2013.
4. Grimes, Jack; Kohn, Les; Bharadhwaj, Rajeev (July–August 1989). "The Intel i860 64-Bit Processor: A General-Purpose CPU with 3D Graphics Capabilities" (https://www.computer.org/csdl/mags/cg/1989/04/mcg1989040085-abs.html). *IEEE Computer Graphics and Applications*. **9** (4): 85–94. doi:10.1109/38.31467 (https://doi.org/10.1109%2F38.31467). Retrieved 2010-11-19.
5. Zachary, G. Pascal (1994). *Showstopper! The Breakneck Race to Create Windows NT and the Next Generation at Microsoft* (https://archive.org/details/showstopperbreak00zach). Warner Books. ISBN 0-02-935671-7.
6. "i860 Processor Family Programmer's Reference Manual" (http://bitsavers.org/components/intel/i860/240875-001_i860_64-Bit_Microprocessor_Programmers_Reference_May91.pdf) (PDF). Intel. 1991. Retrieved September 12, 2019.
7. "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."
8. "AMD64 Programmer's Manual Volume 2: System Programming" (https://support.amd.com/TechDocs/24593.pdf) (PDF). Advanced Micro Devices. December 2016. p. 120.
9. "Intel 64 and IA-32 Architectures Software Developer's Manual Volume 3A: System Programming Guide, Part 1" (https://www.intel.com/content/dam/www/public/us/en/documents/manuals/64-ia-32-architectures-software-developer-vol-3a-part-1-manual.pdf) (PDF). Intel. September 2016. p. 4-2.
10. "Power ISA Version 3.0" (https://openpowerfoundation.org/?resource_lib=power-isa-version-3-0). IBM. November 30, 2015. p. 983.
11. "Oracle SPARC Architecture 2015 Draft D1.0.9" (https://community.oracle.com/docs/DOC-1005258). Oracle. p. 475.
12. "ARM Architecture Reference Manual ARMv8, for ARMv8-A architecture profile" (http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0487a.k_10775/index.html). pp. D4-1723, D4-1724, D4-1731.
13. "i860 64-Bit Microprocessor" (http://smithsonianchips.si.edu/intel/i860.htm). Intel. 1989. Retrieved 30 November 2010.
14. "Atari Jaguar History" (http://www.atariage.com/Jaguar/history.html). *AtariAge*.
15. Joe Heinrich (1994). *MIPS R4000 Microprocessor User's Manual* (2nd ed.). MIPS Technologies, Inc.
16. Richard L. Sites (1992). "Alpha AXP Architecture". *Digital Technical Journal*. Digital Equipment Corporation. **4** (4).
17. Linley Gwennap (3 October 1994). "UltraSparc Unleashes SPARC Performance". *Microprocessor Report*. MicroDesign Resources. **8** (13).
18. J. W. Bishop; et al. (July 1996). "PowerPC AS A10 64-bit RISC microprocessor". *IBM Journal of Research and Development*. IBM Corporation. **40** (4): 495–505. doi:10.1147/rd.404.0495 (https://doi.org/10.1147%2Frd.404.0495).
19. Linley Gwennap (14 November 1994). "PA-8000 Combines Complexity and Speed". *Microprocessor Report*. MicroDesign Resources. **8** (15).

20. F. P. O'Connell; S. W. White (November 2000). "POWER3: The next generation of PowerPC processors". *IBM Journal of Research and Development*. IBM Corporation. **44** (6): 873–884. doi:10.1147/rd.446.0873 (https://doi.org/10.1147%2Frd.446.0873).
21. "VIA Unveils Details of Next-Generation Isaiah Processor Core" (https://web.archive.org/web/20071011053054/http://via.com.tw/en/resources/pressroom/2004_archive/pr041005_fpf-isaiah.jsp). VIA Technologies, Inc. Archived from the original (http://www.via.com.tw/en/resources/pressroom/2004_archive/pr041005_fpf-isaiah.jsp) on 2007-10-11. Retrieved 2007-07-18.
22. "ARMv8 Technology Preview" (https://www.arm.com/files/downloads/ARMv8_Architecture.pdf) (PDF). October 31, 2011. Retrieved November 15, 2012.
23. "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/). *ARMdevices.net*. 2012-10-31.
24. Stefan Berka. "Unicos Operating System" (http://www.operating-system.org/betriebssystem/_english/bs-unicos.htm). www.operating-system.org. Archived (https://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.
25. Jon "maddog" Hall (Jun 1, 2000). "My Life and Free Software" (https://www.linuxjournal.com/article/4047). *Linux Journal*.
26. Andi Kleen. *Porting Linux to x86-64* (https://www.kernel.org/doc/ols/2001/x86-64.pdf) (PDF). Ottawa Linux Symposium 2001. "Status: The kernel, compiler, tool chain work. The kernel boots and work on simulator and is used for porting of userland and running programs"
27. John Siracusa. "Mac OS X 10.6 Snow Leopard: the Ars Technica review" (https://arstechnica.com/apple/reviews/2009/08/mac-os-x-10-6.ars/5). *Ars Technica*. p. 5. Archived (https://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.
28. Joris Evers (5 January 2005). "Microsoft nixes Windows XP for Itanium" (https://web.archive.org/web/20130618025711/http://www.computerworld.com/s/article/98716/Microsoft_nixes_Windows_XP_for_Itanium?taxonomyId=125). Computerworld. Archived from the original (http://www.computerworld.com/s/article/98716/Microsoft_nixes_Windows_XP_for_Itanium?taxonomyId=125) on 18 June 2013. Retrieved 17 October 2017.
29. "Microsoft Raises the Speed Limit with the Availability of 64-Bit Editions of Windows Server 2003 and Windows XP Professional" (https://news.microsoft.com/2005/04/25/microsoft-raises-the-speed-limit-with-the-availability-of-64-bit-editions-of-windows-server-2003-and-windows-xp-professional/) (Press release). Microsoft. April 25, 2005. Retrieved September 10, 2015.
30. Mashey, John (October 2006). "The Long Road to 64 Bits" (https://queue.acm.org/detail.cfm?id=1165766). *ACM Queue*. **4** (8): 85–94. doi:10.1145/1165754.1165766 (https://doi.org/10.1145%2F1165754.1165766). Retrieved 2011-02-19.
31. "Windows 7: 64 bit vs 32 bit?" (https://www.w7forums.com/threads/windows-7-64-bit-vs-32-bit.484/). W7 Forums. Archived (https://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.
32. "Frequently Asked Questions About the Java HotSpot VM" (http://java.sun.com/docs/hotspot/HotSpotFAQ.html#64bit_compilers). Sun Microsystems, Inc. Archived (https://web.archive.org/web/20070510204014/http://java.sun.com/docs/hotspot/HotSpotFAQ.html) from the original on 10 May 2007. Retrieved 2007-05-03.
33. "A description of the differences between 32-bit versions of Windows Vista and 64-bit versions of Windows Vista" (https://support.microsoft.com/en-us/help/946765/). Retrieved 2011-10-14.
34. Mark Russinovich (2008-07-21). "Pushing the Limits of Windows: Physical Memory" (https://blogs.technet.microsoft.com/markrussinovich/2008/07/21/pushing-the-limits-of-windows-physical-memory/). Retrieved 2017-03-09.

35. Chappell, Geoff (2009-01-27). "Licensed Memory in 32-Bit Windows Vista" (http://www.geoffchappell.com/notes/windows/license/memory.htm). *geoffchappell.com*. WP:SPS. Retrieved 9 March 2017.
36. "Inside Native Applications" (https://technet.microsoft.com/en-us/sysinternals/bb897447.aspx). Technet.microsoft.com. 2006-11-01. Archived (https://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.
37. Lincoln Spector (August 12, 2013). "Run an old program on a new PC" (https://www.pcworld.com/article/2045345/run-an-old-program-on-a-new-pc.html).
38. Peter Seebach (2006). "Exploring 64-bit development on POWER5: How portable is your code, really?" (http://www.ibm.com/developerworks/power/library/pa-openpower1/#N100C7).
39. Henry Spencer. "The Ten Commandments for C Programmers" (http://www.lysator.liu.se/c/ten-commandments.html).
40. "The Story of Thud and Blunder" (http://www.datacenterworks.com/stories/thud.html). Datacenterworks.com. Retrieved 2010-11-19.
41. "ILP32 and LP64 data models and data type sizes" (https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.3.0/com.ibm.zos.v2r3.cbcpx01/datatypesize64.htm). *z/OS XL C/C++ Programming Guide*.
42. "64-BIT PROGRAMMING MODELS" (http://www.unix.org/version2/whatsnew/lp64_wp.html).
43. "Using the ILP64 Interface vs. LP64 Interface | Intel Math Kernel Library for Linux" (https://software.intel.com/en-us/mkl-linux-developer-guide-using-the-ilp64-interface-vs-lp64-interface).
44. "Cray C/C++ Reference Manual" (https://web.archive.org/web/20131016001801/http://docs.cray.com/books/004-2179-001/html-004-2179-001/rvc5mrwh.html). August 1998. Table 9-1. Cray Research systems data type mapping. Archived from the original (http://docs.cray.com/books/004-2179-001/html-004-2179-001/rvc5mrwh.html#QEARLRWH) on October 16, 2013. Retrieved October 15, 2013.
45. "Cray C and C++ Reference Manual (8.7) S-2179" (https://pubs.cray.com/content/S-2179/8.7/cray-c-and-c++-reference-manual/about-the-cray-and-c++-reference-manual).
46. "Abstract Data Models - Windows applications" (https://docs.microsoft.com/en-us/windows/desktop/winprog64/abstract-data-models). May 30, 2018.

External links

- 64-bit Transition Guide, Mac Developer Library (https://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?" (https://web.archive.org/web/20010603003655/http://reality.sgi.com/mjk/64bit/64bit.html). Archived from the original (http://reality.sgi.com/mjk/64bit/64bit.html) on June 3, 2001. Retrieved September 26, 2012.
- 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)