32-bit

In computer architecture, **32-bit** integers, memory addresses, or other data units are those that are 32 bits (4 octets) wide. Also, 32-bit CPU and ALU architectures are those that are based on registers, address buses, or data buses of that size. 32-bit microcomputers are computers in which 32-bit microprocessors are the norm.

Contents

Range for storing integers

A 32-bit register can store 2^{32} different values. The range of integer values that can be stored in 32 bits depends on the integer representation used. With the two most common representations, the range is 0 through 4,294,967,295 ($2^{32} - 1$) for representation as an (unsigned) binary number, and −2,147,483,648 (-2^{31}) through 2,147,483,647 ($2^{31} - 1$) for representation as two's complement.

One important consequence is that a processor with 32-bit memory addresses can directly access at most 4 GiB of byte-addressable memory (though in practice the limit may be lower).

Technical history

Memory, as well as other digital circuits and wiring, was expensive during the first decades of 32-bit architectures (the 1960s to the 1980s).[1] Older 32-bit processor families (or simpler, cheaper variants thereof) could therefore have many compromises and limitations in order to cut costs. This could be a 16-bit ALU, for instance, or external (or internal) buses narrower than 32 bits, limiting memory size or demanding more cycles for instruction fetch, execution or write back.

Despite this, such processors could be labeled "32-bit," since they still had 32-bit registers and instructions able to manipulate 32-bit quantities. For example, the original Motorola 68000 had a 16-bit data ALU and a 16-bit external data bus, but had 32-bit registers and a 32-bit based instruction set. Such designs were sometimes referred to as "16/32-bit".[2]

However, the opposite is often true for newer 32-bit designs. For example, the Pentium Pro processor is a 32-bit machine, with 32-bit registers and instructions that manipulate 32-bit quantities, but the external address bus is 36 bits wide, giving a larger address space than 4 GB, and the external data bus is 64 bits wide, primarily in order to permit a more efficient prefetch of instructions and data.[3]

Architectures

Prominent 32-bit instruction set architectures used in general-purpose computing include the IBM System/360 and IBM System/370 (which had 24-bit addressing) and the System/370-XA, ESA/370, and ESA/390 (which had 31-bit addressing), the DEC VAX, the NS320xx, the Motorola 68000 family (the first two models of which had 24-bit addressing), the Intel IA-32 32-bit version of the x86 architecture, and the 32-bit versions of the ARM,[4] SPARC, MIPS, PowerPC and PA-RISC architectures. 32-bit instruction set architectures used for embedded computing include the 68000 family and ColdFire, x86, ARM, MIPS, PowerPC, and Infineon TriCore architectures.

Applications

On the x86 architecture, a 32-bit application normally means software that typically (not necessarily) uses the 32-bit linear address space (or flat memory model) possible with the 80386 and later chips. In this context, the term came about because DOS, Microsoft Windows and OS/2[5] were originally written for the 8088/8086 or 80286, 16-bit microprocessors with a segmented address space where programs had to switch between segments to reach more than 64 kilobytes of code or data. As this is quite time-consuming in comparison to other machine operations, the performance may suffer. Furthermore, programming with segments tend to become complicated; special *far* and *near* keywords or *memory models* had to be used (with care), not only in assembly language but also in high level languages such as Pascal, compiled BASIC, Fortran, C, etc.

The 80386 and its successors fully support the 16-bit segments of the 80286 but also segments for 32-bit address offsets (using the new 32-bit width of the main registers). If the base address of all 32-bit segments is set to 0, and segment registers are not used explicitly, the segmentation can be forgotten and the processor appears as having a simple linear 32-bit address space. Operating systems like Windows or OS/2 provide the possibility to run 16-bit (segmented) programs as well as 32-bit programs. The former possibility exists for backward compatibility and the latter is usually meant to be used for new software development.

Images

In digital images/pictures, 32-bit usually refers to RGBA color space; that is, 24-bit truecolor images with an additional 8-bit alpha channel. Other image formats also specify 32 bits per pixel, such as RGBE.

In digital images, 32-bit sometimes refers to high-dynamic-range imaging (HDR) formats that use 32 bits per channel, a total of 96 bits per pixel. 32-bit-per-channel images are used to represent values brighter than what sRGB color space allows (brighter than white); these values can then be used to more accurately retain bright highlights when either lowering the exposure of the image or when it is seen through a dark filter or dull reflection.

For example, a reflection in an oil slick is only a fraction of that seen in a mirror surface. HDR imagery allows for the reflection of highlights that can still be seen as bright white areas, instead of dull grey shapes.

File formats

A 32-bit file format is a binary file format for which each elementary information is defined on 32 bits (or 4 bytes). An example of such a format is the Enhanced Metafile Format.

See also

- 16-bit
- 64-bit
- History of video games (32-bit era)
- Word (data type)
- Physical Address Extension (PAE)

References

1. Patterson, David; Ditzel, David (2000). *Readings in Computer Architecture*. San Diego: Academic Press. p. 136. ISBN 9781558605398.
2. "68000 users manual" (https://www.nxp.com/files/archives/doc/ref_manual/M68000PRM.pdf) (PDF).
3. Gwennap, Linley (16 February 1995). "Intel's P6 Uses Decoupled Superscalar Design" (http://www.eecg.toronto.edu/~moshovos/ACA05/read/ppro1.pdf) (PDF). Microprocessor Report. Retrieved 3 December 2012.
4. "ARM architecture overview" (https://web.eecs.umich.edu/~prabal/teaching/eecs373-f10/readings/ARM_Architecture_Overview.pdf) (PDF).
5. *There were also variants of UNIX for the 80286.*

External links

- HOW Stuff Works "How Bits and Bytes work"
- Ken Colburn on LockerGnome.com: *32-Bit Vs. 64-Bit Windows*