

By: [Caleb 2008](#), Science News Editor

How Cache Memory Works

Fast information retrieval technologies

Have you ever noticed how as soon as you enter a computer shop the seller starts talking in some alien language, mentioning words such as 'L2 cache', '16 megabytes buffer' or 'virtual memory', but never has time to explain what those words really mean? If you have, then you're in the right place, because this article describes exactly what a memory cache is and how it works. Caching is a technology based on the memory subsystem of your computer, having as main role the increase of computing power while maintaining computer price at a low level. Let's take for example a Random Access Memory. RAM's are basically used to temporarily store data retrieved from the hard disk - or any other sources - so that the processor has faster access to the information required (RAMs are much faster than hard drives, floppy drives or CD-ROM drives). However, the RAM is even slower than the processor clock. A typical computer processor is capable of completing an operation in less than 2 nanoseconds, while RAMs are accessed in about 60 nanoseconds, that's 15 times slower. To speed up this process, cache memories are used, low capacity memories with speeds almost double of that of the RAM.

How it works Let's take for example an action when the processor requests an information stored on the hard drive. Fetching information from the hard drive takes time, because hard drives are very sluggish in relation to the processor clock. Thus, every time information is retrieved, processing time is lost, although the requested information may be the same, over and over again. The cache memory has the role of backing up the RAM. This means that before retrieving the information from the hard drive, the computer first looks for it in the cache memory. If it is present in the cache, then it is instantly relayed to the processor. This could quickly turn into a disadvantage if the requested information cannot be found in the cache, because processing time is first being lost by checking the cache, then actually retrieving the information from the hard drive. The two processes are called 'cache hit' and 'cache miss', in case the requested information cannot be found in the cache memory. Usually, the cache maximum size is extremely small in relation to the large storage devices. However, with the help of multiple cache layers processing speed can be further improved.

Types of cache Typical computers usually have one to three cache levels, L1, L2 and L3. L1 cache refers to small memory systems built into the processor. L2 cache, on the other hand, is represented by small memory banks posted on the motherboard, while processors with two cache layers - the memory banks on the motherboard exist between the processor and the main system memory - represent a level 3 cache. Level 1 and 2 caches caching the main memory may also exist on slow data storing devices such as the hard drive or the CD-ROM drives. RAMs, hard drives, CD-ROM drives, even the Internet are in fact cache memories, each with different accessing speeds and storing capacities. L1 cache has capacities between 4 to 16 kilobytes and accessing speeds of 10 nanoseconds, while the L2 cache can reach sizes of 512 kilobytes and a speed of only 20 to 30 nanoseconds. The typical accessing speed of a RAM is 60 nanoseconds and a capacity of up to several gigabytes. Hard drives - 12 milliseconds and several hundred gigabytes to a couple of terabytes of storing capacity, while the Internet has unlimited size and accessing speeds ranging from one second up to 3 or 4 days.

Why aren't all caches as fast as L1? To make all the cache memories run at the same speed as the L1 cache - 10 nanoseconds - would be extremely expensive and in fact the computer doesn't need large amounts of cache memory due to locality of reference. Locality of reference means that no matter how large or small in size a running application is, only a small part of the respective program is being stored into the cache at one given time. This is because the vast majority of programs mostly contain fairly large amounts of code lines which are executed only once. The other lines are represented by program loops which are executed over and over again. Less than 10 percent of processor time is spent by the programs running at one time.