Cache memory

Cache memory, also called CPU memory, is random access memory (RAM) that a computer microprocessor can access more quickly than it can access regular RAM. This memory is typically integrated directly with the CPU chip or placed on a separate chip that has a separate bus interconnect with the CPU.

The basic purpose of cache memory is to store program instructions that are frequently re-referenced by software during operation. Fast access to these instructions increases the overall speed of the software program.

As the microprocessor processes data, it looks first in the cache memory; if it finds the instructions there (from a previous reading of data), it does not have to do a more time-consuming reading of data from larger memory or other data storage devices.

Most programs use very few resources once they have been opened and operated for a time, mainly because frequently re-referenced instructions tend to be cached. This explains why measurements of system performance in computers with slower processors but larger caches tend to be faster than measurements of system performance in computers with faster processors but more limited cache space.

Multi-tier or multilevel caching has become popular in server and desktop architectures, with different levels providing greater efficiency through managed tiering. Simply put, the less frequently access is made to certain data or instructions, the lower down the cache level the data or instructions are written.

Ön Bellek

İşlemci hafızası olarak da adlandırılan ön bellek, bir bilgisayar işlemcisinin sıradan bir RAM'a kıysla daha hızlı erişebildiği bir rastgele erişilebilir hafızadır (RAM). Bu hafıza tipik olarak doğrudan işlemci çipiyle birlikte bütünleştirilmiştir veya CPU ile içten bağlı ve ayrı bir yola sahip müstakil bir çip üzerine yerleştirilmiştir.

Ön belleğin temel amacı, işlemler esnasında yazılım tarafından sıkça tekrar başvurulan program komutlarını depolamaktır. Bu komutlara hızlı erişim yazılım programının tamamının hızını artırır.

Mikroişlemci veriyi işlediğinde ilk olarak ön belleğe bakar, eğer burada komutlar bulursa (bir önceki veri okumasından) daha geniş bir hafızadan veya diğer veri depolama aygıtlarından veri okuma için daha fazla zaman harcamasına gerek yoktur.

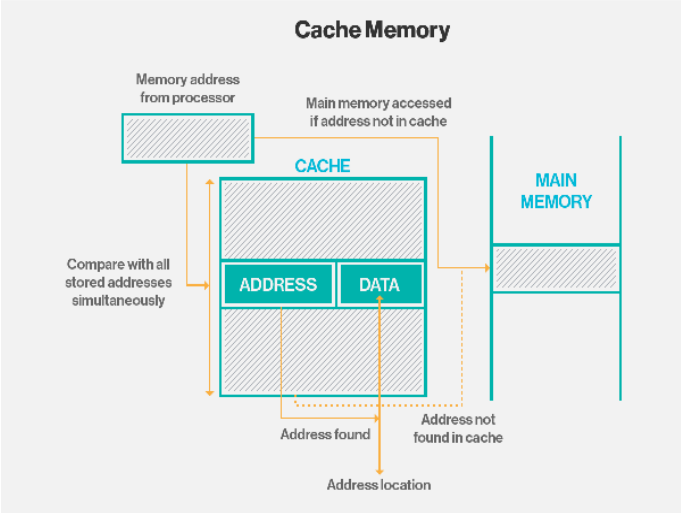
Birçok program bir süreliğine açıldıklarında ve yürütüldüklerinde çok az kaynak kullanırlar, bunun başlıca sebebi sıkça tekrarlanan başvuruların ön belleğe alınmaya yatkın olmasıdır. Bu da düşük işlemcili fakat geniş ön bellekli bilgisayarların, sistem performans testlerinde daha hızlı işlemcili fakat daha sınırlı önbellek alanı olan bilgisayarlardan neden daha hızlı olmaya yatkın olduklarını açıklamaktadır.

Çok katlı veya çok seviyeli ön bellekleme, sunucu ve masaüstü mimarisinde, daha fazla verim sağlayan farklı seviyeler ve yönetilebilen kat yapısı sayesinde popüler hale geldi. Özetle belirli verilere veya komutlara ne kadar az sıklıkla erişim yapılırsa, ön belleğe yazılan veri veya komutların seviyesi o kadar azalır.

Cache memory levels explained

Cache memory is fast and expensive. Traditionally, it is categorized as "levels" that describe its closeness and accessibility to the microprocessor:

- **Level 1 (L1) cache** is extremely fast but relatively small, and is usually embedded in the processor chip (CPU).
- **Level 2 (L2) cache** is often more capacious than L1; it may be located on the CPU or on a separate chip or coprocessor with a high-speed alternative system bus interconnecting the cache to the CPU, so as not to be slowed by traffic on the main system bus.
- **Level 3 (L3) cache** is typically specialized memory that works to improve the performance of L1 and L2. It can be significantly slower than L1 or L2, but is usually double the speed of RAM. In the case of multicore processors, each core may have its own dedicated L1 and L2 cache, but share a common L3 cache. When an instruction is referenced in the L3 cache, it is typically elevated to a higher tier cache.



Memory cache configurations

Caching configurations continue to evolve, but memory cache traditionally works under three different configurations:

Ön Bellek Seviyelerinin Açıklanması

Ön bellek hızlı ve pahalıdır. Ön bellek, geleneksel olarak mikro işlemciye erişebilirliğine ve yakınlığına göre tanımlanan seviyeler şeklinde sınıflandırılır.

- **Seviye 1 (L1) önbelleği** oldukça hızlı fakat nispeten küçüktür ve genellikle mikro işlemci çipine (CPU) gömülüdür.
- **Seviye 2 (L2) önbelleği** çoğunlukla L1'den daha fazla kapasiteye sahiptir; CPU üzerine yerleştirilebilir veya ayrı bir çip üzrine yada ana sistem veri yolundaki trafik tarafından yavaşlatılmaması için CPU ön belleğine içten bağlı yüksek hızlı alternatif sistem veri yolllu yardımcı bir işlemci üzerine yerleştirilebilir.
- **Seviye 3 (L3) önbelleği** tipik olarak L1 ve L2'nin performansını arttırmak için çalışan özel olarak yapılmış bir hafızadır. L1 ve L2 den önemli ölçüde yavaş olabilir ama genellikle RAM'in hızını ikiye katlar. Çok çekirdekli işlemcilerde herbir çekirdeğin kendine tahsisli bir L1 ve L2 ön belleği olabilir lakin ortak bir L3 ön belleğini paylaşırlar. L3 ön belleğindeki bir komuta başvurulduğunda tipik olarak komut, daha yüksek bir kattaki önbelleğe yükseltilir.

Hafıza Önbellek Konumlandırılması

Ön bellek konfigürasyonları gelişmeye devam etmektedir fakat hafıza önbelleği geleneksel olarak üç farklı konfigürasyon (biçim) altında çalışır.

- **Direct mapping**, in which each block is mapped to exactly one cache location. Conceptually, this is like rows in a table with three columns: the data block or cache line that contains the actual data fetched and stored, a tag that contains all or part of the address of the fetched data, and a flag bit that connotes the presence of a valid bit of data in the row entry.
- **Fully associative mapping** is similar to direct mapping in structure, but allows a block to be mapped to any cache location rather than to a pre-specified cache location (as is the case with direct mapping).
- **Set associative mapping** can be viewed as a compromise between direct mapping and fully associative mapping in which each block is mapped to a subset of cache locations. It is sometimes called *N-way set associative mapping*, which provides for a location in main memory to be cached to any of "N" locations in the L1 cache.

Specialized caches

In addition to instruction and data caches, there are other caches designed to provide specialized functions in a system. By some definitions, the L3 cache is a specialized cache because of its shared design. Other definitions separate instruction caching from data caching, referring to each as a specialized cache.

Other specialized memory caches include the translation lookaside buffer (TLB) whose function is to record virtual address to physical address translations.

Still other caches are not, technically speaking, memory caches at all. Disk caches, for example, may leverage RAM or flash memory to provide much the same kind of data caching as memory caches do with CPU instructions. If data is frequently accessed from disk, it is cached into DRAM or flash-based silicon storage technology for faster access and response.

- **Doğrudan Haritalama**, her bloğun tam olarak bir önbellek konumuna haritalanması. Kavramsal olarak bu üç sütunlu bir tablodaki satırlar gibidir: data bloğu veya elde adilen ve depolanan gerçek veriyi barındıran önbellek hattı, elde edilen verinin adresinin tümünü veya bir parçasını içeren ve giriş satırındaki doğru veri bitinin varlığını belirten bir bayrak biti.
- **Tam ilişkisel haritalama** yapısal olarak doğrudan haritalamaya benzerdir, fakat bir bloğun ön tanımlı bir konuma doğrudan haritalamada olduğu gibi) haritalanmasından ziyade bloğun herhangi bir ön bellek konumuna adreslenmesine izin verir .
- **Küme ilişkisel haritalama,** doğrudan haritalama ve her bir bloğun önbellek konumlarının bir alt kümesine haritalandığı tam ilişkisel haritalama arasındaki bir uzlaşma olarak görülebilir. ana bellekteki bir konumun L1 önbelleğindeki herhangi bir "N" konumuna önbelleklenmesini sağlayan küme ilişkisel haritalama, N-Yolu olarak da isimlendirilir.

Özelleştirilmiş Önbellekler

Veri ve komut önbelleklerine ilave olarak bir sistemde özelleştirilmiş fonksiyonlar sağlaması için dizayn edilmiş diğer önbellekler vardır. Bazı tanımlamalara göre, L3 önbelleği paylaşımlı dizaynı sebebiyle özelleştirilmiş bir önbellektir. Diğer tanımlamalar komut önbelleklemesini veri önbelleklemesinden ayırırlar ve her birine özelleştirilmiş ön bellek olarak bakarlar.

Diğer özelleştirilmiş önbellek hafızası, işlevi sanal adresi fiziksel adrese çevirmek olan dönüşüm hafızasını (TLB) içerir.

Teknik konuşmayla, henüz diğer önbellekler hafıza önbelleği sayılmazlar. Örneğin, Disk önbelleği, RAM'a veya flaş belleğe hafıza önbelleğinin işlemci komutlarıyla yaptığı gibi oldukça benzer tipte veri önbelleklemesi sağlamak için baskı yapabilir. Eğer veri sık bir şekilde diskten erişiliyorsa, daha hızlı erişim ve yanıt için DRAM'a veya flaş tabanlı silikon depolama teknolojisine önbelleklenir.

caches also exist for such applications as Web browsers, databases, network address binding and client-side Network File System protocol support. These types of caches might be distributed across multiple networked hosts to provide greater scalability or performance to an application that uses them.

Increasing cache size

L1, L2 and L3 caches have been implemented in the past using a combination of processor and motherboard components. Recently, the trend has been toward consolidating all three levels of memory caching on the CPU itself. For this reason, the primary means for increasing cache size has begun to shift from the acquisition of a specific motherboard with different chipsets and bus architectures to buying the right CPU with the right amount of integrated L1, L2 and L3 cache.

http://searchstorage.techtarget.com/definition/cache-memory

Web tarayıcıları, veritabanları, ağ adresi bağlama ve istemci taraflı ağ dosya sistemi protokol desteği gibi uygulamlar için de önbellekler mevcuttur. Bu tip önbellekler, onları kullanan bir uygulamaya daha büyük tırmanabilirlik veya performans sağlamak için çoklu ağ sunucularının genişliğine göre paylaştırılabilir.

Önbellek boyutunu arttırma

L1, L2 ve L3 önbellekler, geçmişte bir işlemci kombinasyonu ve anakart bileşenleri kullanılarak uygulandı. Son zamanlarda eğilim, üç önbellek hafıza seviyesinin de işlemcinin üzerinde birleştirilmesi yönünde. Bu sebeple, önbellek arttırımının esas sermayesi, farklı çipsetli ve veri yolu mimarili belirli bir anakart elde etmekten doğru miktarda L1,L2 ve L3 önbellek entegre edilmiş doğru mikroişlemci satın almaya kaymaya başladı.