Learn how Cache works (Part 2)

In this section we are only interested in previous historical aspects of Cache. If you are not interested in this topic, you can skip it to read the next section.

Learn how cache works (Part 1)

History of Memory Cache on personal computers (PC)

In this section we are only interested in previous historical aspects of Cache. If you are not interested in this topic, you can skip it to read the next section.

Cache memory was first used on 386DX computers. Although the CPU itself has no internal cache, its support circuit - chipset - has a memory Cache controller. Because of this, the memory cache at this time is outside the CPU and is optional, meaning that the motherboard manufacturer can add or not. If you have a motherboard that does not have cache memory, your computer will be much slower than computers with this component. The amount of cache offered varies and depends on the motherboard model and the typical values ??for that time are 64 KB and 128 KB. At the same time, the memory cache controller used an architecture known as 'write-through', for write operations - that is, when the CPU wants to store data in memory, the controller The memory cache will update RAM immediately.

With 486DX microprocessors, Intel added a small amount (8KB) ??of memory cache inside each CPU. This internal memory cache is called L1 (level 1) or 'internal', while the external memory cache is called (level 2) or 'external'. The number and existence of the external memory cache depends on the model of the motherboard. The typical number for that time is 128 KB and 256 KB. Later the 486 models added the Cache architecture 'write back', which is the architecture that has been used to this day, the RAM write operations are not updated immediately but the CPU stores data. Whether the memory cache and the memory controller update RAM memory only when there is no Cache.

Then with the first Pentium processors, Intel created two separate internal memory caches, one for instructions and one for data (at this time each memory cache was 8 KB). This architecture is still in use today and that's why sometimes you still find that the L1 memory cache exists written 64 KB + 64 KB (for example) - this is because there is an L1 64KB instruction cache and a 64KB L2 memory cache. We will explain to you the difference of these two types of memory cache. At that time, L2 memory cache is usually placed on the motherboard, so its number and existence depend on the motherboard model. Obviously, the system without memory cache is unacceptable. The typical number for that time is 256 KB and 512 KB.

AMD K5, K6 and K6-2 processors have also used this architecture, with K6-III having a third memory cache (L3, level 3).

The problem with the memory cache outside of L2 is that it is accessed with a lower clock rate because the

486DX2 generation of the CPU internal clock rate is completely different from the external clock rate of the CPU. For example, the Pentium-200 works internally at 200MHz, it has access to its L2 cache at 66MHz.

After P6 architecture, Intel switched to memory cache from the motherboard to the inside of the CPU - which allowed the CPU to access the internal clock rate - except Pentium II (Cache memory was not placed inside CPU but on the same printed circuit board - where the CPU is mounted (this board is placed inside a container)), again running half of the CPU's internal clock speed. On Celeron-266 and Celeron-300, these models do not have memory cache (so they are the worst CPUs in history).

The architecture used to this day is similar: both L1 and L2 memory caches are located inside the CPU and run at the CPU internal clock rate. Therefore, the amount of memory cache you have on the system will depend on the CPU model; There is no way to increase the amount of memory cache without replacing the CPU.

Consider the memory cache

In Figure 2, you will see the basic block diagram of a single-core CPU. The specific block diagram will vary depending on the CPU.

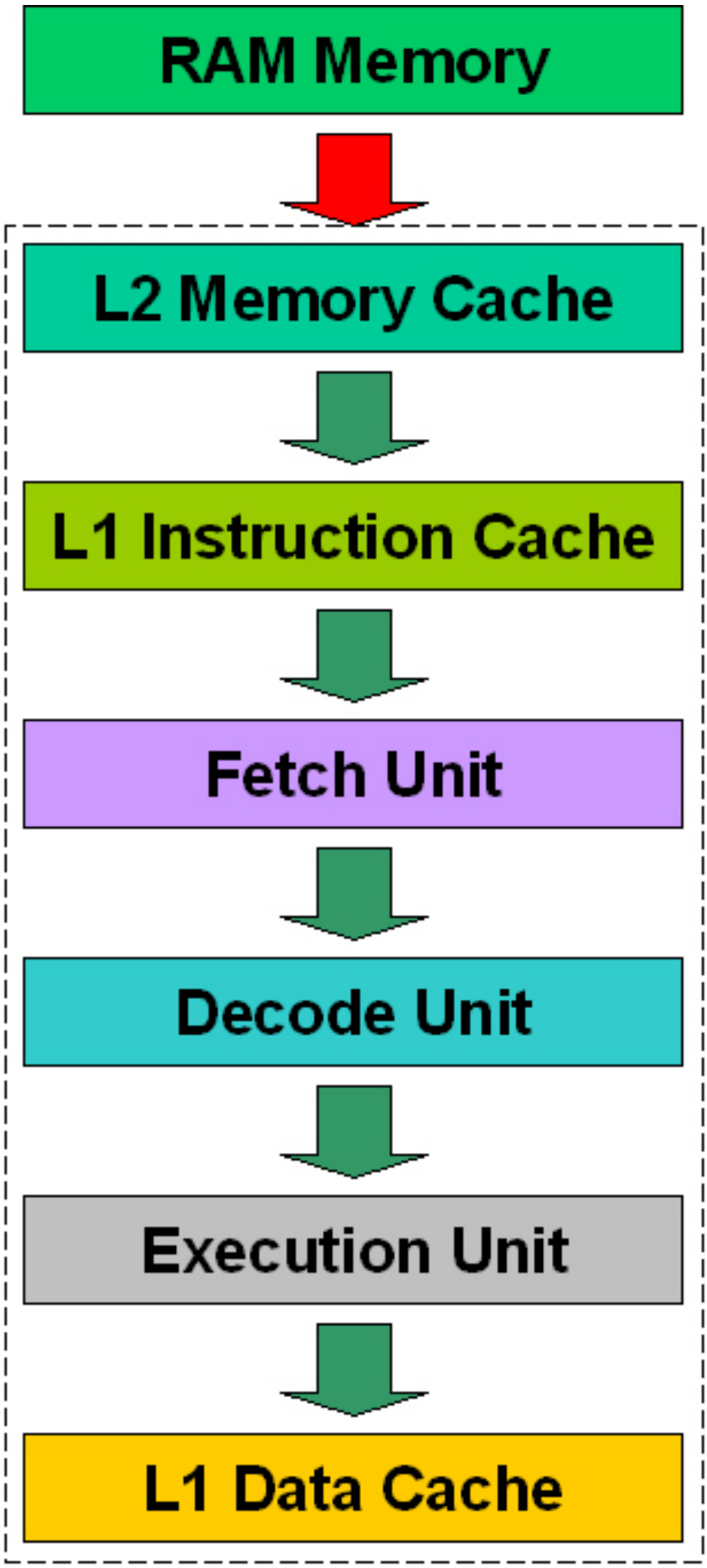

RAM Memory
L2 Memory Cache
L1 Instruction Cache
Fetch Unit
Decode Unit
Execution Unit
L1 Data Cache

Figure 2: Basic block diagram of a CPU
The dotted line in Figure 2 shows the body of the CPU, because the RAM memory is located outside the CPU. Data path between RAM and normal 64-bit CPU (or 128 when the two-channel configuration is used), running at the memory clock rate or the external CPU clock (or the bus's memory clock, in case for AMD microprocessors).

All circuits inside the dot run with the CPU clock rate. Depending on the CPU, some of its components can even run at a higher clock rate. The path between CPU blocks can also be wider, meaning more bits will be transmitted per clock cycle (more than 64 or 128). For example, the data path between L2 cache and L1 instruction cache on modern microprocessors is usually 256-bit wide. The data path between the L1 instruction cache and the CPU fetch unit also changes depending on the model of each CPU - 128 bits is typical value, but at the end of this tutorial we will introduce one Technical indicators of the main memory cache for CPUs on the market today. The higher the number of bits transmitted over a clock cycle, the faster the transmission (in other words, the faster the transfer rate).

In general, all modern CPUs have up to three memory caches: L2 is a larger memory cache and can be found between the RAM and the L1 instruction cache, which holds both instructions and data; The instruction cache L1 is used to store instructions executed by the CPU and store the data so that it can be written back to memory.

L1 and L2 mean 'Level 1' and 'Level 2', referring to the distance from them to the CPU core (execution block). There is a doubt that why there are three separate memory caches (L1 memory cache, L1 instruction memory cache and L2 cache).

To make the static memory latency drop to '0' is very difficult, especially for CPUs running at very high clock speeds. Since the production of static RAM has an approximate '0' latency, it is difficult for manufacturers to use a memory type only above the memory cache L1. The L2 memory cache uses static RAM that is not as fast as the memory used on the L1 memory cache, which is because it has a certain delay, so it will be slightly slower than the L1 memory cache.

Notice in Figure 2 that we will see that the L1 instruction cache works like an 'input cache', while the L1 data cache works like an 'output cache'. L1 instruction cache (usually smaller than L2 cache) is more effective when the program starts to repeat some of its small parts, which is also because the required instructions will be closer to the fetch block.

It is also rarely mentioned, but the L1 instruction cache is also used to store other data with decoded instructions. Depending on the CPU it can be used to store some pre-decoding data and branching information (in general, control data will increase the speed of the decoding process) and sometimes the Cache L1 instructs even bigger than it has stated, this is because the manufacturer often does not add to the available expansion space for these extended pieces of information.

In the CPU specification page, L1 cache can have many different types. Some manufacturers list two L1 caches completely separate (sometimes call the instruction cache 'I' and the data cache 'D'), sometimes adding both the number and symbol part 'separated', if '128 KB, separated' then that means the Cache instructs 64KB and 64KB Data Cache, some firms have done so that you can guess the overall number and have to divide it to get capacity of each Cache. However, there are exceptions to CPUs built on Netburst architecture, such as Pentium 4, Pentium D, Pentium 4 based on Xeon and Celeron CPUs from Pentium 4.

Microprocessors based on Netburst architecture have no instruction cache L1, instead they have a trace execution cache (or can be called traces), this cache is placed between the decoding block and the execution block , save the decoded instructions. Therefore, it can be said that the L1 instruction cache is it, but is hidden under a completely different name and placed in a different location. We will mention this here because this is a very good mistake, people often think that Pentium 4 CPUs do not have L1 instruction cache. This leads to the phenomenon of comparing Pentium 4 to other CPUs, often thinking that its L1 cache is smaller, because they only count 8KB of L1 data cache. The cache executes traces of CPUs built on Netburst architecture of 150KB.

L2 Memory Cache on multi-core CPUs

On CPUs with more than one core, the L2 Cache architecture varies a lot, this change depends on the CPU type.

With dual-core Pentium D and AMD CPUs built on K8 architecture, each CPU core has its own L2 memory cache. That's why each core works as if it were working for a standalone CPU.

Intel's dual-core CPUs are built on Core and Pentium M architectures, so the two L2 memory caches can be shared between two cores.

Intel said that this shared architecture offers better performance because on a separate Cache method, at one time, one core might be overloaded while the other is not used or used. Use up the performance on its own L2 Cache. When this happens, the overloaded core will retrieve data from the main RAM memory, although the space on the L2 memory cache is completely empty, which should have been used to store the data and prevent the core from overloading. Accessing data from RAM reduces the overall system performance. With this new method, the Core 2 Duo processor with 4MB L2 memory cache, this one can use up to 3.5MB while the other core uses 0.5MB, quite the same with the coefficient Fixed division 50% -50% as on dual-core CPUs.

In other words, Intel's current quad-core CPUs such as Core 2 Extreme QX and Core 2 Quad use 2 dual-core chips, meaning that this sharing only occurs between cores 1 and 2 and 3. and 4. Currently, Intel has planned quad-core CPUs using a single chip. With this method, L2 cache will be shared between four cores.

On Figure 3 you can see the comparison between these L2 memory cache solutions.

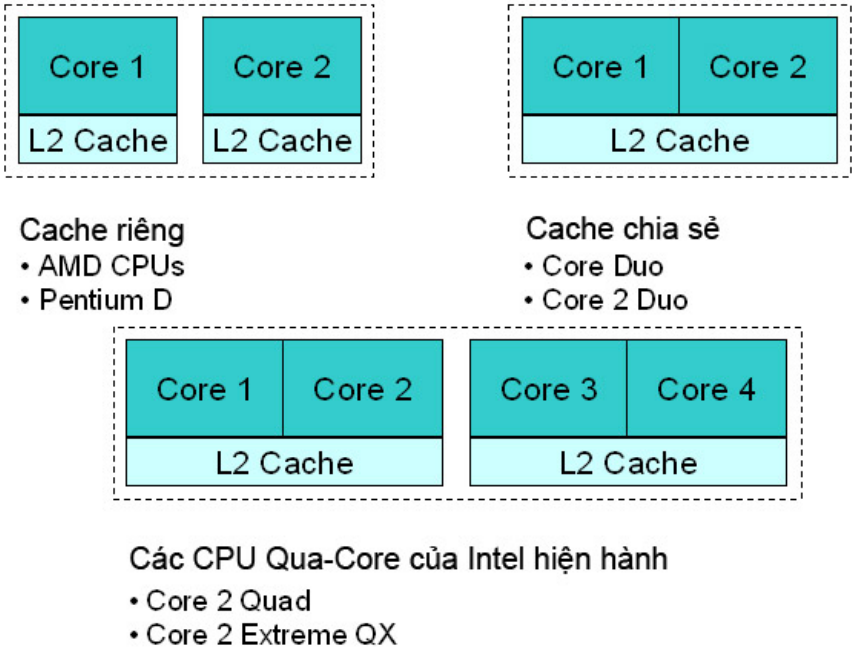


Figure 3: Comparison of existing L2 memory cache solutions on multi-core CPUs

AMD processors built on K10 architecture will have a shared L3 Cache located inside the CPU, and there is a hybrid between these two methods. This problem is shown in Figure 4. The size of this cache will depend on the CPU model, just like what happens with the size of the L2 Cache.

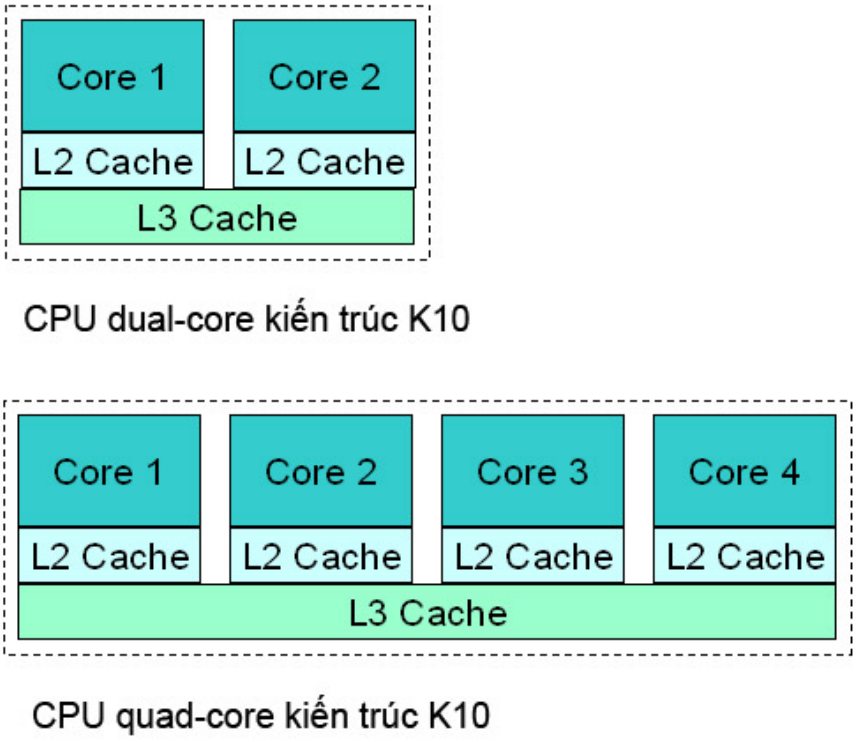
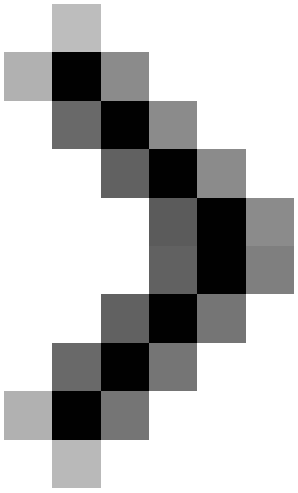
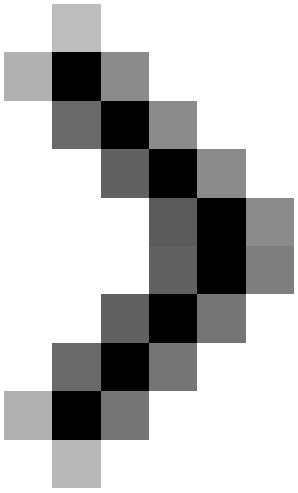


Figure 4: Architecture K10 Cache



Learn how Cache works (Part 3)



Learn how cache works (End section)

You finished reading the article "**Learn how Cache works (Part 2)**" edited by the **TipsMake** team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.