12.9. Cache

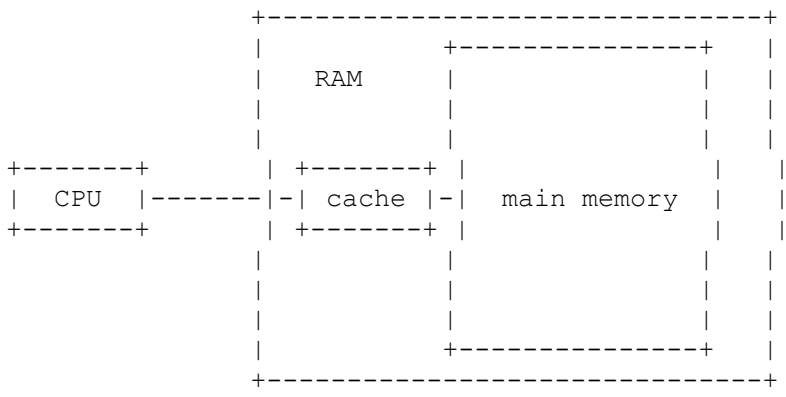
12.9.1. Overview

Main memory consists of large RAM chips outside the CPU, usually mounted in special slots on the *motherboard*. Because main memory has a large number of addresses, it takes a long time to decode the address before it can read or write the memory cell. Also, RAM chips are designed to pack a large amount of memory into a small space, so power and heat dissipation take precedence over speed. This often results in the CPU sitting idle for many clock cycles while waiting for a response from the memory unit.

To reduce this memory bottleneck, most modern systems employ *cache* memory. The word cache comes from French, and means “hidden”.

The cache RAM is a small, very fast RAM unit that is managed entirely by the hardware. The full details of cache operation are left to a course in computer architecture.

Access to cache requires a fraction of the time it takes to access main memory. Neither user programs nor the operating system have direct access to cache memory, but knowing that it is there and how it works can help programmers take better advantage of it.



Each time main memory is read, a copy of the data is stored in the cache. The next time that same main memory address is accessed, the cache is checked first. If the data is still in the cache, it is taken from there, and it is not necessary to access the main memory chips at all. Since cache is much smaller than main memory, only recently accessed data will be present in the cache. The cache fills quickly when

programs are running, and when it is full, old data is overwritten by the most recent data read from main memory.

The *hit ratio* for cache is defined as the number of RAM references satisfied by the cache divided by the total number of RAM references. The cache hit ratio is not as critical as the virtual memory hit ratio, but a good hit ratio can double or triple program performance. It is common in well-designed software to see a cache hit-ratio around 0.9, even if the size of cache is 1/1000 the size of main memory.

If a cache hit ratio is 0.8, cache access take 3ns, and main memory access takes 10ns, what is the average memory access time?

```
avg access time = 0.8 * 3ns + 0.2 * 10ns
                = 4.4ns
```

With a fairly good hit ratio, cache can more than double average memory speed. Experience has shown that a hit ratio of 0.8 is achievable with a very small amount of cache, and well-written program code. Given that a large percentage of memory references by most subprograms are to the loop counters and a few other scalar variables, hit ratios tend to be high unless arrays larger than the cache memory are in heavy use.