The Different Mapping Techniques in Cache Memory

| Paul Vincent B. Espina (Researcher) | Lorenzo R. Macaso (Leader) | Maria Pamela P. Tagayon (Rapporteur) |

Abstract

The fastest among all the memories in a computer, the cache memory, is a computer memory that helps the processor read data faster since data stored within these cache are usually the computer programs, applications, instructions, and data that are frequently used. Mapping techniques are introduced for the ease of communication with data from main memory to cache memory, these techniques consist of direct mapping, associative mapping, and set-associative mapping. Cache memory serves as an improvement in data retrieval since it can relay data to the processor quickly and efficiently.

I. Cache Memory

A cache memory is a way of fast communication within the processor and data it needs since there is familiarity with the commonly used data that is stored within the cache. Static random access memory (SRAM) is used on the cache memory since SRAM gives faster access to data but much more expensive than the dynamic random access memory (DRAM), this sacrifices storage but provides speed. The cache memory has different levels that help read data faster and efficiently.

Level 1 Cache (L1) is the fastest among the cache among the three since it is situated in the CPU. It has two kinds of caches: the instruction cache and the data cache, as the name suggests, which stores the instructions and data in the CPU, respectively. Level 2 Cache (L2) is slower than L1 cache, can either be situated also inside the CPU, but usually it is outside. This cache can be shared among the cores present in the CPU or can be set-up individually for each core. Level 3 Cache (L3) is the slowest among the three, this cache is not present in all processors, it is usually found within high-end processors. Its purpose is to enhance the performance of the L1 and L2 caches.

II. Mapping Techniques

There are three different mapping techniques in Cache Memory, the first would be Direct Mapping. It is the simplest technique out of the three where it maps out each block of main memory into only one possible cache line. Meaning that if a line is previously taken up by a memory when a memory is to be uploaded, then the old block is thrashed. The next method would be Associative Mapping. In this technique, the associative memory is used to store content and addresses of the memory word. This is so much more flexible compared to Direct Mapping because any block of the Main memory can reside in any cache block position. It is considered to be the most flexible mapping form out of the three. The last technique would be called the Set-Associative Mapping. This is almost similar to the previous one which is the Associative Mapping as blocks of cache are grouped together into sets and the mapping allows a block of main memory to reside in any block of the specific set. This is also known as the enhanced form of direct mapping as it addresses the drawbacks of the possible thrashing in direct mapping.

III. Difference of Mapping Techniques

In a cache system, Direct Mapping maps each block of main memory into one possible cache line, their hit ratio is also poor due to it only having one fixed location. However, the Associative mapping enables each main memory block to be loaded by any line of the cache, therefore their hit ratio increased as they can take any location in a cache. In set associative mapping, the cache is divided into numerous sets of cache. In this case, the hit ratio is more compared to direct mapped cache and amount of energy consumption is less as compared to associative cache because of limited no tag patterns need to be searched .

IV. Relevance of Using Different Mapping Techniques

Caches are a hardware/software that are used to keep track of the recently used data and to make processors faster by reducing the performance wall. A cache memory needs to be smaller in size compared to main memory for it is placed closer to the execution unit in the processor.

The cache is made out of blocks where each block has a smallest storage element. In a cache design, *placement* is one of the most important parameters because it tells us how the blocks have to be placed in a cache and will then, tells us how fast can we access the cache and how the latency could be reduced.

V. References

[1] W. Stallings. "Cache Memory" in *Computer Organization and Architecture Designing for Performance*, 10th ed., Hoboken, NJ: Pearson Education, 2016, pp. 133-144.

[2] “Cache memory in Computer Organization”. Internet:https://www.geeksforgeeks.org/cache-memory-in-computer-organization/#:~:text=Usually%2C%20the%20cache%20memory%20can,specified%20by%20a%20mapping%20function. June 08,2020 [Mar. 08, 2021]

[3] “Notes on cache memory.” Internet: http://www.bowdoin.edu/~allen/courses/cs220/lab7/notes.html ,n.d. [Mar. 08, 2021]

[4] “What is memory caching? How memory caching works.” Internet: https://hazelcast.com/glossary/memory-caching/, Mar. 19, 2020 [Mar. 08, 2021]

[5] “Computer organization and architecture - mapping functions and replacement algorithms.”Internet:https://examradar.com/mapping-functions-replacement-algorithms/#:~:text=The%20mapping%20functions%20are%20used,main%20memory%20to%20cache%20memor, Oct. 2, 2020 [Mar. 08, 2021]

[6] “SRAM (static random access memory)” Internet:https://whatis.techtarget.com/definition/SRAM-static-random-access-memory, Apr. 2005 [Apr. 2, 2021]