An Analysis of Cache Memory

Tyler S. Unsworth

CS 312 – Computer Architecture II

Dr. Donald Davendra

Central Washington University

Abstract

This research paper aims to investigate the functionality and processes of computer architecture in the scope of cache memory. Memory systems within a computer allow for the storage of information, and data. These are also very slow and cannot keep up with the speed at which the CPU can process that data. Cache memory helps bridge that gap by offering the fastest memory available because of its multilevel architecture. Therefore, this paper sets to accomplish discussing the architecture of cache memory and contrasting it to other memory counterparts like RAM, SSD, and HDD. Additionally, there will be a focus on the functionality of the different levels contained within the cache memory.

Keywords: Cache Memory, L1, L2, L3, Hard Disk, Ram, SSD, Multi-core design

Contents

[Abstract 2](#_Toc86847443)

[Cache Memory 4](#_Toc86847444)

[Brief Overview of Memory Architectures 4](#_Toc86847445)

[Cache Memory Hierarchy 5](#_Toc86847446)

[Cache Memory Architecture 6](#_Toc86847447)

[Cache Memory Levels 7](#_Toc86847448)

[Conclusion 8](#_Toc86847449)

[References 9](#_Toc86847450)

[Table of Figures 10](#_Toc86847451)

Table of Figures

[Figure 1 Speed vs. Cost 9](#_Toc86764415)

[Figure 2 Hard Disk 9](#_Toc86764416)

[Figure 3 Cache Memory 9](#_Toc86764417)

[Figure 4 Cache Location 9](#_Toc86764418)

Cache Memory

In the modern world of the internet of things, information and big data are all around us. From what we see, to what we hear, to what we upload to the internet, and even download from it is ready to be processed and analyzed. As big data is becoming a more prominent part of our society and increasingly more involved in our day-to-day operations has driven the innovation of processor architecture design. However, there hasn’t been much innovation in storage or on-chip off-chip memory architectures. With processor innovation leading towards multi-core designs that vastly improves performance, calls to action the ever-growing need to process big data at a faster, more efficient rate. Dedicating resources to research novel innovations toward cache memory may be a viable solution to this problem in accelerating the pace we can process data.

Therefore, this paper aims to investigate the functions and processes in the scope of cache memory systems while also analyzing the different performance scenarios with varying architectures at the cache level. The cache memory component of a computer is the fastest memory available within the architecture of a computer compared to its other memory counterparts. Additionally, there will be an analysis of the different levels of cache memory in the form of L1, L2, and L3. There will also be a brief overview of the history, evolution, and future of this component as well.

# Brief Overview of Memory Architectures

Modern-day computers are composed of intricate systems which are comprised of smaller sub-systems. Computer memory is no exception to that. Computers have various types of memory in the form of hard disks, solid-state drives, registers, and cache systems (see figure 1). These components makeup computer storage and memory, which allows the computer to store, process, and analyze data in an efficient, timely manner.

Hard disk drives are composed of magnetized, mechanical rotating disks which stores data by magnetizing certain regions on the disk (see figure 2). If the region is magnetized, then it is processed by the computer as a 1, and if not, then processed as a 0. Hard disks are the slowest type of memory. The hard disk can rotate at either 5400 RPM or 7200 RPM. They can typically only transfer data at a rate of 80 to 150 Mbps. However, in contrast, the speed at which a modern-day processor can take information is between one to four gigahertz.

Solid State Drives (SSD) are a sufficient improvement over the aforementioned hard disk. SSDs use a flash memory block that writes I/O operations to their pages while simultaneously erasing data as well. The speed at which this can be done is anywhere between 500 – 3,500 MB/s. However, due to the erase-before-writing feature, it is not cost-effective in performance but still a much better alternative to hard disks.

The Cache memory of a system is a tiny yet blazing fast component that allows the fastest access to data in a computer (see figure 3). The cache memory is located between the central processing unit and the RAM (see figure 4). Its primary purpose is to ameliorate the slow performance of other memory components and to ensure close to access time of the operations needed by the CPU.

## Cache Memory Hierarchy

Computers have a multitude of different memory systems in which all have their unique purposes. The uses of these systems are measured by their speed vs. cost (see figure 1). These memory systems are either volatile or non-volatile. Volatile systems lose their state (stored inputs) when the system is powered down, or the systems can be non-volatile, which means they can preserve their stored inputs. The cache memory system is volatile, losing its information to send to the processor once the system is shut down.

The cache memory system also has a type of functionality regarded as the cache controller. Cache controllers both helps the cache handle the input and sending the data over to the processor and other memory systems. When the CPU is ready and requests input from the cache, the controller then scans through each element in the cache and, if it is stored, sends the information to the processor. If the element is not available in the cache, then the controller will signal to the RAM. All read/write processes are also taken care of by the cache controller.

Cache memory stores information for the processor to retrieve and the information is searched through by the cache controller. If the cache controller retrieval is successful, then it is considered a cache hit, and if it is unsuccessful, then it is considered a cache miss. A cache hit is when the data element is requested and is present within the cache and successfully retrieved by the controller. A cache miss is when the data is requested from the cache, and it is not available. If the data is not available, then it is requested by memory, and then it is sent to the cache and passed onto the processor (Khan, 2020). When there is a cache missed, there is some propagation delay that occurs.

## Cache Memory Architecture

The cache memory is a hardware component that acts as a blazing fast buffer located between the processor and the main memory, also known as RAM. The main functionality of the cache memory is to reduce the overall time and energy it takes for the processor to fetch data. It is a small part of a greater RAM component (Hill, 1987). Cache memory is also a type of RAM, but instead of it being dynamic, it is static.

The cache is able to improve a system's cost performance by retrieving the data from the computer’s memory more efficiently and delivering it to the processor faster of that of the main memory. It, therefore, acts as a temporary storage by keeping key memory references without other memory systems getting involved. There are two critical functions of the cache memory. The first is the temporal locality, and the second is called spatial locality. Temporal locality predicts future references that have a high probability of being called (Sibai, 2008). The spatial locality aids the cache by capturing consecutive words. This helps when a miss occurs, then the CPU can access the cache and retrieve the word (Sibai, 2008).

## Cache Memory Levels

The cache memory component on a computer has multiple levels of architecture. These levels have their own unique purposes, along with different techniques on how they are designed. As you climb to different levels, both the access time and capacity of the cache increases, and in turn, the cache levels become increasingly slow. The closer the levels are to the processor, the faster the levels operate. Level 1 cache is much faster than level 2, and level 2 cache is much faster than level 3.

The first level cache (L1) is the closest to the processor, and each core within the CPU has its own L1 memory cache. First-level caches are placed near the processor. This allows the L1 cache to operate at the same speed as the CPU itself (Khan, 2020). The size of the L1 cache can range anywhere between 256 KB to 1 MB.

The second level cache (L2) as stated previously, is slower than that of L1, and is larger than it as well. The size of L2 can be anywhere up to 8MB (Khan, 2020). Its main function is to provide the processor with data that is predicted to be used by the processor. This level cache is isolated from the other cores.

The third level cache (L3) is the slowest of all the caches. It also has a greater size in comparison. The third level cache can be installed either on the motherboard, or inside the CPU. The L3 cache works with the L2 cache by sending it the information it needs, which then L2 cache relates to the L1 cache, to which it sends to the processor.

# Conclusion

Through the research that has been concluded in the paper, the cache memory system is the fastest memory system available within the computers architecture. This is because of the engineering architectural design of the cache having multiple levels and being placed so closely, or even sometimes embedded in the CPU itself. Contrasting cache memory to any other memory system demonstrates the need for more innovation in the area to grow parallel with processor innovation. This is especially important as the direction of processor architecture is heading into the multicore realm. More research efforts should be placed in level sharing within the cache memory design to keep up with the growing innovations of processors.

References

Bae, H. J., & Choi, L. (2020). Filter cache: Filtering useless cache blocks for a small but efficient shared last-level cache. *The Journal of Supercomputing*, *76*(10), 7521–7544. https://doi.org/10.1007/s11227-020-03177-2

Choi, G. S., & On, B.-W. (2011). Study of the performance impact of a cache buffer in solid-state disks. *Microprocessors and Microsystems*, *35*(3), 359–369. <https://doi.org/10.1016/j.micpro.2010.09.010>

Hill, M. D. (1987). (tech.). *Aspects of Cache Memory and Instruction Buffer Performance* (pp. 1–185).

Khan, Ameer. (2020). Brief Overview of Cache Memory. 10.13140/RG.2.2.22359.21921.

Sardashti, S., Seznec, A., & Wood, D. A. (2016). Yet another compressed cache. *ACM Transactions on Architecture and Code Optimization*, *13*(3), 1–25. https://doi.org/10.1145/2976740

Sibai, F. N. (n.d.). On the performance benefits of sharing and privatizing second and third-level cache memories in homogeneous multi-core architectures. *Microprocessors and Microsystems*, *32*(7), 405–412.

Xie, Y. (2011). Modeling, architecture, and applications for emerging memory technologies. *IEEE Design & Test of Computers*, *28*(1), 44–51. https://doi.org/10.1109/mdt.2011.20

Table of Figures

A picture containing graphical user interface

Description automatically generated

Figure Speed vs. Cost

A picture containing hard disc

Description automatically generated

Figure Hard Disk



Figure Cache Memory

Diagram

Description automatically generated

Figure Cache Location