University of Babylon, College of science for women
Dept. of Computer science

Computer Architecture

Second year

Dr. Salah Al-Obaidi

Lecture #11: Memory technology and caches Spring 2024

Contents

12. Memory technology and caches

12.1 Memory Hierarchy Design

The importance of the memory hierarchy has increased with advances in performance of processors. More recently, high-end processors have moved to multiple cores, further increasing the bandwidth requirements versus single cores. In fact, the aggregate peak bandwidth essentially grows as the numbers of cores grows. A modern high-end processor such as the Intel Core i7 can generate two data memory references per core each clock cycle; with four cores and a 3.2 GHz clock rate, the i7 can generate a peak of 25.6 billion 64-bit data memory references per second, in addition to a peak instruction demand of about 12.8 billion 128-bit instruction references; this is a total peak bandwidth of 409.6 GB/sec!. This incredible bandwidth is achieved by multiporting and pipelining the caches; by the use of multiple levels of caches, using separate first- and sometimes second-level caches per core; and by using a separate instruction and data cache at the first level. In contrast, the peak bandwidth to DRAM main memory is only 6% of this (25 GB/sec).

12.2 Characteristics of Memory Systems

The complex subject of computer memory is made more manageable if we classify memory systems according to their key characteristics. The most important of these are listed in Figure 12.1.

Location	**Performance**
Internal (e.g., processor registers, cache, main memory)	Access time
External (e.g., optical disks, magnetic disks, tapes)	Cycle time
Capacity	Transfer rate
Number of words	**Physical Type**
Number of bytes	Semiconductor
Unit of Transfer	Magnetic
Word	Optical
Block	Magneto-optical
Access Method	**Physical Characteristics**
Sequential	Volatile/nonvolatile
Direct	Erasable/nonerasable
Random	**Organization**
Associative	Memory modules

Figure 12.1: Key Characteristics of Computer Memory Systems.

The term **location** in Figure 12.1 refers to whether memory is internal or external to the computer. The types of internal memory are:

- Internal memory is often equated with main memory.
- The processor requires its own local memory, in the form of registers.
- Cache is another form of internal memory.

External memory consists of peripheral storage devices, such as *disk* and *tape*.

An obvious characteristic of memory is its **capacity**. For internal memory, this is typically expressed in terms of *bytes* (1 byte = 8 bits) or *words*. Common word lengths are 8, 16, and 32 bits. External memory capacity is typically expressed in terms of *bytes*.

A related concept is the **unit of transfer**. For internal memory, the unit of transfer is equal to *the number of electrical lines into and out of the memory module*. This may be equal to the word length, but is often larger, such as 64, 128, or 256 bits.

From a user's point of view, the two most important characteristics of memory are **capacity** and **performance**. Three performance parameters are used:

- **Access time (latency)**: For random-access memory, this is *the time it takes to perform a read or write operation*, that is, the time from the instant that an address is presented to the memory to the instant that data have been stored or made available for use. For non-random-access memory, access time is *the time it takes to position the read–write mechanism at the desired location.*

- **Memory cycle time**: This concept is primarily applied to random-access memory and consists of *the access time plus any additional time required before a second access can commence.* Note that memory cycle time is concerned with the system bus, not the processor.

- **Transfer rate**: This is *the rate at which data can be transferred into or out of a memory unit.* For random-access memory, it is equal to **1/(cycle time)**. For non-random-access memory, the following relationship holds:

$$T_n = T_A + \frac{n}{R}, \tag{12.1}$$

where

T_n = Average time to read or write n bits

T_A = Average access time

n = Number of bits

R = Transfer rate, in bits per second (bps)

12.3 The Memory Hierarchy

The design constraints on a computer's memory can be summed up by three questions: How much? How fast? How expensive?.

The question of how much is somewhat open ended. If the capacity is there, applications will likely be developed to use it. The question of how fast is, in a sense, easier to answer. To achieve greatest performance, the memory must be able to keep up with the processor. That is, as the processor is executing instructions, we would not want it to have to

pause waiting for instructions or operands. The final question must also be considered. For a practical system, the cost of memory must be reasonable in relationship to other components.

As might be expected, there is a trade-off among the three key characteristics of memory: capacity, access time, and cost. A variety of technologies are used to implement memory systems, and across this spectrum of technologies, the following relationships hold:

- Faster access time, greater cost per bit;
- Greater capacity, smaller cost per bit;
- Greater capacity, slower access time.

The designer would like to use memory technologies that provide for large-capacity memory, both because the capacity is needed and because the cost per bit is low. However, to meet performance requirements, the designer needs to use expensive, relatively lower-capacity memories with short access times.

The way out of this dilemma is not to rely on a single memory component or technology, but to employ a memory hierarchy. A typical hierarchy is illustrated in Figure 12.2.

As one goes down the hierarchy, the following occur:

1. Decreasing cost per bit;
2. Increasing capacity;
3. Increasing access time;
4. Decreasing frequency of access of the memory by the processor.

Thus, smaller, more expensive, faster memories are supplemented by larger, cheaper, slower memories.

The fastest, smallest, and most expensive type of memory consists of the registers internal to the processor. Typically, a processor will contain a few dozen such registers,

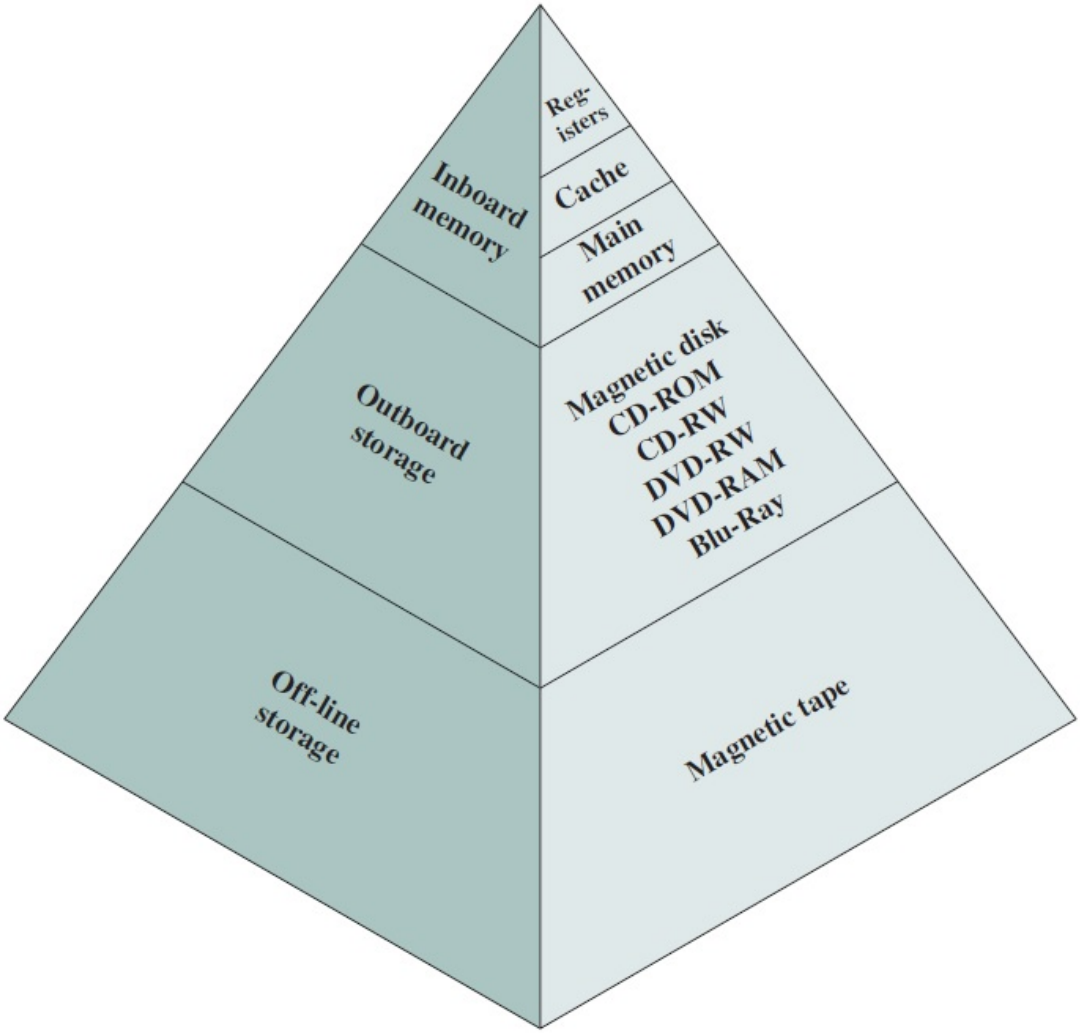
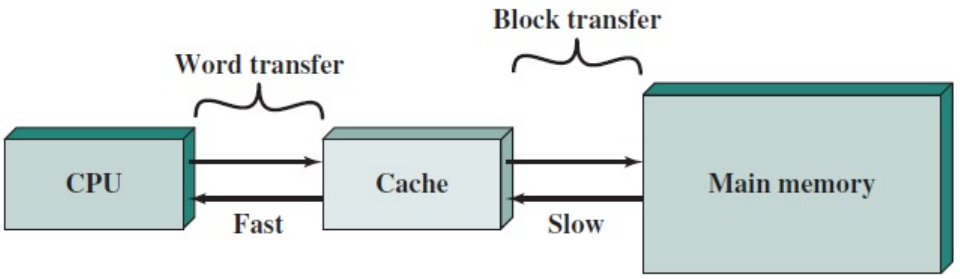


Figure 12.2: The Memory Hierarchy.

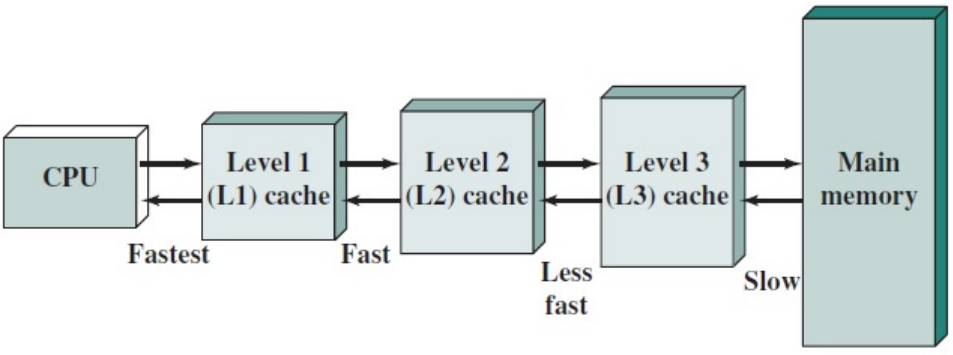
although some machines contain hundreds of registers. Main memory is the principal internal memory system of the computer. Each location in main memory has a unique address. Main memory is usually extended with a higher-speed, smaller cache. The cache is not usually visible to the programmer or, indeed, to the processor. It is a device for staging the movement of data between main memory and processor registers to improve performance.

12.4 CACHE MEMORY PRINCIPLES

Cache memory is designed to combine the memory access time of expensive, high- speed memory combined with the large memory size of less expensive, lower- speed memory. The concept is illustrated in Figure 12.3a. There is a relatively large and slow main memory together with a smaller, faster cache memory. The cache contains a copy of portions of main memory. When the processor attempts to read a word of memory, a check is made to determine if the word is in the cache. If so, the word is delivered to the processor. If not, a block of main memory is read into the cache and then the word is delivered to the processor. Because of the phenomenon of locality of reference, when a block of data is fetched into the cache to satisfy a single memory reference, it is likely that there will be future references to that same memory location or to other words in the block.



(a) Single cache



(b) Three-level cache organization

Figure 12.3: Cache and Main Memory

Figure 12.3b depicts the use of multiple levels of cache. The **L2** cache is slower and typically larger than the **L1** cache, and the **L3** cache is slower and typically larger than the **L2** cache.

Figure 12.4 depicts the structure of a cache/main-memory system. Main memory consists of up to 2^n addressable words, with each word having a unique $n - bit$ address. For mapping purposes, this memory is considered to consist of a number of fixed-length blocks of K words each. That is, there are $M = 2^n/K$ blocks in main memory. The cache consists of m blocks, called lines. Each line contains K words, plus a tag of a few bits. The length of a line, not including tag and control bits, is the line size. The line size may be as small as 32 bits, with each "word" being a single byte; in this case the line size is 4 bytes. The number of lines is considerably less than the number of main memory blocks $(m \ll M)$. At any time, some subset of the blocks of memory resides in lines in the cache. If a word in a block of memory is read, that block is transferred to one of the lines of the cache. Because there are more blocks than lines, an individual line cannot be uniquely and permanently dedicated to a particular block. Thus, each line includes a tag that identifies which particular block is currently being stored. The tag is usually a portion of the main memory address.

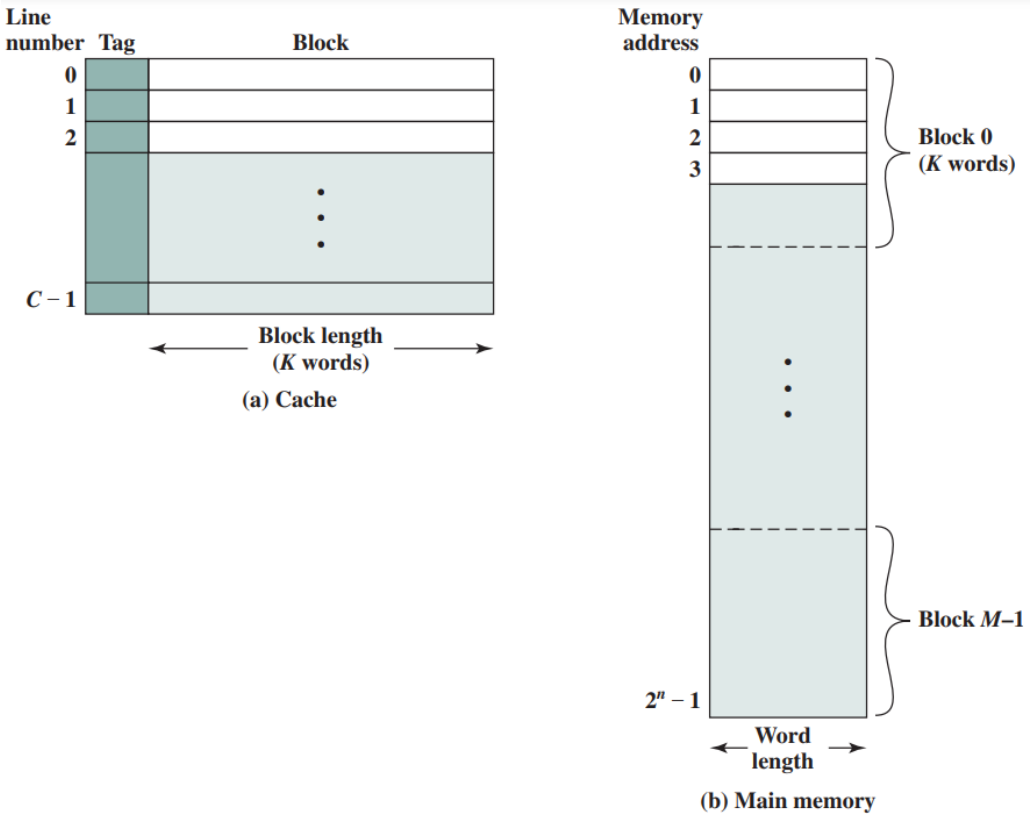


Figure 12.4: Cache/Main Memory Structure

Figure 12.5 illustrates the read operation. The processor generates the read address (RA) of a word to be read. If the word is contained in the cache, it is delivered to the processor. Otherwise, the block containing that word is loaded into the cache, and the word is delivered to the processor. The cache connects to the processor via data, control, and address lines. The data and address lines also attach to data and address buffers, which attach to a system bus from which main memory is reached. When a cache hit occurs, the data and address buffers are disabled and communication is only between processor and cache, with no system bus traffic. When a cache miss occurs, the desired address is loaded onto the system bus and the data are returned through the data buffer to both the cache and the processor.

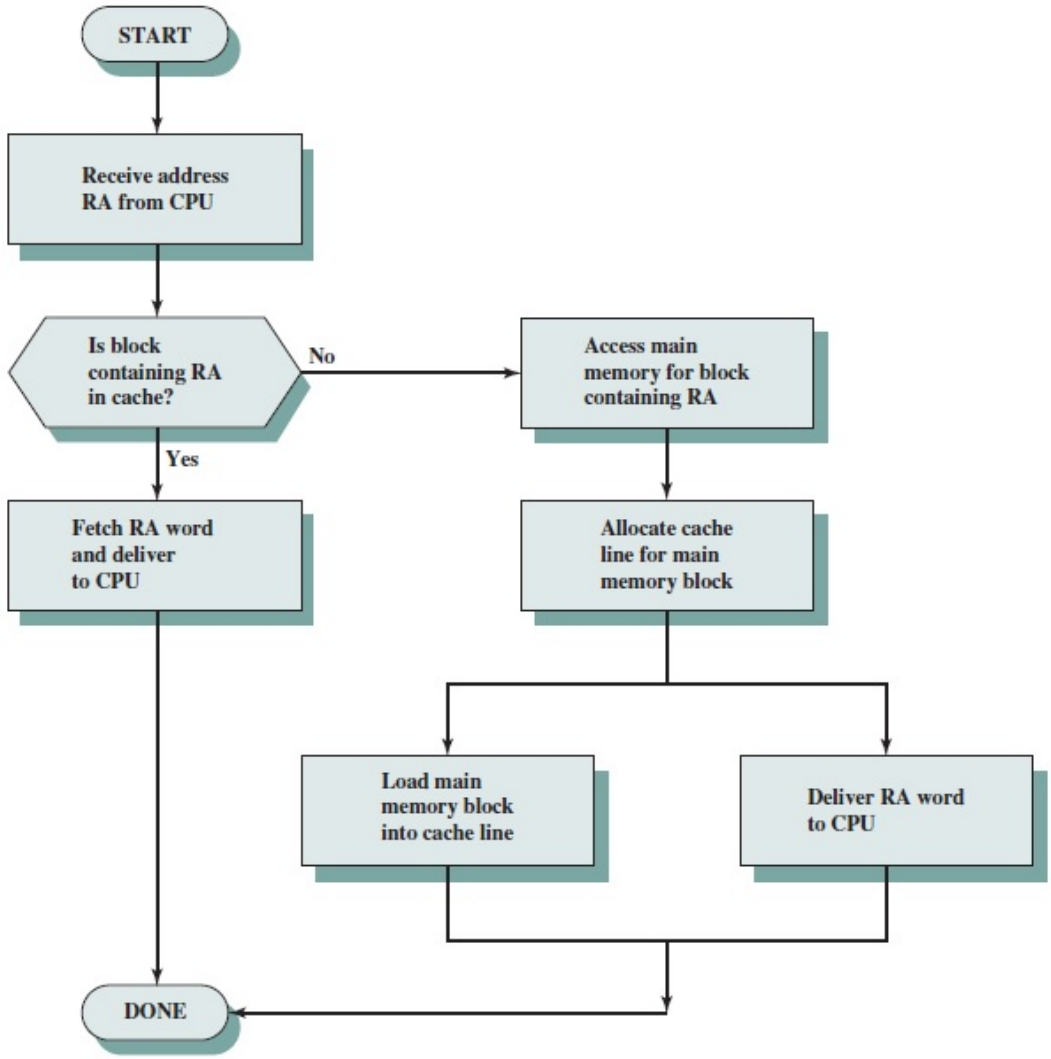


Figure 12.5: Cache Read Operation