Chapter No. 04 – Cache Memory (Part – 01)

Lecture – 08

12-11-2018

Topics to Cover

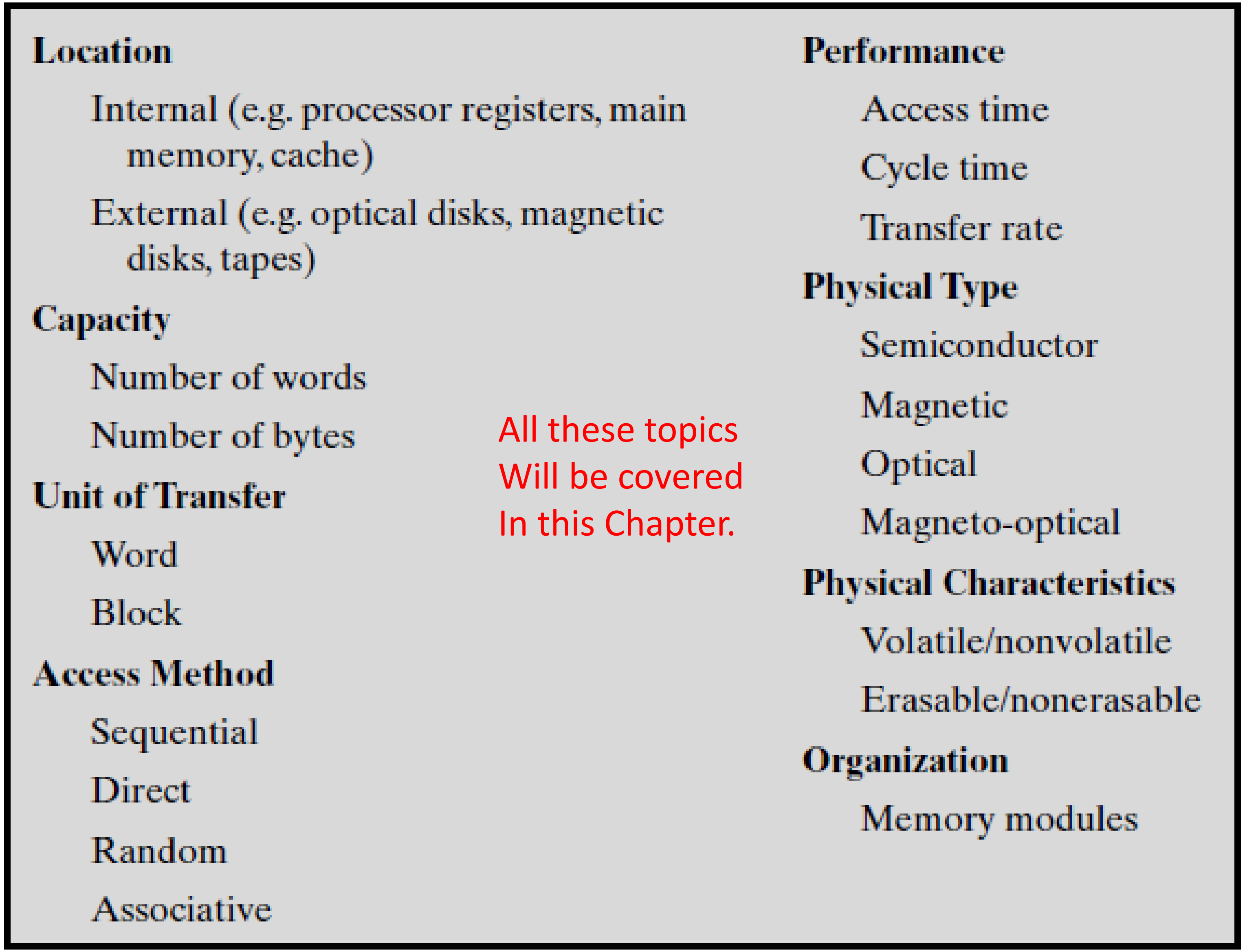
- 4.1 – Computer Memory System Overview
- Characteristics of Memory Systems
- The Memory Hierarchy
- 4.2 – Cache Memory Principles
- 4.3 – Elements of Cache Design

Cache Addresses	Cache Size
Mapping Function	Replacement Algorithms
Write Policy – Line Size	Number of Caches

Introduction

- **Computer memory** exhibits perhaps the widest range of type, technology, organization, performance and cost of any feature of a computer system.
- To satisfy the memory requirements of a computer system, the typical system is equipped with a hierarchy of memory subsystems.
- Because conventional (main) memory (RAM) is so much slower than the CPU, microcomputers have high-speed **cache memory** that holds the most recently used instructions and data.
- Whenever possible, the CPU reads from cache memory, giving programs a noticeable boost in performance.

Table 4.1 Key Characteristics of Computer Memory Systems



Location

- Internal (e.g. processor registers, main memory, cache)
- External (e.g. optical disks, magnetic disks, tapes)

Capacity

- Number of words
- Number of bytes

Unit of Transfer

- Word
- Block

Access Method

- Sequential
- Direct
- Random
- Associative

Performance

- Access time
- Cycle time
- Transfer rate

Physical Type

- Semiconductor
- Magnetic
- Optical
- Magneto-optical

Physical Characteristics

- Volatile/nonvolatile
- Erasable/nonerasable

Organization

- Memory modules

All these topics
Will be covered
In this Chapter.

1. Memory ‘Location’

- **Location** refers to whether memory is internal and external to the computer.
- ‘Memory location’ can be of two types: 1) Internal 2) External

1. **Internal memory** is located inside the computer and is often equated with main memory (RAM). This memory refers to **chips/IC.**

- ‘Cache’ is another form of internal memory.

2. **External memory** consists of peripheral storage devices, such as Hard disk, that are accessible to the processor via I/O Controllers.

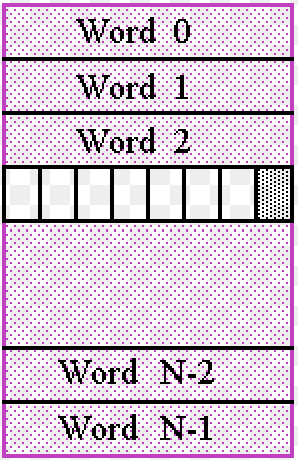
- ‘External memory’ is sometimes called ‘secondary’ and is used for permanent storage of large quantities of data on ‘magnetic disks’.

2. Memory ‘Capacity’

- For internal memory, this is typically expressed in terms of **bytes** (1 byte = 8 bits) or **words** e.g. Kilo-bytes, mega-bytes, giga-bytes.
- Common word lengths are 8, 16, 32 and 64 bits.
- External memory capacity is typically expressed in terms of **bytes** e.g. hundreds of gia-bytes.

- Note: The smallest addressable unit is ‘byte’.

What is a ‘Word’ in memory?



- In memory, a **word** is the natural unit of data used by a particular processor design. A word is a fixed-sized piece of data handled as a unit by the instruction set or the hardware of the processor.
- The number of bits in a word (the ***word size***, ***word width***, or ***word length***) is an important characteristic of any specific processor design or computer architecture.
- The majority of the registers in a processor are usually word sized and the largest piece of data that can be transferred to and from the working memory in a single operation is a word.
- Modern general purpose computers/CPUs, Word is a 32 or 64 bits.

3. Memory ‘Unit of Transfer’

- **Addressable units:** In some systems, the addressable unit is the word. However, many systems allow addressing at the byte level.
- The relationship between the length in bits A of an address and the number N of addressable units is **2^A = N**. e.g. 2^4 = 16 locations/words.
- For internal memory, the ‘unit of transfer’ is equal to the number of bytes read out of or written into memory at a time.
- This may be equal to the word length.
- For external memory, data are often transferred in much larger units than a word, and these are referred to as ‘blocks’. (multiples of word)

4. Memory ‘Method of Accessing’

- **Method of accessing** units of data. These include the following:

1) Sequential access 2) Direct access 3) Random access 4) Associative

1. **Sequential access:** the access must be made in a specific linear sequence, passing and rejecting each intermediate units of data.

- The time to access data is variably slow e.g. reading a ‘tape unit’.

2. **Direct access:** The access is accomplished by block’s address and to directly reach a general vicinity plus sequential searching to reach the final location.

- Again, the data access time is variable, the examples are ‘disk units’.

Method of Accessing (Continued)

3. **Random access:** Any location can be selected at random using its unique address, and can be directly addressed and accessed.

- The data access time is constant independent of its location. The example is ‘main memory’ (RAM).

4. **Associative:** It’s a random access type of memory that enables one to make a comparison of desired bit locations within a word for a specified match, for all words simultaneously.

- Thus, a word is retrieved based on a portion of its contents rather than address.
- ‘Cache memory’ may employ associative access.

5. Memory ‘Performance’

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

1) Access time (latency) 2) Memory cycle time 3) Transfer rate

1. **Access time (latency):** For RAM, this is the time it takes to perform a read or write operation, once address is presented to the memory.

- For non-RAM memory, it is the time it takes to position the read-write mechanism at the desired location.

Performance (Continued)

2. **Memory cycle time:** Consists of the ‘access time (latency) plus (+) any additional time’ required before a second access can commence.

- This additional time may require for transients to die out on the ‘system bus’.

3. **Transfer rate:** This is the rate at which data can be transferred into or out of a memory unit.

- For RAM, it is equal to 1/(cycle time). (more cycle-time less transfer)
- For non-RAM memory, the Average access time to read or write n bits is equal to the ‘average access time + (no. of bits/Transfer rate(bps))’.

6. ‘Physical Types’ of Memory

- **Physical types** of memory are:

1) Semiconductor memory (in the form of chips e.g. RAM)

2) Magnetic surface memory (used for Hard-disk)

3) Optical memory (as CDs)

7. ‘Physical Characteristics’ of Memory

- **Physical characteristics** of data storage are:

1) Volatile/non-Volatile 2) Erasable/non-Erasable

1. In a **volatile memory** e.g. RAM, information decays naturally or is lost when electrical power is switched off.
2. In a **non-volatile memory** e.g. Hard-Disk information once recorded remains without deterioration until deliberately changed; no electrical power is needed to retain information.
3. **Non-erasable memory** can not be altered. Semiconductor memory of this type is known as read-only memory (ROM).

- A practical ‘non-erasable memory’ must also be ‘non-volatile’.

(Break)

Key Characteristics of Memory

- The three key-characteristics of memory are: (dependent on each other)

1) Capacity 2) Access time 3) Cost

- To achieve greatest performance, the memory must be able to keep up with the processor demands.
- That is, as the processor is executing instructions, the memory must not make the processor to pause and wait for instructions or operands.
- Increasing ‘memory capacity’, decreases cost per bit for storage.
- Lower capacity memories have short ‘access time’.

Trade-off Among ‘Memory Characteristics’

➢In memory systems, the following relationship holds:

- Faster access time, greater cost per bit. (E.g. cache)
- Greater capacity, smaller cost per bit. (E.g. RAM)
- Greater capacity, slower access time. (E.g. RAM)

- The way out of this trade-off is not to rely on a single memory component or technology, but to employ a **memory hierarchy.**

Memory Hierarchy

- As one goes down the hierarchy, the following occur:

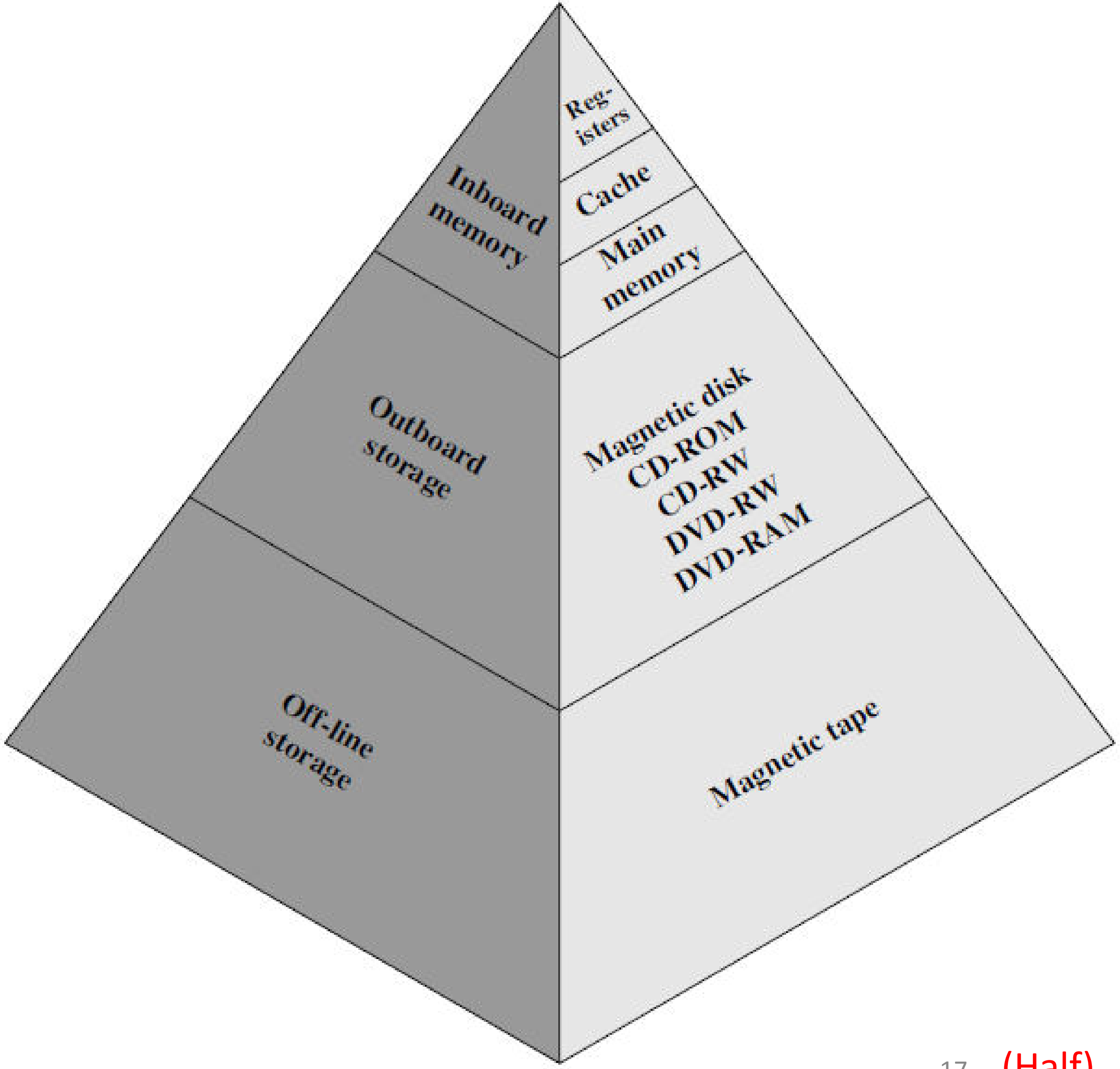
a. Decreasing cost per bit

b. Increasing capacity

c. Increasing access time

d. Decreasing frequency of access of the memory by the processor.

Fastest



Slowest

(Half)

Principal of ‘Locality of Reference’

- During the course of execution of a program, memory references by the processor, for both instructions and data, tend to cluster.
- ‘Programs’ typically contain a number of iterative loops and subroutines.
- Once a loop or subroutine is entered, there are repeated references to a small set of instructions.
- Similarly, operations on tables and arrays involve access to a ‘clustered set of data words’.
- Over a short period of time, the processor is primarily working with fixed clusters of memory references called **locality of reference.**
- This **locality** is fetched to decrease frequency of memory access by CPU

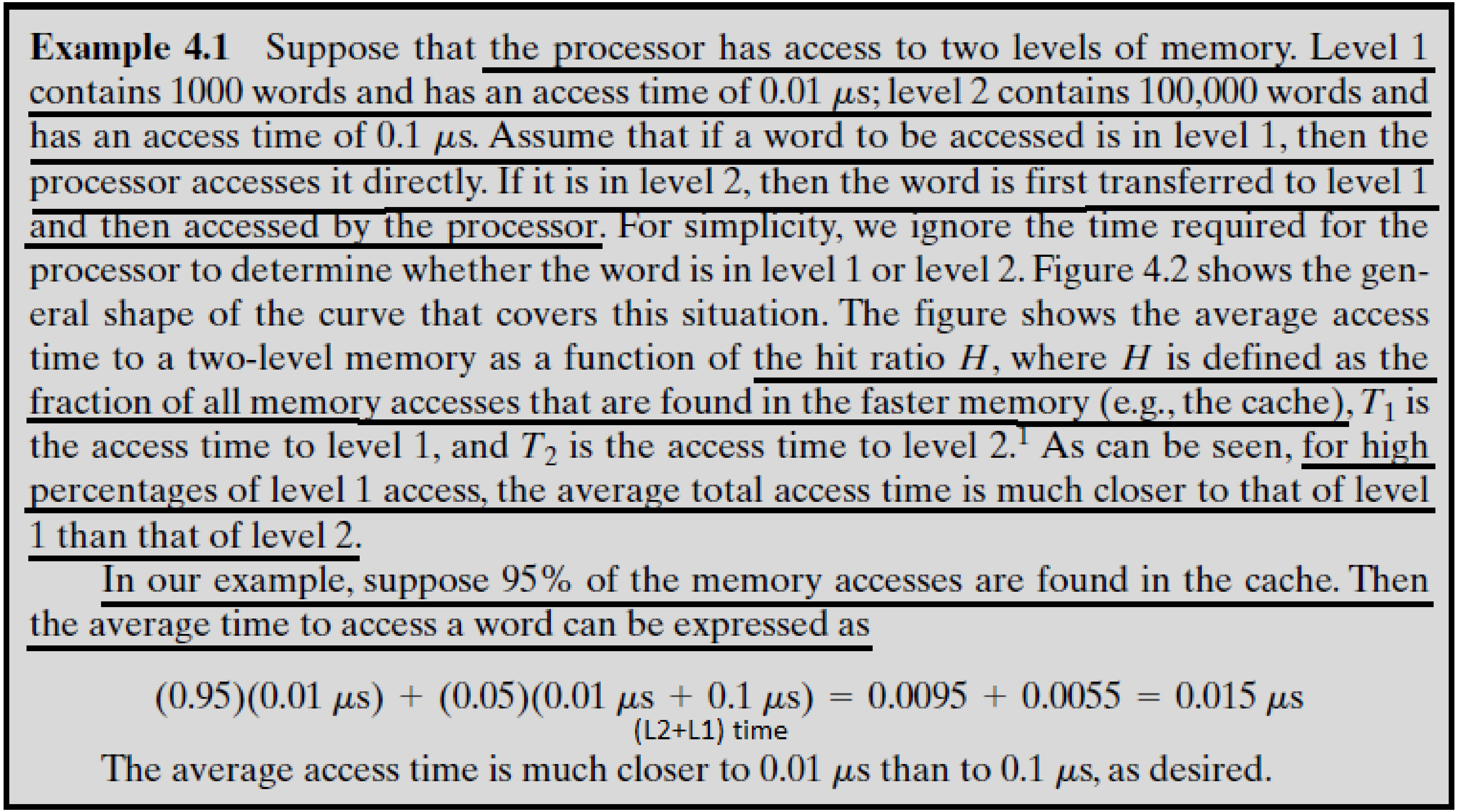
The Two Levels of Cache Memory

- **<u>Cache</u>** is a smaller, faster memory used by the CPU of a computer to reduce the average time to access data from the main memory.
- The use of two levels of memory to reduce average access time works.
- There are two type of cache memory in a system:
- **Level-1 (L1) cache:** is inside the processor itself, and is often accessed in one cycle by the processor. It is also called ‘on-chip cache’.
- **Level-2 (L2) cache:** is located on separate high-speed memory chips next to the CPU, often built with SRAM IC’s. It’s called ‘off-chip cache’.
- Level-1 cache is faster and more expensive than Level-2 cache.

A **Hit** is counted if the desired data appears in either the L1 or the L2 cache.

Example 4.1

Note: μ is 10^{-6}



Example 4.1 Suppose that the processor has access to two levels of memory. Level 1 contains 1000 words and has an access time of 0.01 μs; level 2 contains 100,000 words and has an access time of 0.1 μs. Assume that if a word to be accessed is in level 1, then the processor accesses it directly. If it is in level 2, then the word is first transferred to level 1 and then accessed by the processor. For simplicity, we ignore the time required for the processor to determine whether the word is in level 1 or level 2. Figure 4.2 shows the general shape of the curve that covers this situation. The figure shows the average access time to a two-level memory as a function of the hit ratio *H*, where *H* is defined as the fraction of all memory accesses that are found in the faster memory (e.g., the cache), T_1 is the access time to level 1, and T_2 is the access time to level 2.[1] As can be seen, for high percentages of level 1 access, the average total access time is much closer to that of level 1 than that of level 2.

In our example, suppose 95% of the memory accesses are found in the cache. Then the average time to access a word can be expressed as

$$(0.95)(0.01\ \mu s) + (0.05)(0.01\ \mu s + 0.1\ \mu s) = 0.0095 + 0.0055 = 0.015\ \mu s$$

(L2+L1) time

The average access time is much closer to 0.01 μs than to 0.1 μs, as desired.

Tip to Improve Data Access Time from Memory

- Organize data across the hierarchy, such that the percentage of access to each slower level memory is substantially less that that of the level above.
- E.g. let level-2 memory contains all program instructions and data.
- The current clusters can be temporarily placed in level-1.
- From time to time, one of the clusters in level-1 will have to be swapped back to level-2 to make room for a new cluster coming in to level-1.
- On average, however, most references will be to instructions and data contained in level-1 memory.

General Purpose Registers (GPRs)

- The fastest, smallest, and most expensive type of memory consists of the registers internal to the processor called GPRs.
- A modern processor has typically 32 integer registers and 32 floating point registers. They are also called register file.

- Main memory is the internal memory, and each location in main memory has a unique address.
- Main memory is usually extended with a higher-speed, smaller cache.
- Cache is a device for staging the movement of data between main memory and processor register to improve performance.
- All these memories are volatile, and use semiconductor technology.
- The cache is not visible to programmer, memories differ in speeds.

Secondary Memory

- Data are stored more permanently on external mass storage devices, of which the most common are hard disk and removable media.
- External, non-volatile memory is also referred to as **secondary memory** or **auxiliary memory.**
- These are used to store program and data files, and are usually visible to the programmer only in terms of files or records, as opposed to individual bytes or words.
- Disk is also used to provide an extension to main memory known as **virtual memory.** (Will be taught later)
- Other forms of secondary memory include optical disks.

4.2 Cache Memory Principles

- 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, consisting of some fixed number of words (k-words), is read into the cache and then the word is delivered to the processor. (data delivered in blocks due to locality of reference)
- The cache connects to the processor via data, control & address lines.

Fig. 4.3(a) Single Cache

- Because of the 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 memory. This is called Temporal locality.

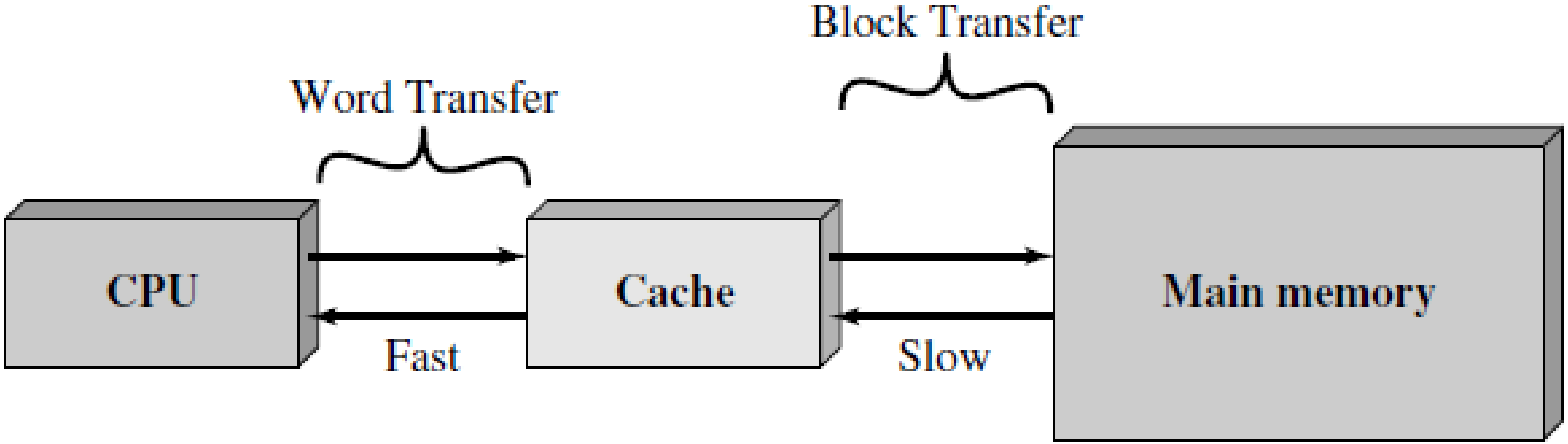
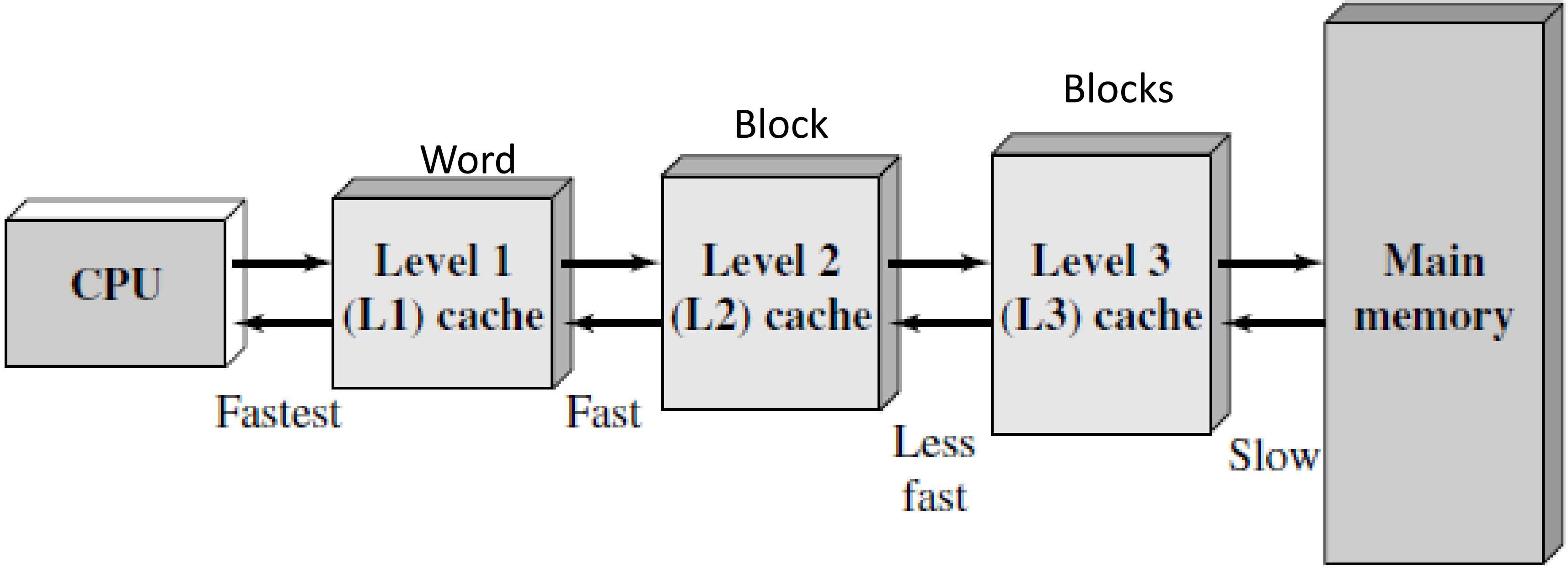


Fig. 4.3(b) Three-level Cache Organization

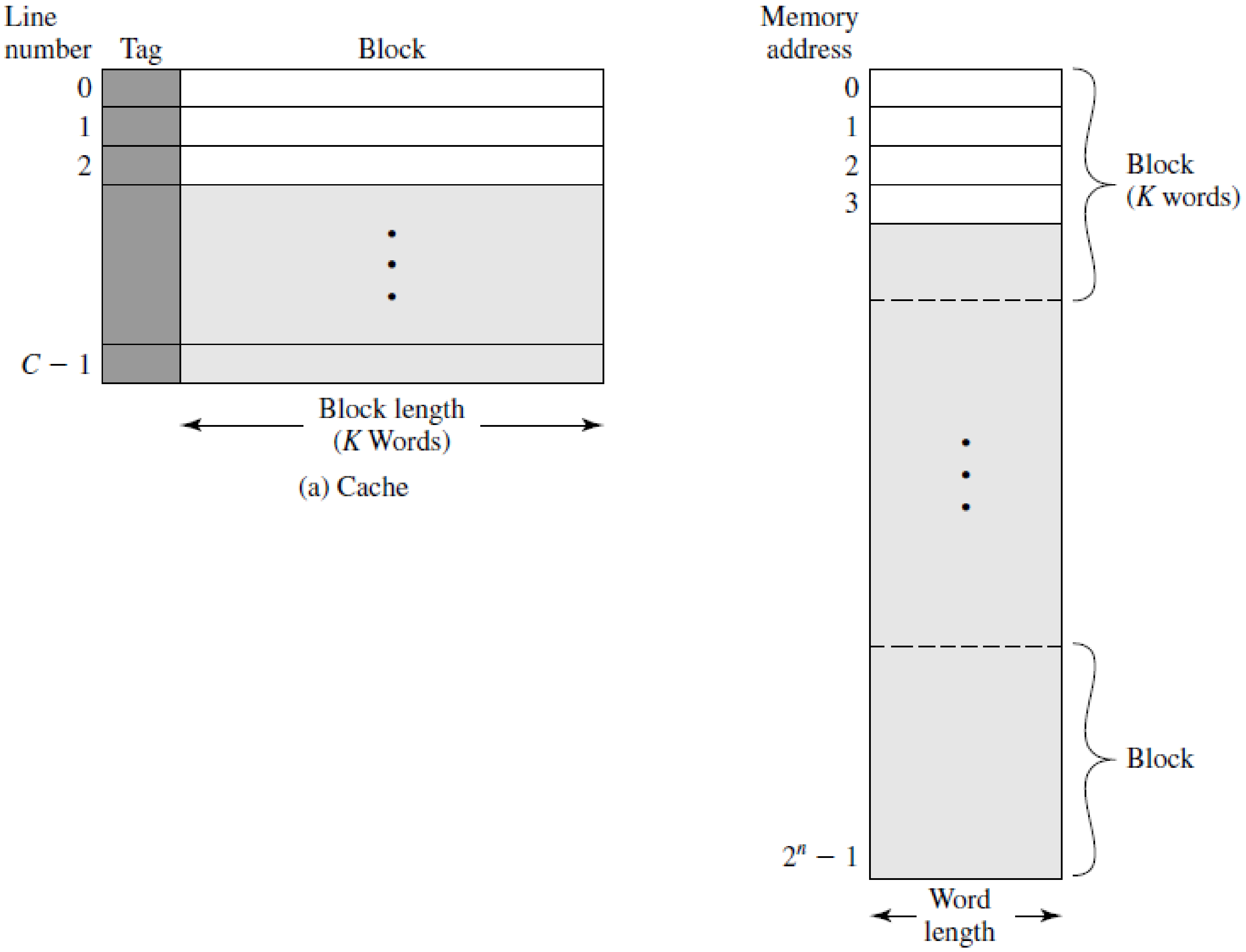
- When we use multiple levels of cache, L1, L2 and L3.
- The L2-cache is slower and typically larger than the L1-cache.
- And the L3-cache is slower and typically larger than the L2-cache.



Cache/Main Memory Mapping (Figure Next)

- 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. (e.g. 1000/4 =250)
- The cache consists of **m**-blocks, called **lines.** (block resides in line)
- Each line contains K-words & a **tag** which identifies a particular block.
- The length of a line, not including tag, 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/words.

Fig. 4.4 Cache/Main Memory Structure



(a) Cache

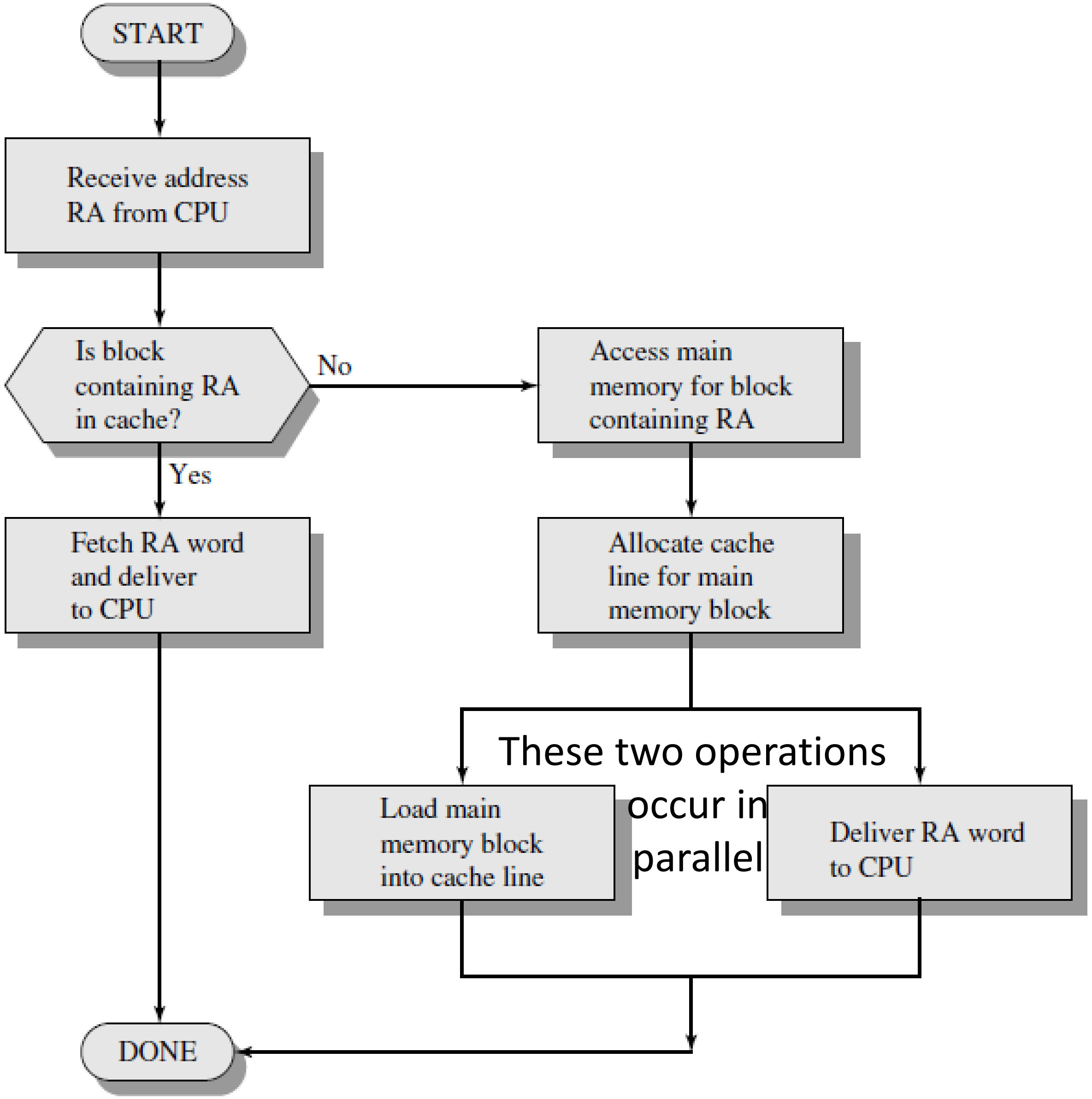
(b) Main memory

Cache Lines VS Memory Blocks

- The cache lines are much less than main memory blocks (m<<M).
- So an individual line can not be permanently dedicated to a block.
- At any time, some subset of the blocks of memory resides in the cache.
- Which block is in a line is identified by a tag. Which is a portion of the main memory address.
- If a word in a block of memory is read, that block is transferred to one of the lines of the cache.

Cache Read Operation

1. The processor generates the read address (RA) of a word to be read.
2. If the word is contained in the cache, it is delivered to the processor. (fast access)
3. Otherwise, the block containing that word is loaded into the cache, and the word is delivered to the processor.


START
Receive address RA from CPU
Is block containing RA in cache?
No
Yes
Access main memory for block containing RA
Fetch RA word and deliver to CPU
Allocate cache line for main memory block
Load main memory block into cache line
Deliver RA word to CPU
DONE

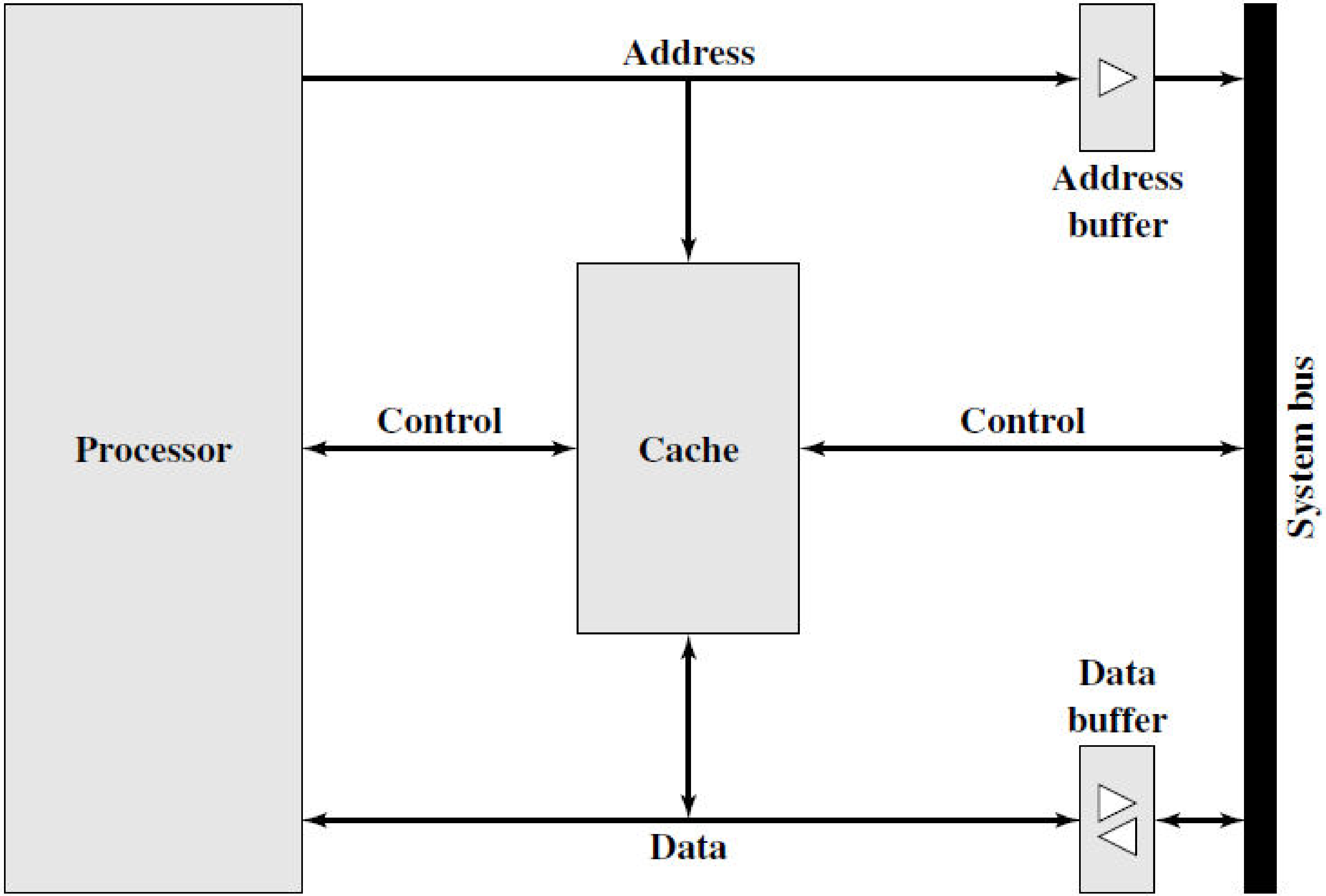
These two operations occur in parallel

Cache HIT and Cache MISS (Figure Next)

A **Hit** is counted if the desired data appears in either the L1 or the L2 cache.

- The cache connects to the processor via data, control & 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 **cache hit** occurs, the data and address buffers are disabled and communications is only between processor and cache, with **NO** system bus traffic.
- When a **cache miss** occurs, the desired address are loaded onto the system bus and the word is returned to both the cache & processor.
- The cache is physically interposed between the processor and main memory for all data, address, and control lines.

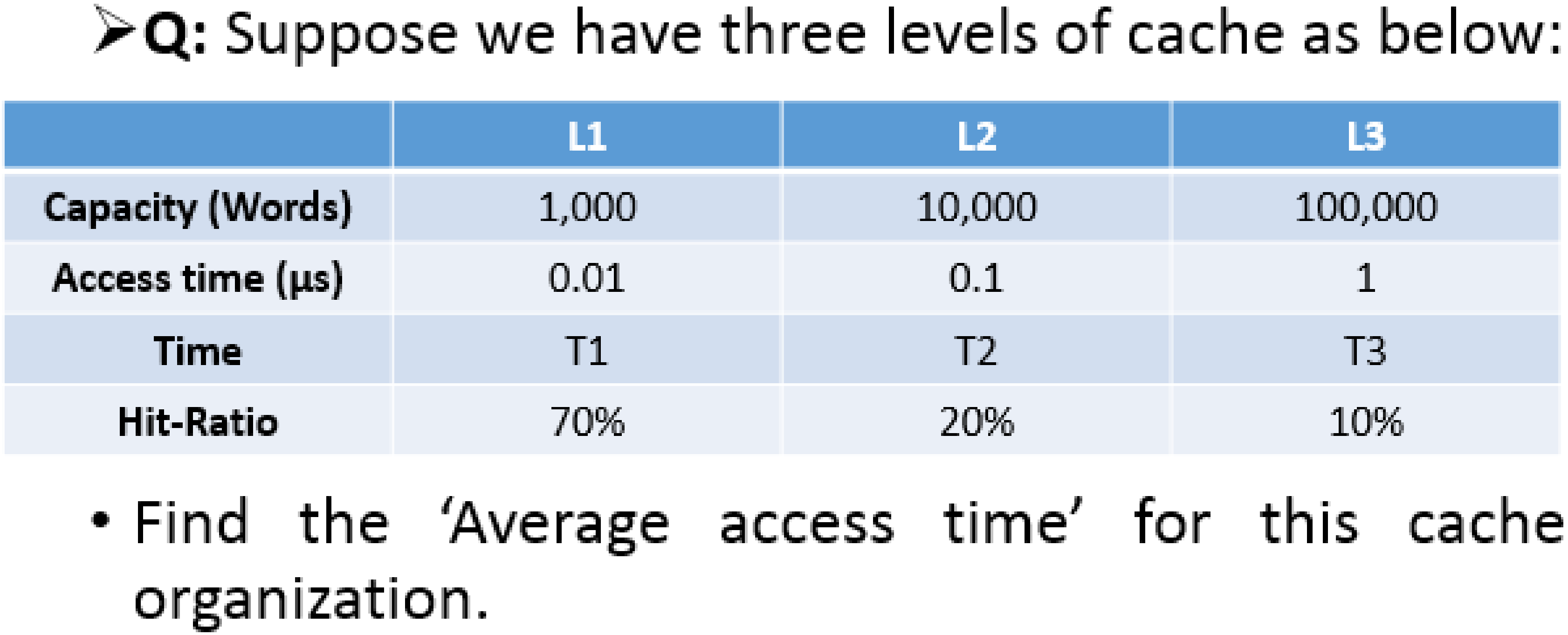
Fig. 4.6 Typical Cache Organization



Numerical Problem (Cache)

Que. If a direct mapped L1-cache has a hit-rate of 93%, a hit-time of 3ns. If an L2-cache is added with a hit-time of 25ns, a hit-rate of 65% and a miss penalty of 100ns, average memory access time is:

Answer: ?



➤**Q:** Suppose we have three levels of cache as below:

	L1	L2	L3
Capacity (Words)	1,000	10,000	100,000
Access time (µs)	0.01	0.1	1
Time	T1	T2	T3
Hit-Ratio	70%	20%	10%

- Find the 'Average access time' for this cache organization.

Preparatory Questions

Q1. What is the difference among 'direct access, random access and associative access'? (Slide – 09 and 10)

Q2. What are the different parameters that affect the 'performance' of a computer memory? (Slide – 11 and 12)

Q3. What are the 'physical characteristics' of memory? (Slide – 14)

Q4. What is the general relationship among 'access time, memory cost and capacity'? (Slide – 16)

Q5. Why is the 'principal of locality' used in the design of cache memory? (Slide – 18)

Q6. What is a 'cache' memory? What are the two-levels of cache memory? (Slide – 19)

Q7. What are the steps of a 'cache Read operation'? (Slide – 30)

Q8. What is a 'cache hit' and a 'cache miss'? What happens when a cache-miss occurs? (Slide – 31)