Cache Memory

ABDUL HASEEB

Objectives

After studying this chapter, you should be able to:

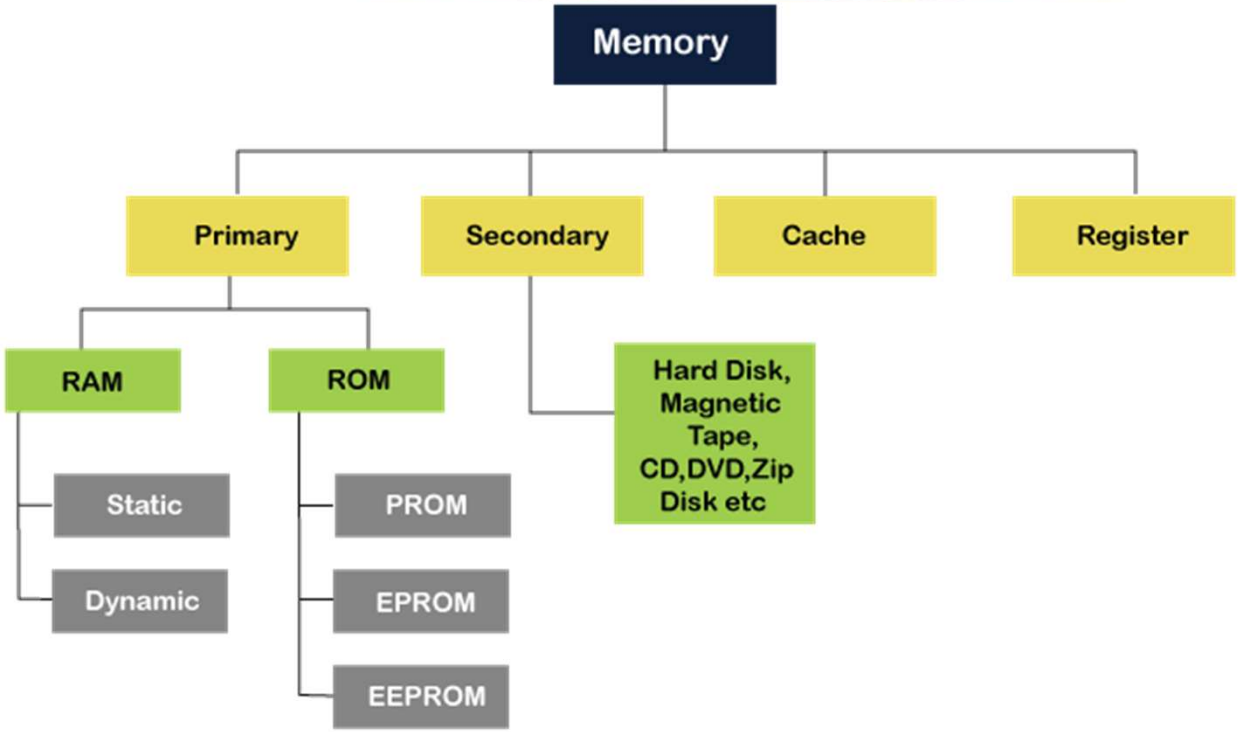
- Present an overview of the main characteristics of computer memory systems and the use of a memory hierarchy.
- Describe the basic concepts and intent of cache memory.
- Discuss the key elements of cache design.
- Distinguish among direct mapping, associative mapping, and set-associative mapping.
- Explain the reasons for using multiple levels of cache. Understand the performance implications of multiple levels of memory.

4.1 Computer Memory System Overview

What is memory!

- Electronic holding/storage space:
  - To store information (data and instructions) for immediate use in computers.

Classification of Memory



Characteristics of Computer Memory

- **Location:**
  - Whether memory is external or internal
  - **Internal:**
    - Local to the processor
    - Normally concerned with RAM
    - But there are other types too
  - **External:**
    - Peripheral storage Devices, like?

Characteristics of Computer Memory

- **Capacity:**
  - Amount of data a memory can hold
  - Bit is the basic unit (0 or 1)
  - Byte is the smallest possible unit for storage (byte)
  - Internal memory Capacity:
    - expressed in bytes (8 bits) or words which are normally 8,16 and 32 bits
  - External memory capacity is expressed in bytes

Characteristics of Computer Memory

- **Unit of Transfer:**
  - Number of electrical lines in and out of the memory
  - Normally it is equal to length of a word, but can be often larger than a word such as 64,128, 256:
    - Data can be transferred in block of words

Characteristics of Computer Memory

- **Methods of Access:**
  - Sequential
  - Random
  - Direct

Characteristics of Computer Memory

- **Sequential Access:**
  - When ever an access request is made:
    - Memory is searched linearly in sequential fashion
    - Search is started from the first memory location
    - Incremented by moving one step ahead at a time, until desired location is found
    - Simple but slow technique

A
B
C
D
F
G
H

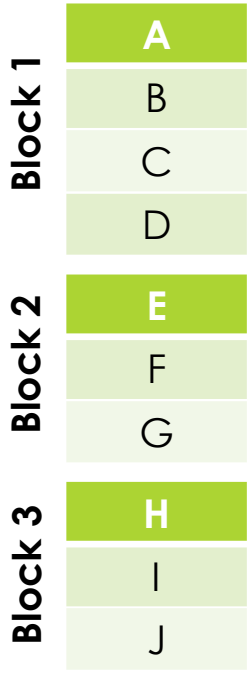
Characteristics of Computer Memory

- **Random Access:**
  - Data present at any location can be accessed directly
  - Time to access any memory location is same
  - Faster way to retrieve the data
  - RAM

A
B
C
D
F
G
H

Characteristics of Computer Memory

- **Direct Access:**
  - Memory divided into blocks (multiple memory locations),used by magnetic and optical discs
  - **Like Random access:**
    - Blocks can be accessed randomly, so block access time is same
  - **Unlike Random Access:**
    - But time taken to access a memory location A is different as compared to accessing a memory location J:
      - Sequential access inside the blocks



Characteristics of Computer Memory

- **Performance:**
  - Very important for user
  - Parameters for performance:
    - Access Time (latency)
    - Memory cycle time
    - Transfer Rate

Characteristics of Computer Memory

- **Access Time:**
  - **For Random Access Memory:**
    - Time taken for a read or write operation
    - Time from the instant that an address is presented to the time instant the data is stored or made available for use
  - **For Non-Random Access Memory:**
    - Time to position read-write mechanism at desired location

Characteristics of Computer Memory

- **Memory Cycle Time:**
  - Access Time+ Wait time (regenerate signals)
- **Transfer Rate:**
  - The rate at which data is transferred in/out of the memory
  - For Random-access memory, it is: 1/(cycle time)

Characteristics of Computer Memory

- **Transfer Rate for Non-Random Access Memory:**

$$T_n = T_A + \frac{n}{R}$$

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)

Problem

- Find out the time taken for write operation to take place in RAM, when memory cycle time is 5 seconds
- Find out the Transfer Rate for a Solid State Drive, when unit to read is 16 bits, average access time for read operation is 5 seconds, and average time to read n bits is 0.001 seconds

Characteristics of Computer Memory

- **Physical Types:**
  - Memory can be:
    - Semiconductor
    - Magnetic
    - Optical

Characteristics of Computer Memory

- **Physical Characteristics:**
  - Memory can be:
    - Volatile/Non Volatile
    - Erasable/Non Erasable

Summary

Table 4.1 Key Characteristics of Computer Memory Systems

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

The Memory Hierarchy

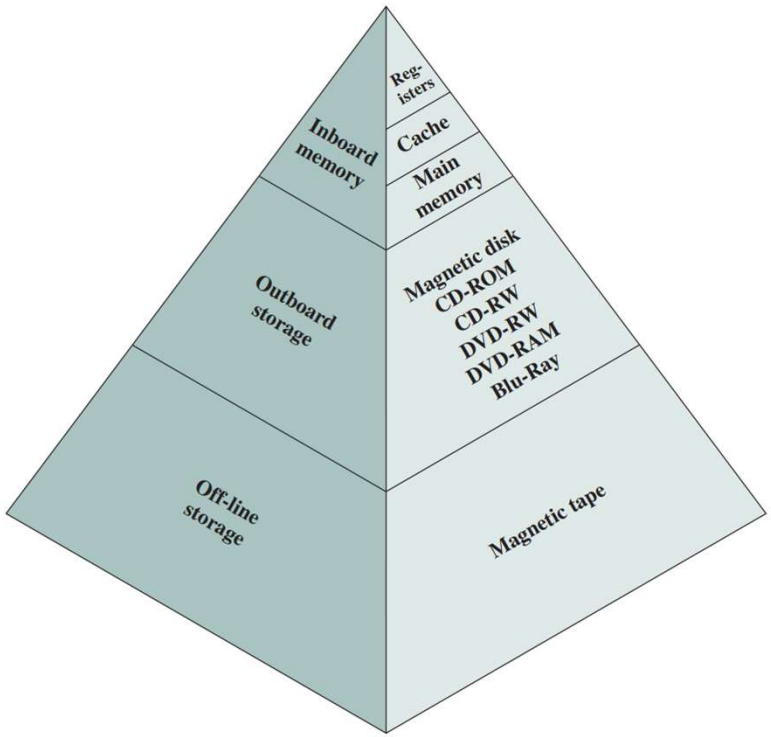
- **Design constraints on Computer's memory:**
  - How Much?
    - According to applications demand
  - How Fast?
    - Keep up with the processor, so that processor isn't idle
  - How Expensive?
    - Lets understand it

The Memory Hierarchy

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

The Memory Hierarchy

- Way out of this problem is to employ a memory hierarchy



The Memory Hierarchy

- As one goes down the hierarchy

  a. Decreasing cost per bit;
  b. Increasing capacity;
  c. Increasing access time;
  d. Decreasing frequency of access of the memory by the processor.

Hit and Miss

- **Hit:**
  - If the accessed word is found
- **Miss:**
  - If the accessed word is not found

Principle of Locality of reference

- A processor executes a program:
  - A program has arrays, loops etc
  - Inside a loop there are instructions which repeat
    - Same reference is used for a particular period of time

Principle of Locality of reference

- **Principle of locality of reference is the:**
  - **Tendency of a processor to access the same set of memory locations repetitively over a short period of time**
  - It makes sure frequently accessed instructions are in faster memory
  - Improves the hit ratio

4.2 Cache Memory Principles

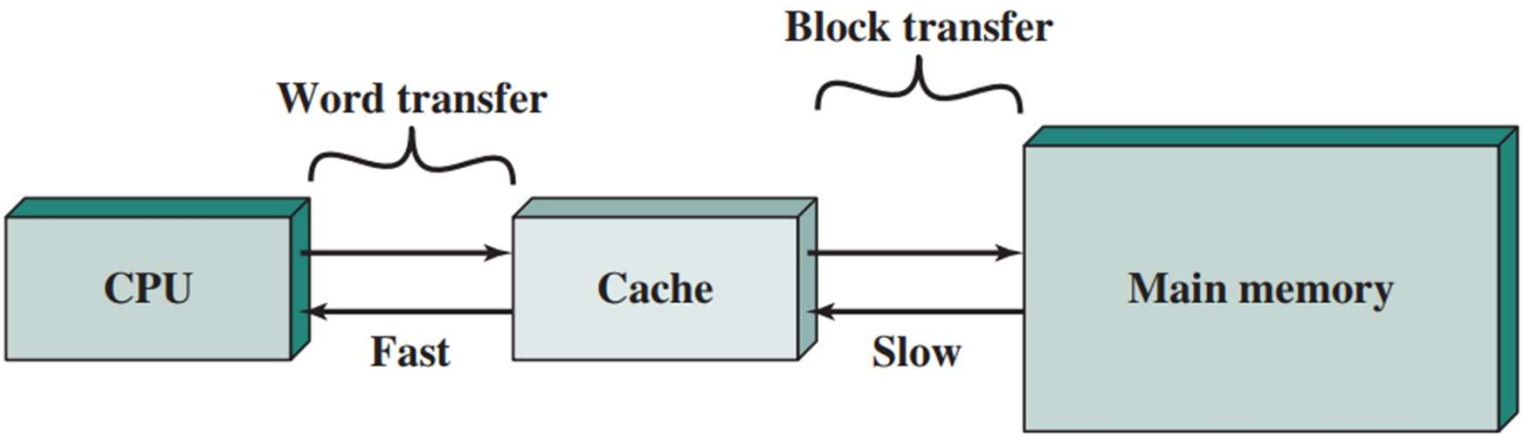
Cache Memory

- It combines the benefits of both:
  - Memory **access time** of **high speed** memory (cache itself)
  - **Large memory size** of **less expensive, lower speed** memory (main memory)

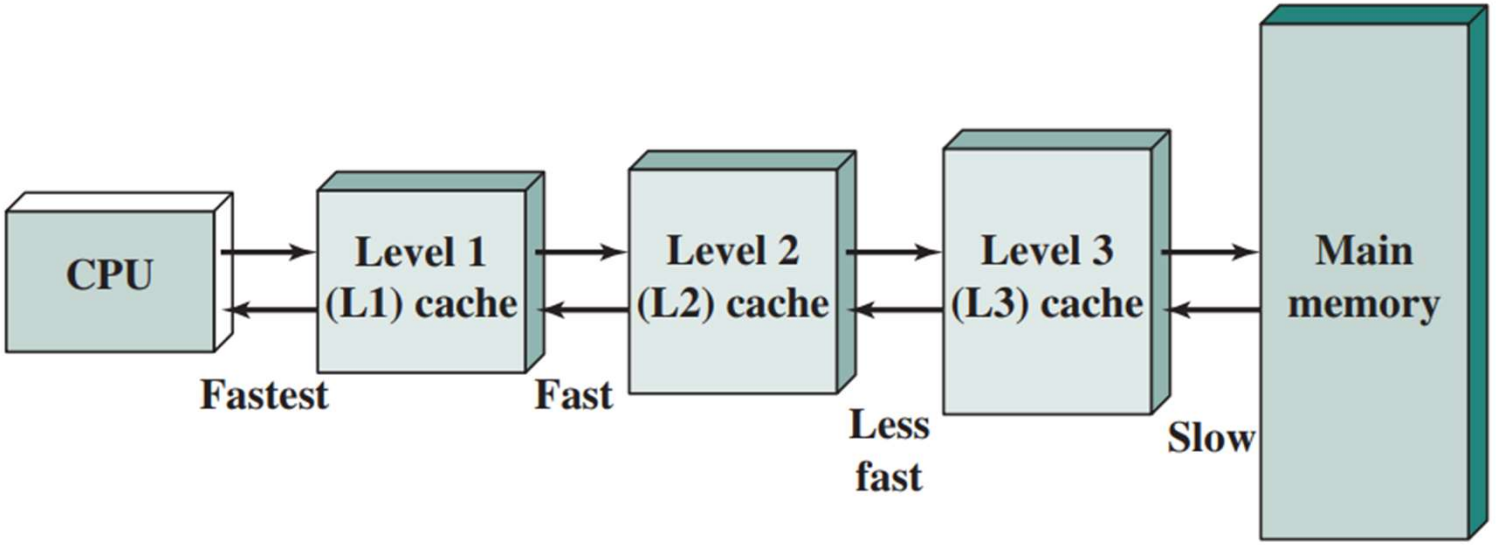
What happens when a processor wants to read a word from memory!

- Check is made in the **cache**
  - If **available**:
    - Word is delivered to the processor
  - If **not available:**
    - A Block of words is read from the memory, and the required word is delivered
    - But Why **Block**, Why not a single word:
      - **Principle of locality of reference:** Same word could be referenced again, or a word adjacent to the current word may be referenced

Cache memory

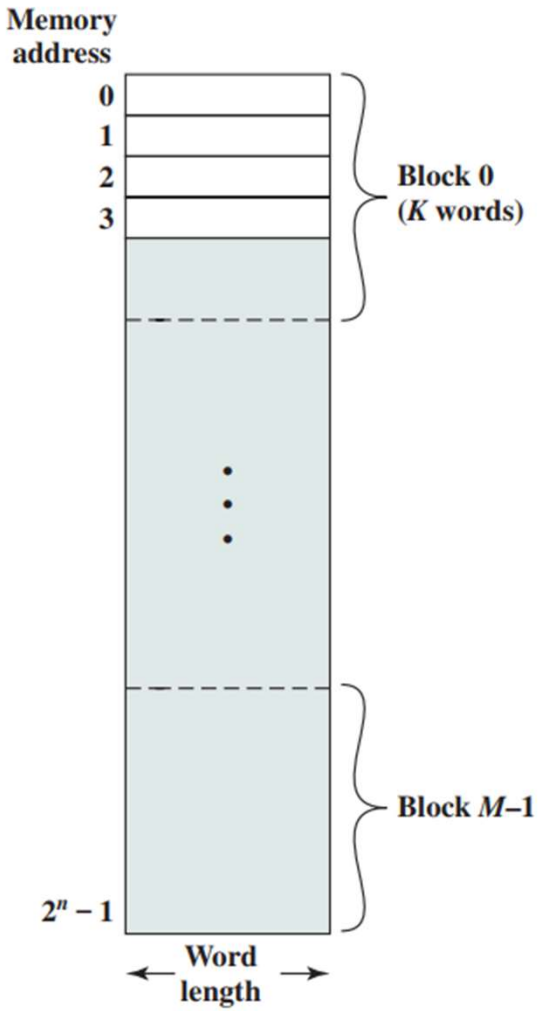


Levels of Cache



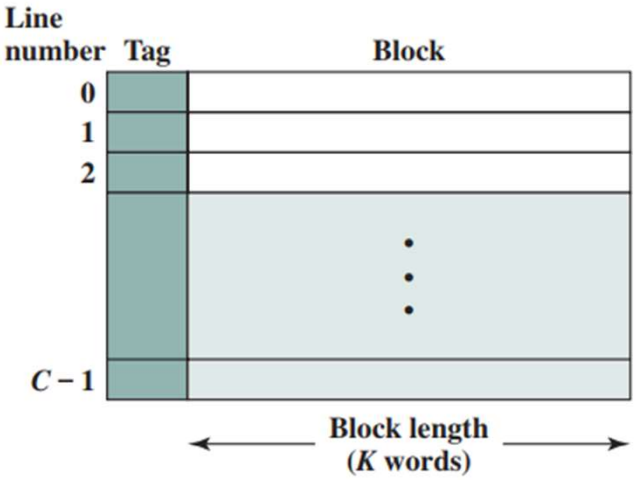
Structure of Main Memory System

- **N** is the **length** of each word in bits, so total **2^n -1 words** can be addressed
- Further organization can be converted to **blocks**:
  - Where each block may contain **K Words**


Memory address
0
1
2
3
Block 0 (K words)
Block M–1
2^n – 1
Word length

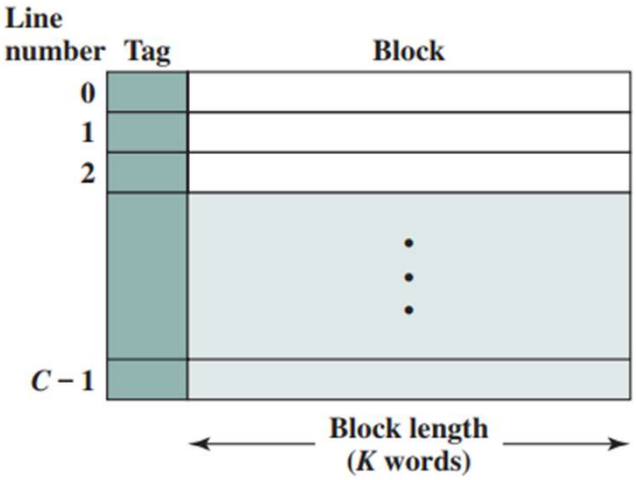
Structure of Cache System

- **Cache** consists of m blocks called **LINES**, each line may contain **K Words**



Structure of Cache System

- **Cache** consists of m blocks called **LINES**, each line may contain **K Words,** plus **a tag**



Structure of Cache

- Each **LINE** may also include some control bits:
  - To show whether the line has been modified since the data is loaded into it
- **Line Size:**
  - length of line, excluding the control and tag bits
  - Line size may be as small as 32 bits (4 words, each of one byte)

Structure of Cache

- Understanding the **Tag**:
  - Number of **lines in cache** is **smaller** than **the number of blocks** in main memory m<M
  - We can not map one-one (one line to one block)
  - Tag is usually a **memory address** which identifies **which block of memory** is residing in a **cache line**

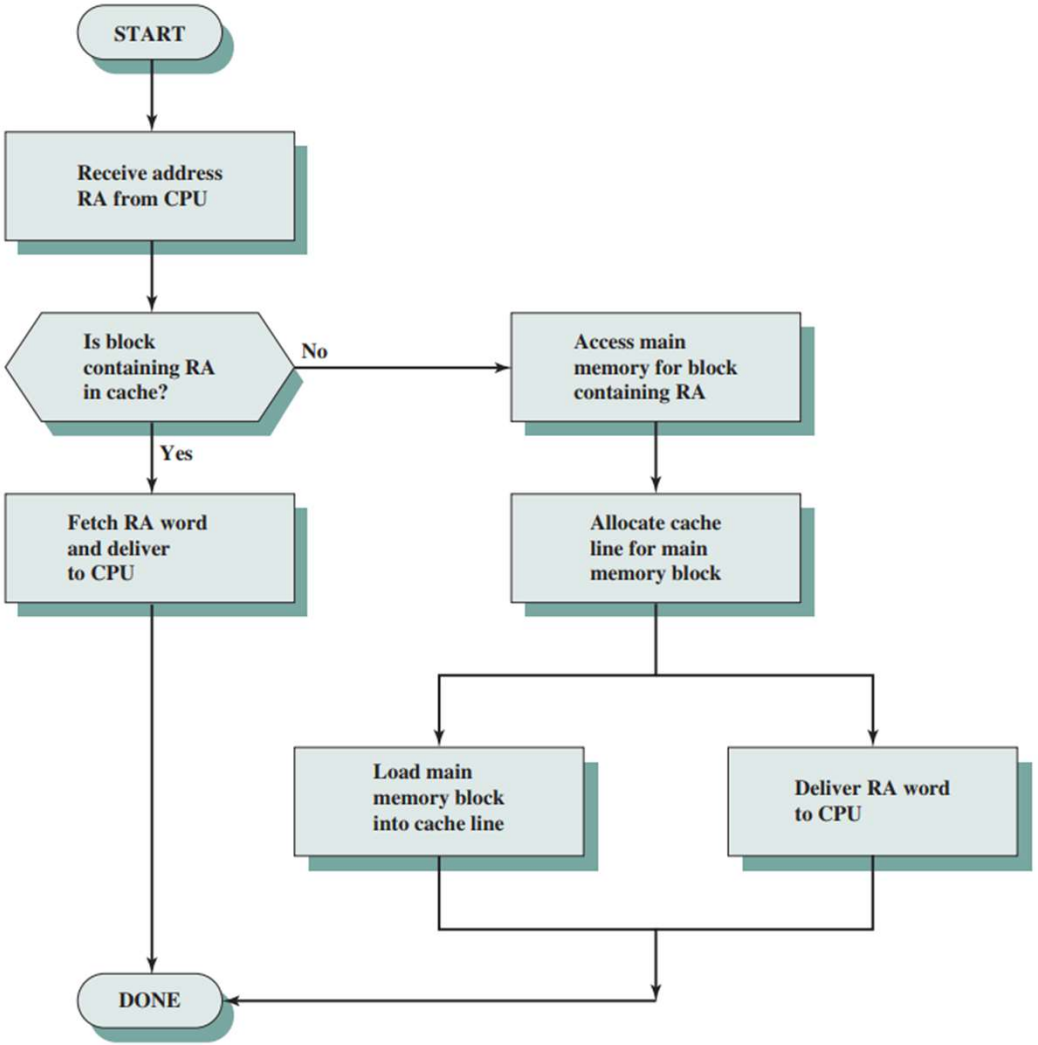


Figure 4.5 Cache Read Operation