COMPUTER MEMORY

Slides prepared by:
Dr. Zubair Ahmad Shah
Department of Computer Science and Engineering
Islamic University of Science and Technology, Awantipora

Why computers use memory?

– To store ***programs*** and ***data***.

What types of memories are there?

– Registers

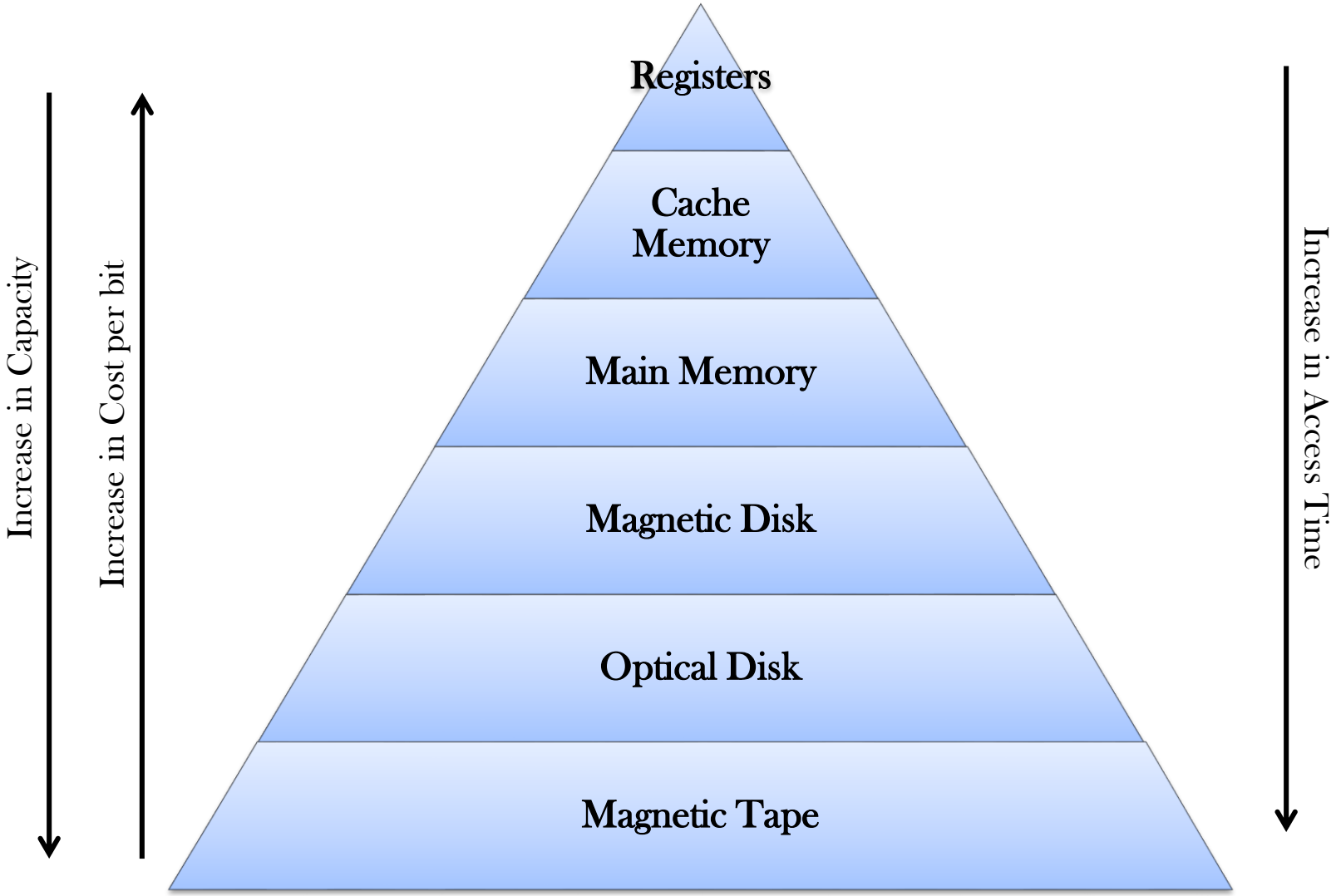
– Cache memory

– Main memory

– Auxiliary memory

– …

MEMORY HIERARCHY



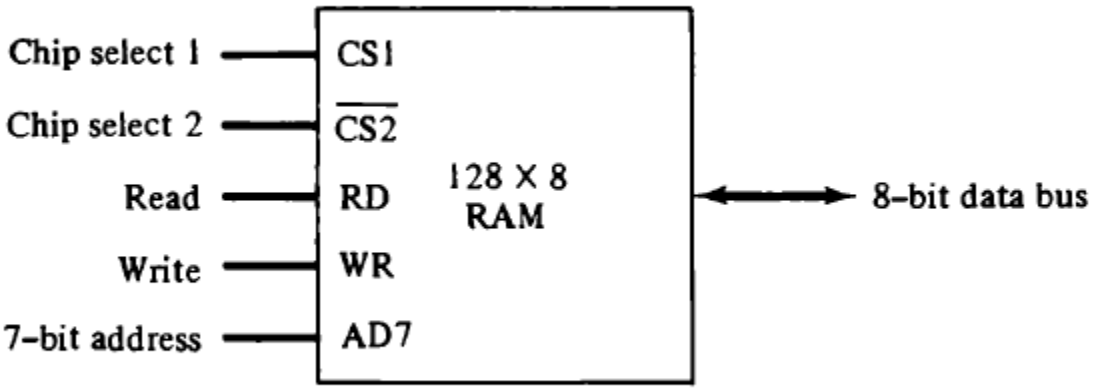
Slides prepared by: Dr. Zubair Ahmad Shah, Dept. of CSE, IUST

TOPIC: MAIN MEMORY

MAIN MEMORY

- The main technology used for the main memory is based on semiconductor integrated circuits.
- Types:
  - **Random Access Memory (RAM)**:
    - Static RAMs --> Flip-flops
    - Dynamic RAMs --> Capacitors (MOSFETS)
  - **Read-Only Memory (ROM)**

Typical RAM chip.

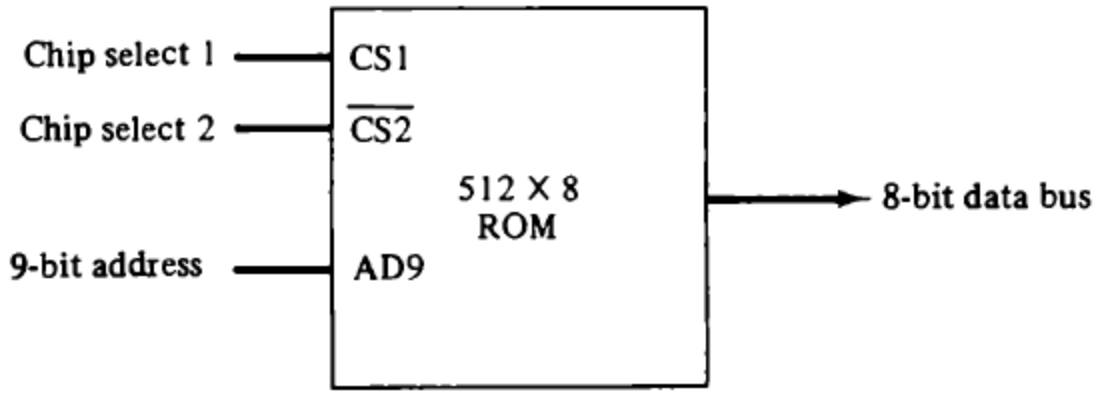


(a) Block diagram

CS1	$\overline{CS2}$	RD	WR	Memory function	State of data bus
0	0	×	×	Inhibit	High-impedance
0	1	×	×	Inhibit	High-impedance
1	0	0	0	Inhibit	High-impedance
1	0	0	1	Write	Input data to RAM
1	0	1	×	Read	Output data from RAM
1	1	×	×	Inhibit	High-impedance

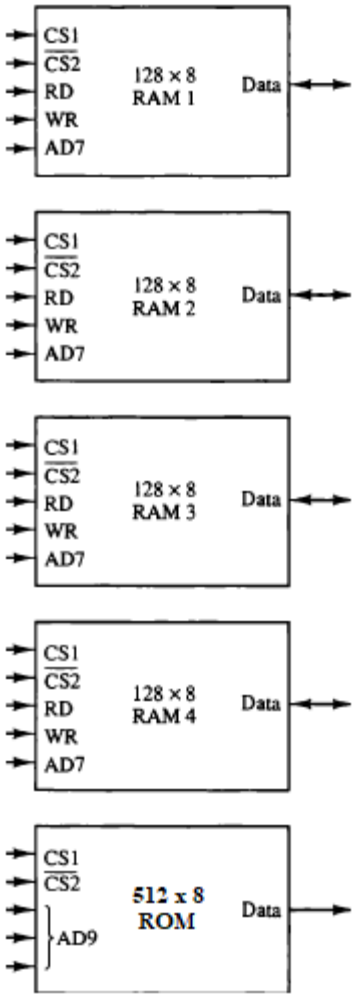
(b) Function table

MAIN MEMORY (cont.)



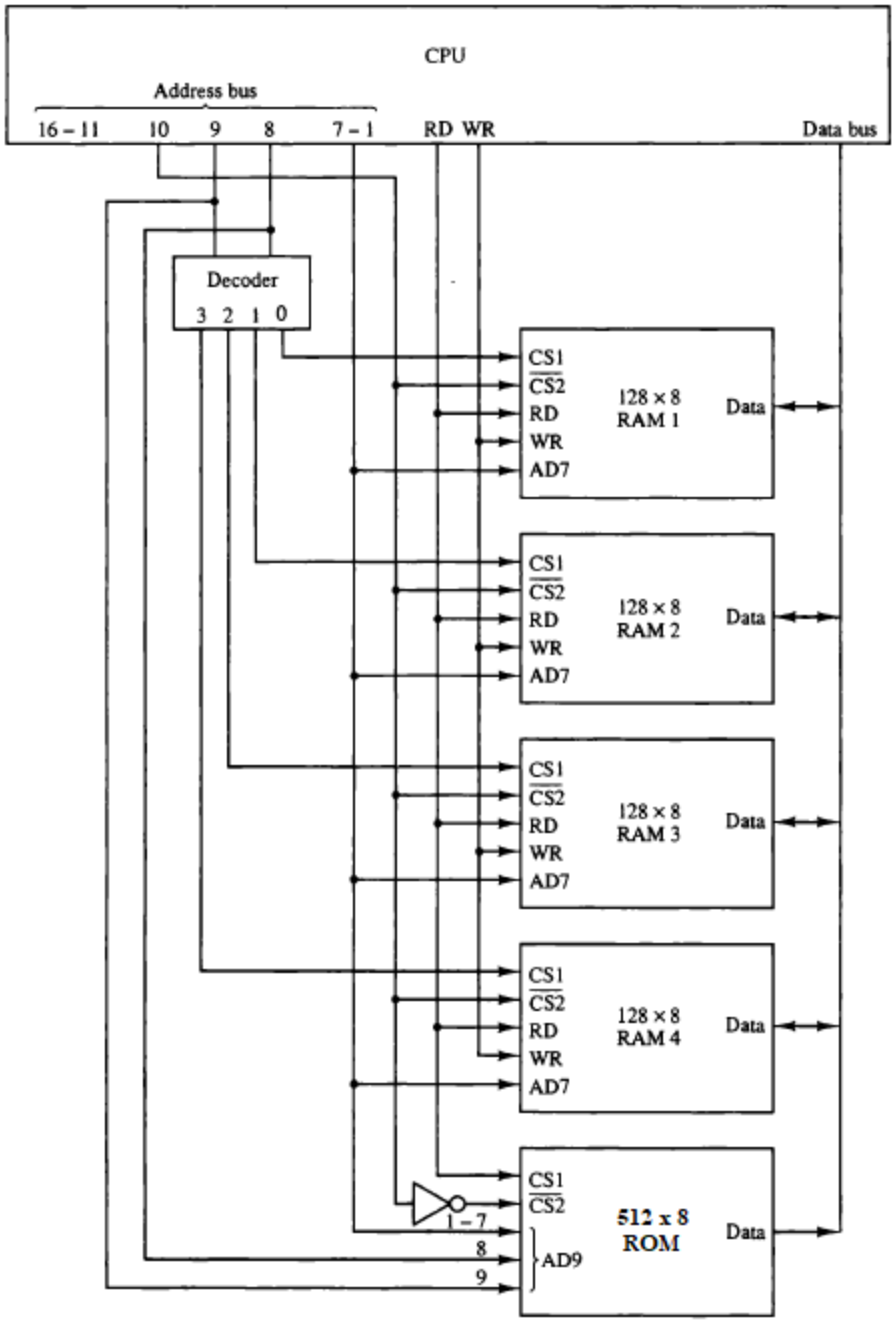
Typical ROM chip.

- Memory Address Map:
  - It is a pictorial representation of assigned address space for each chip in a system.



Memory Address Map for a micro-computer having *four 128x8 RAM chips and one 512x8 ROM chip*:

Component	Hexadecimal address	Address bus									
		10	9	8	7	6	5	4	3	2	1
RAM 1	0000–007F	0	0	0	x	x	x	x	x	x	x
RAM 2	0080–00FF	0	0	1	x	x	x	x	x	x	x
RAM 3	0100–017F	0	1	0	x	x	x	x	x	x	x
RAM 4	0180–01FF	0	1	1	x	x	x	x	x	x	x
ROM	0200–03FF	1	x	x	x	x	x	x	x	x	x



Memory Connections to the CPU
(of a micro-computer having four 128x8 RAM chips and one 512x8 ROM chip)

NUMERICAL PROBLEMS

Problem 1:

a) How many 128 x 8 RAM chips are needed to provide a memory capacity of 2048 bytes?

b) How many lines of the address bus must be used to access 2048 bytes of memory? How many of these lines will be common to all chips?

c) How many lines must be decoded for chip select? Specify the size of decoders.

Solution:

a) Required memory capacity = 2048 bytes = 2048 x 8 bits

Size of each RAM chip = 128 x 8 bits

=> No. of RAM chips needed = $\frac{2048 \times 8}{128 \times 8}$ = 16

b) Lines of address bus needed to access 2048 bytes of memory = $\log_2(2048)$

$= \log_2(2^{11}) = 11 \times \log_2(2) = 11$

Since each chip is of size 128 x 8, therefore lines that will be common to all chips = $\log_2(128) = 7$

c) Lines to be decoded for chip select = 4

Size of decoder = 4 x 16

Problem 2:

A ROM chip of 1024 x 8 bits has four select inputs and operates from a 5-volt power supply. How many pins are needed for the IC package?

Solution:

Address lines = $\log_2(1024)$ = 10

Data lines = 8

Power lines = 2 (1 for power supply and 1 for ground)

Chip select lines = 4

Total = 24

=> Pins needed for the IC package = 24

TOPIC: AUXILIARY MEMORY

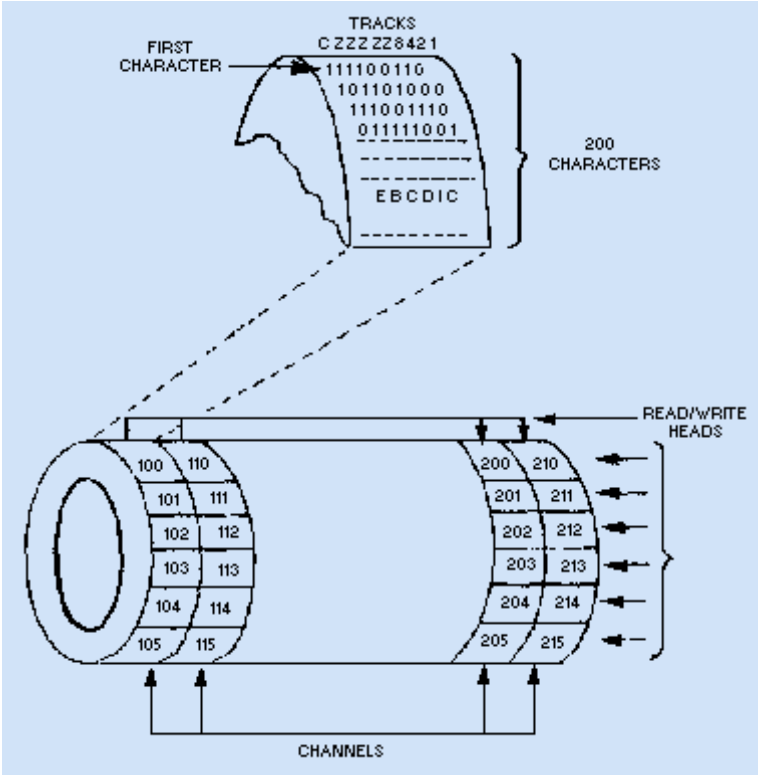
Auxiliary Memory

- **Examples**
  - **Magnetic Disk**
  - **Magnetic Drum**
  - **Magnetic Tape**
  - **Optical Disk**
  - **……..**

Auxiliary Memory Cont.

- **Important Characteristics:**
  - **Access Mode**
  - **Access Time** (= Seek Time + Transfer Time)
  - **Transfer Rate**
  - **Capacity**
  - **Cost**

MAGNETIC DRUM

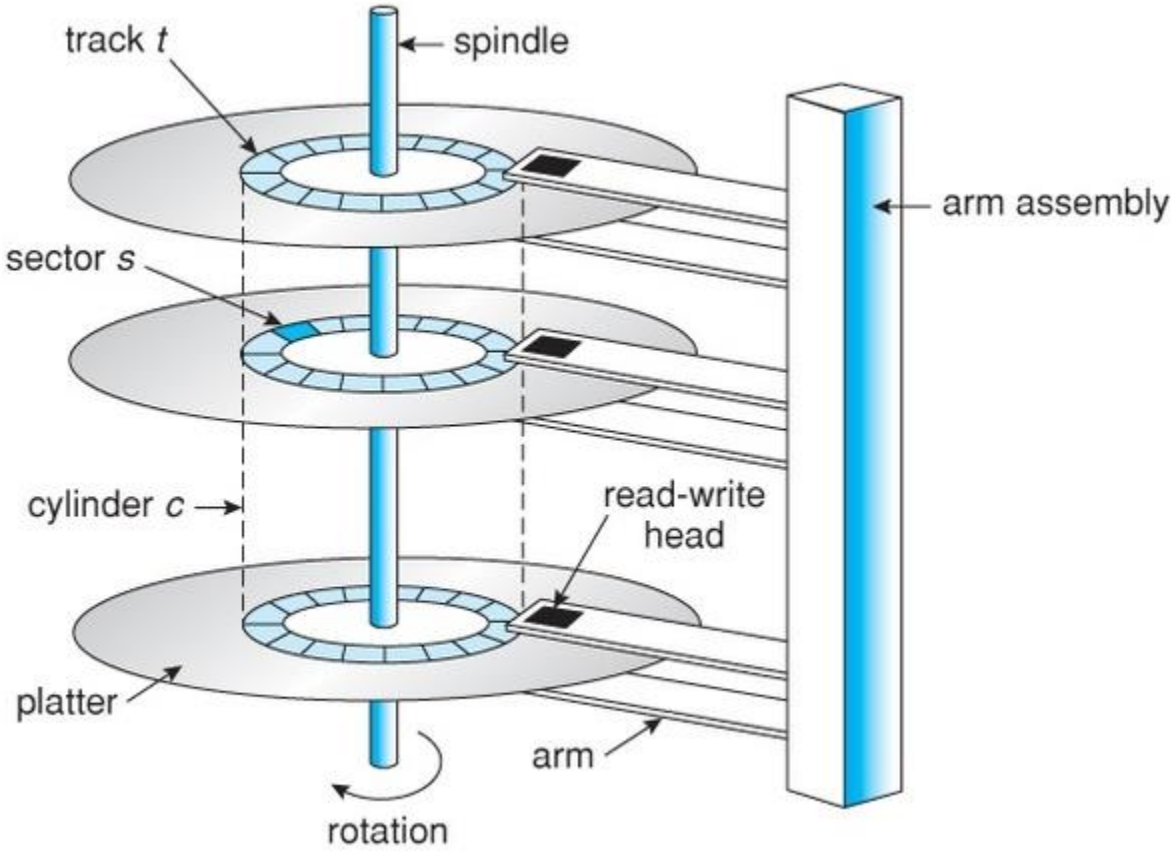


MAGNETIC DISK

Hard Disk

Floppy Disk

Hard Disk Internal Structure



Magnetic Disk Cont.

- **A disk system is addressed by address bits that specify**
  - **Disk number**
  - **Disk surface**
  - **Track**
  - **Sector number**

PROBLEM 1

A magnetic disk system has the following parameters:

T_s = average time to position the magnetic head over a track

R = rotation speed of disk in revolutions per second

N_t = number of bits per track

N_s = number of bits per sector

Calculate the average time T_a that it will take to read one sector.

SOLUTION:

Average time to read one sector = Average time to position the magnetic head over the track + Average time to reach the particular sector + Time to read bits of a sector

$$\Rightarrow \quad T_a = T_s + \left(\frac{1}{2} * \frac{1}{R}\right) + \left(Ns * \frac{1}{Nt} * \frac{1}{R}\right)$$

PROBLEM 2

What is the transfer rate of an eight-track magnetic tape whose speed is 120 inches per second and whose density is 1600 bits per inch?

SOLUTION:

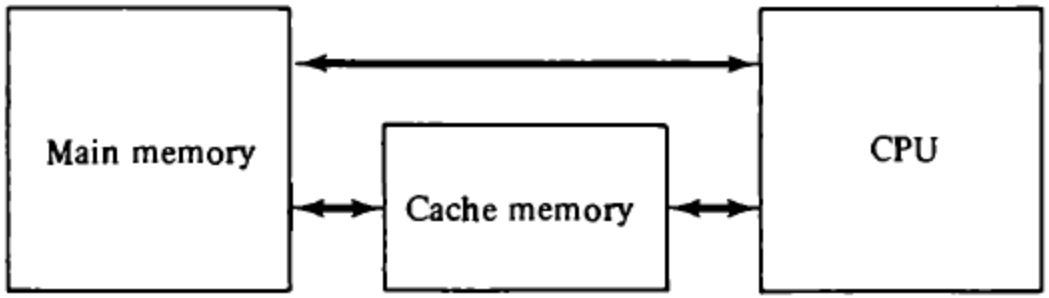
Bits in one inch of a track = 1600
Bits in one inch of the tape = 1600 x 8
Inches covered per second = 120

Transfer rate = Number of bits transferred per second
= 120 x (1600 x 8)

TOPIC: CACHE MEMORY

CACHE MEMORY

- Locality of Reference Property: It states that over a short interval of time, the addresses generated by a typical program refer to a few localized areas of memory repeatedly, while the remainder of memory is accessed relatively infrequently.
- If the active portions of the program and data (frequently accessed instructions and data) are placed in a fast small memory, the average access time can be reduced, thus reducing the total execution time of the program. Such a fast small memory is referred to as cache memory.



- When the CPU refers to memory and finds the word in cache, it is said to produce a *hit*.
- If the word is not found in cache, it is counted as a *miss*.

$$\textit{hit ratio} = \frac{\text{Number of hits}}{\text{Total number of CPU references to memory}}$$

$$= \frac{\text{Number of hits}}{\text{Number of hits + Number of misses}}$$

Problem 1:

The access time of a cache memory is 100 ns and that of main memory 1000 ns. It is estimated that 80 percent of the memory requests are for read and remaining 20 percent for write. The hit ratio for read accesses only is 0.9. A write-through procedure is used.

a) What is the average access time of the system considering only memory read cycles?

b) What is the average access time of the system for both read and write requests?

c) What is the hit ratio taking into consideration the write cycles?

Solution:

Access time of cache memory = 100 ns

Access time of main memory = 1000 ns

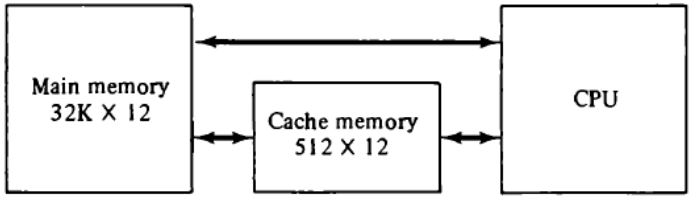
a) 0.9 x 100 + 0.1 x (100 + 1000) = 90 + 110 = 200 ns

b) (80/100) x 200 + (20/100) x 1000 = 160 + 200 = 360 ns

c) 0.9 x (80/100) = 0.72

- Mapping: The transformation of data from main memory to cache memory is referred to as mapping process.
- Considering the organization of cache memory, three types of mapping procedures are of practical interest:
  - Associative Mapping
  - Direct Mapping
  - Set-Associative Mapping

- **Associative Mapping**:



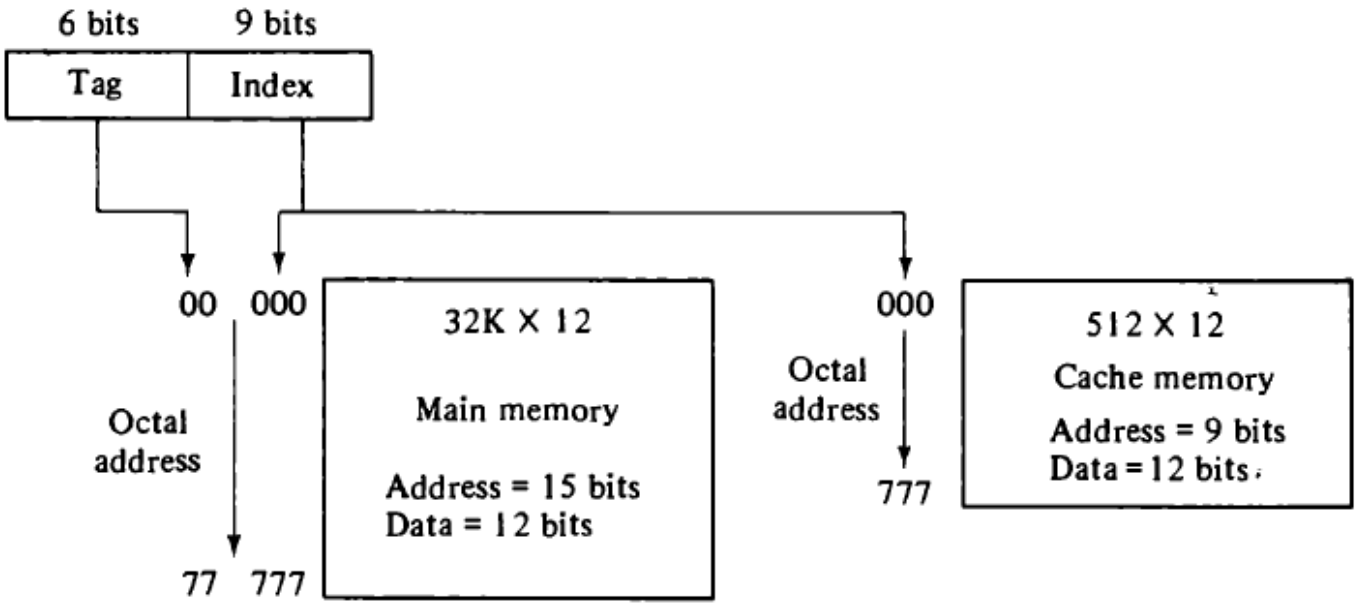
Example

CPU address (15 bits)

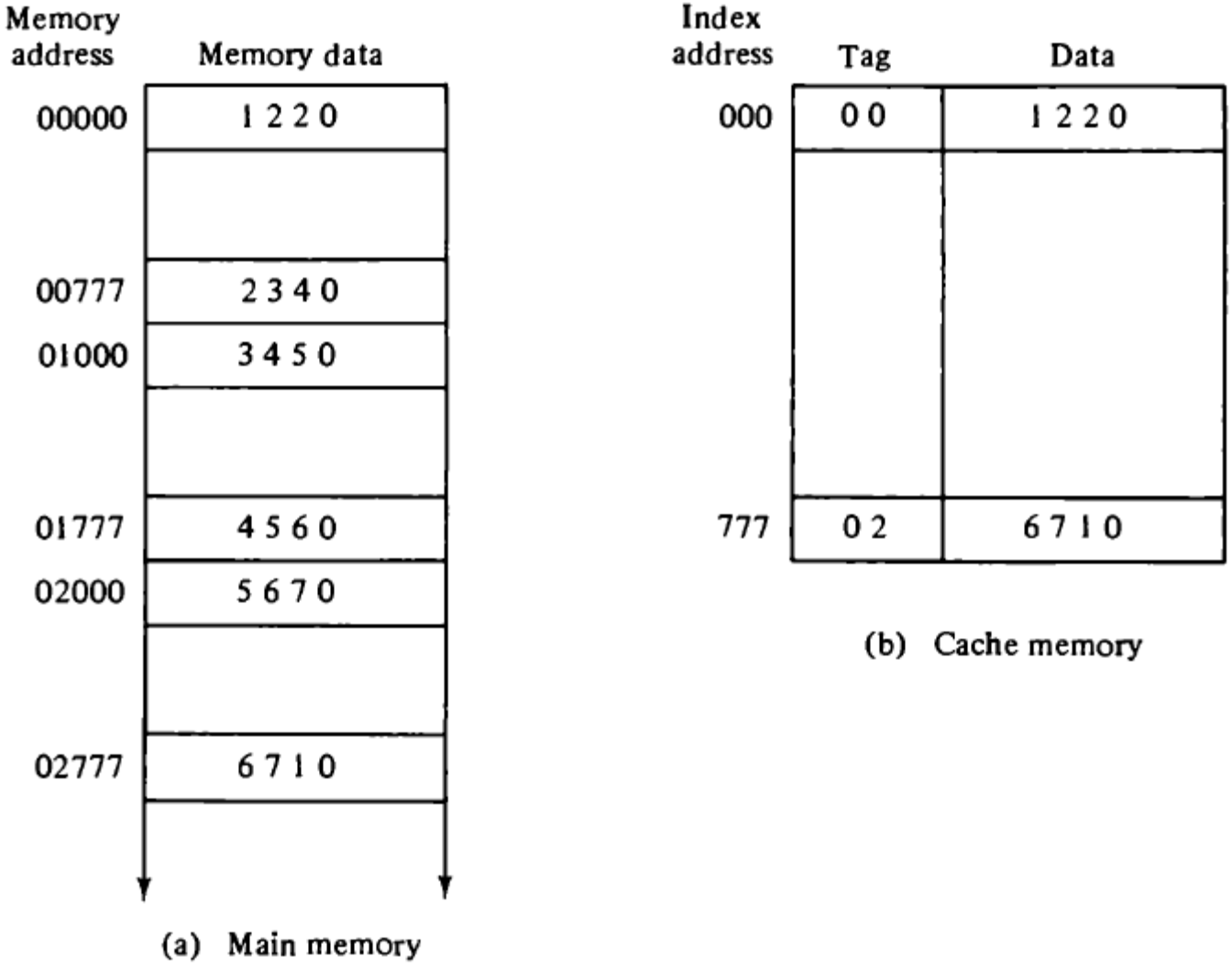
Address	Data
01000	3450
02777	6710
22345	1234

Associative mapping cache (all numbers in octal).

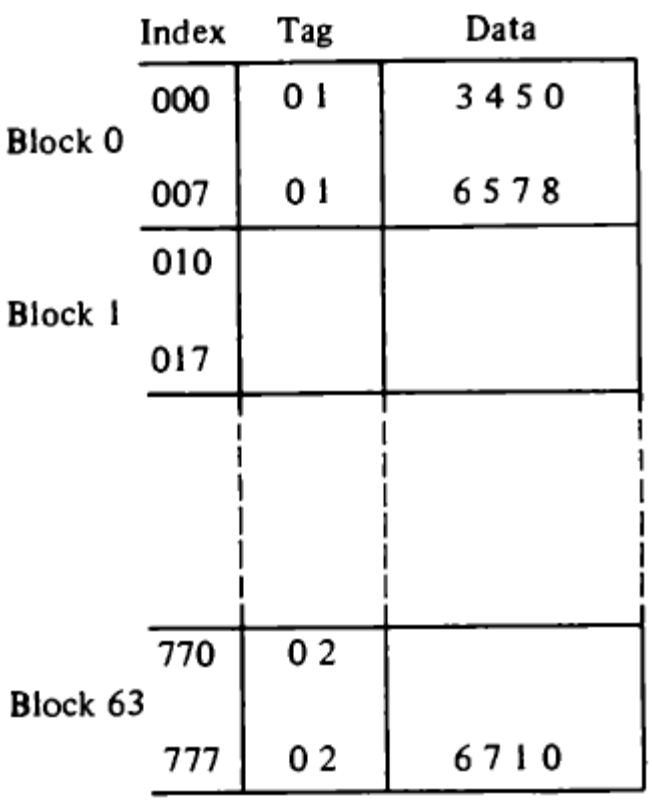
- **Direct Mapping:**

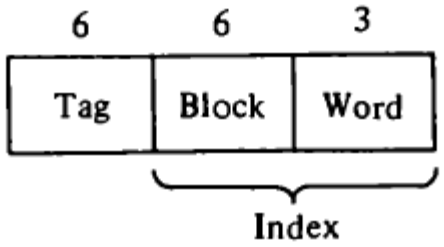


Addressing relationships between main and cache memories.



Direct mapping cache organization.





Direct mapping cache with block size of 8 words.

Problem 2:

A digital computer has a memory unit of 64K x 16 and a cache memory of 1K words. The cache uses direct mapping with a block size of four words.

a) How many bits are there in the tag, index, block and word fields of the address format?

b) How many bits are there in each word of cache, and how are they divided into functions? Include a valid bit.

c) How many blocks can the cache accommodate?

Solution:

Size of main memory = 64K x 16

Size of cache memory = 1K words

a) CPU Address bits = $\log_2(64K) = \log_2(2^{16}) = 16$

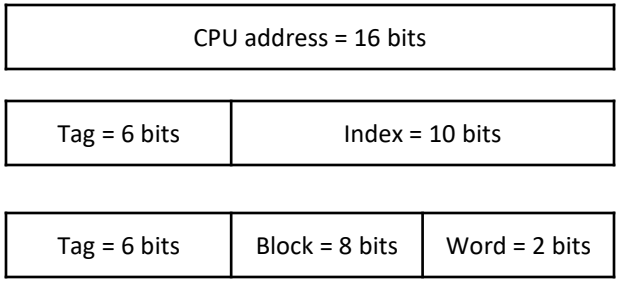
Index field bits = $\log_2(1K) = \log_2(2^{10}) = 10$

=> Tag field bits = 16 – 10 = 6

Cache block size = 4 words

=> Word field bits = 2

=> Block field bits = 10 – 2 = 8



b) Each word of cache = Tag field bits + Data bits + Valid bit

= 6 + 16 + 1 = 23 bits

c) Number of blocks the cache can accommodate = $2^8 = 256$

- **Set-Associative Mapping**:

Index	Tag	Data	Tag	Data
000	01	3450	02	5670
777	02	6710	00	2340

Two-way set-associative mapping cache.

Problem 3:

A two-way set associative cache memory uses blocks of four words. The cache can accommodate a total of 2048 words from main memory. The main memory size is 128K x 32.

a. Formulate all pertinent information required to construct the cache memory.

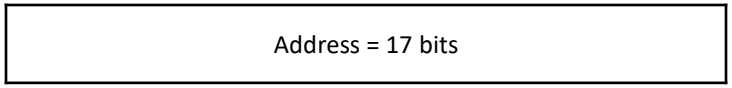
b. What is the size of the cache memory?

Solution:

Main memory size = 128K x 32

=> CPU Address bits = $\log_2(128K) = \log_2(2^{17}) = 17$

and Data bits = 32

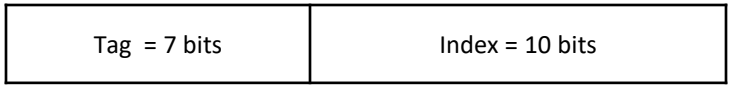


a. The (t*wo-way* set associative) cache can accommodate = 2048 words from main memory

=> Number of rows in cache = 2048/2 = 1024

=> Cache address bits (Index bits) = $\log_2(1024) = 10$

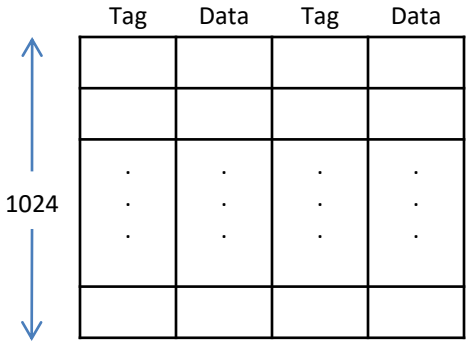
=> Tag bits = 17 – 10 = 7



Each block = 4 words

=> Word bits = 2

=> Block bits = 10 – 2 = 8



b. One row of cache memory

= Tag + Data + Tag + Data

= 7 + 32 + 7 + 32 = 78 bits

=> Size of cache memory = 1024 x 78

- **Writing into Cache:**
  - *Write-through* method
    - Update main memory with every memory write operation, with cache memory being updated in parallel if it contains the word at the specified address.
  - *Write-back* method
    - Only the cache memory is updated during a write operation.
    - The location is then marked by a flag so that later when the word is removed from the cache it is copied into main memory.

TOPIC: VIRTUAL MEMORY

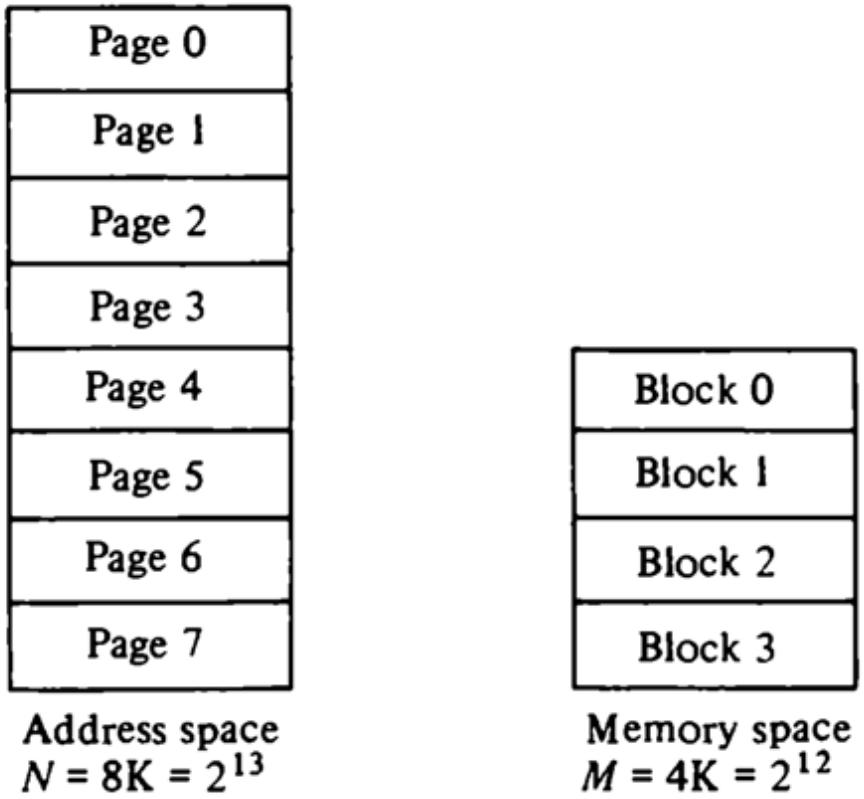
VIRTUAL MEMORY

- Virtual memory is a memory management technique where secondary memory can be used as if it were a part of the main memory.
- It is a concept by which a computer can address more memory than the amount physically installed on the system.
- The computer's operating system, using a combination of hardware and software, maps memory addresses used by a program, called virtual addresses, into physical addresses in computer memory.

- Virtual Address vs. Physical Address
- Address Space vs. Memory Space
- Paging
  - Page vs. Frame

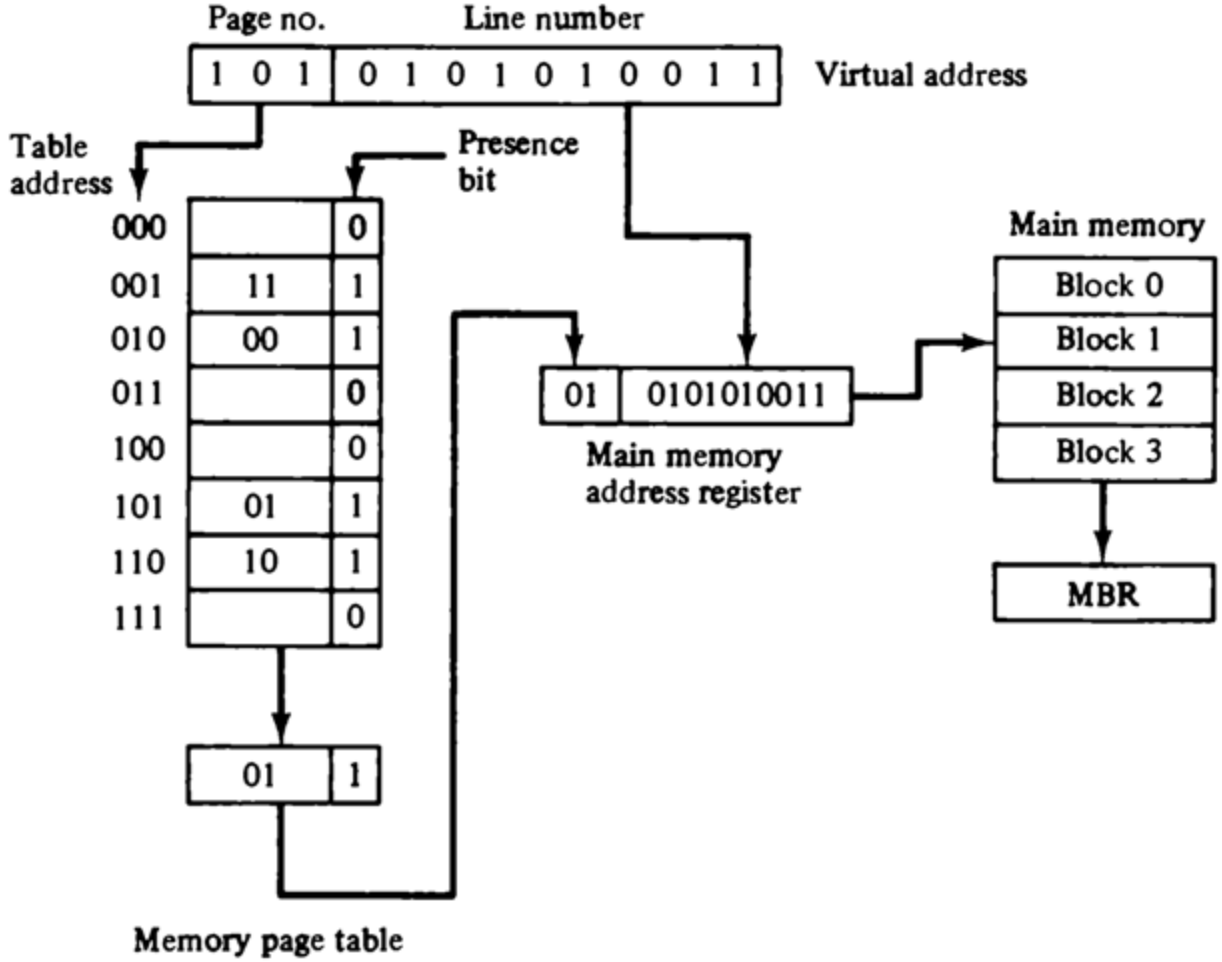
Paging

- In computer operating systems, paging is a memory management scheme by which a computer stores and retrieves data from secondary storage for use in main memory.
- In this scheme, the operating system retrieves data from secondary storage in same-size blocks called pages.
- Paging is an important part of virtual memory implementations in modern operating systems, using secondary storage to let programs exceed the size of available physical memory.



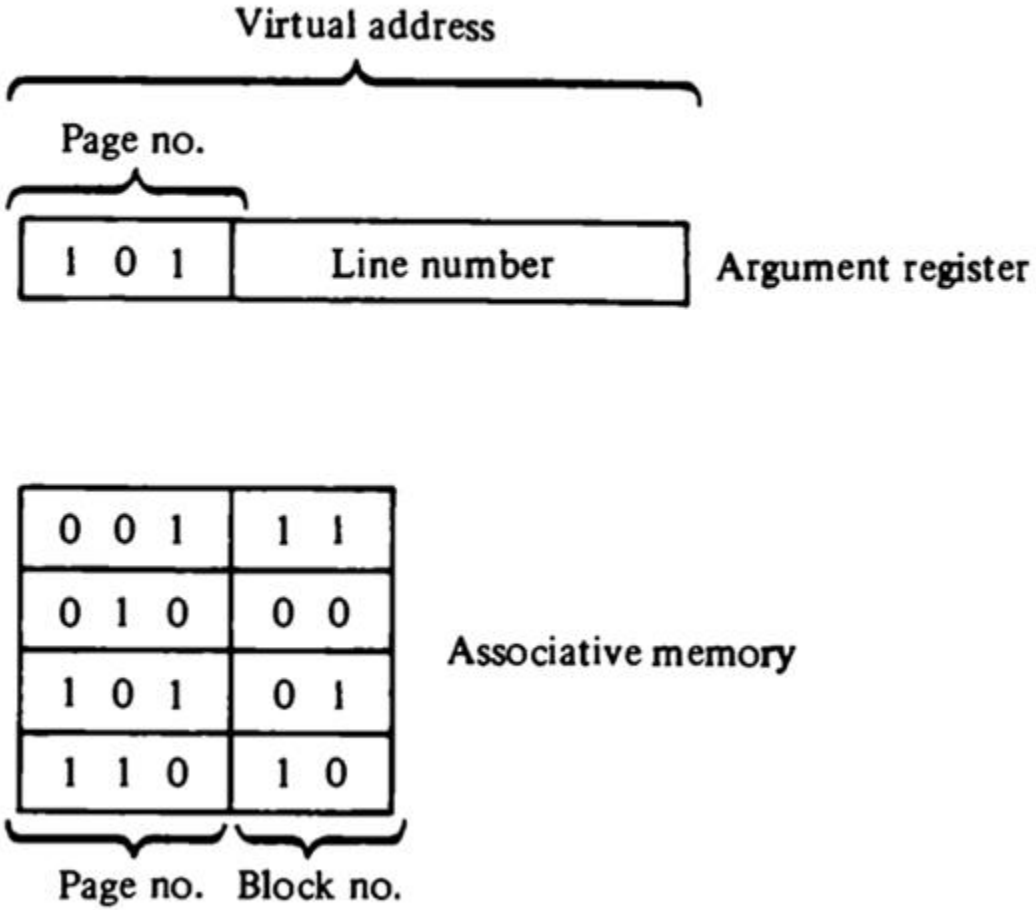
Address space and memory space split into groups of 1K words.

Memory table in a paged system.



- A page table can also be built using associative memory:

An associative memory page table.



- What is a Page Fault?

- If main memory is full, then replacement is needed at the time of page fault.
  - Page Replacement Algorithms:
    - FIFO (First In First Out)
    - LRU (Least Recently Used)

Numerical Problem

Question:

An address space is specified by 24 bits and the corresponding memory space by 16 bits.

a. How many words are there in the address space?

b. How many words are there in the memory space?

c. If a page consists of 2K words, how many pages and blocks are there in the system?

Solution:

a. 2^{24}

b. 2^{16}

c. No. of Pages $= 2^{24}/2K$

$= 2^{24}/2^{11} = 2^{13} = 8192$

No. of Blocks $= 2^{16}/2K$

$= 2^{16}/2^{11} = 2^{5} = 32$

REFERENCE

Morris Mano, Computer System Architecture, Prentice-Hall of India.