Noida Institute of Engineering and Technology, Greater Noida

Memory Unit

Unit: 4

Computer Organization & Architecture

Pradeep Kumar
Assistant Professor
CSE
Department

B Tech 3rd Sem

CSE 3rd Semester Evaluation Scheme

EVALUATION SCHEME
SEMESTER-III

Sl. No.	Subject Codes	Subject Name	Periods			Evaluation Schemes				End Semester		Total	Credit
			L	T	P	CT	TA	TOTAL	PS	TE	PE		
WEEKS COMPULSORY INDUCTION PROGRAM													
1	AAS0301A	Engineering Mathematics III	3	1	0	30	20	50		100		150	4
2	ACSE0306	Discrete Structures	3	0	0	30	20	50		100		150	3
3	ACSE0304	Digital Logic & Circuit Design	3	0	0	30	20	50		100		150	3
4	ACSE0301	Data Structures	3	1	0	30	20	50		100		150	4
5	ACSE0302	Object Oriented Techniques using Java	3	0	0	30	20	50		100		150	3
6	ACSE0305	Computer Organization & Architecture	3	0	0	30	20	50		100		150	3
7	ACSE0354	Digital Logic & Circuit Design Lab	0	0	2				25		25	50	1
8	ACSE0351	Data Structures Lab	0	0	2				25		25	50	1
9	ACSE0352	Object Oriented Techniques using Java Lab	0	0	2				25		25	50	1
10	ACSE0359	Internship Assessment-I	0	0	2				50			50	1
11	ANC0301/ ANC0302	Cyber Security*/ Environmental Science*(Non	2	0	0	30	20	50		50		100	0

Syllabus

UNIT-I	Introduction	8 Hours
Computer Organization and Architecture, Functional units of digital system and their interconnections, buses, bus architecture, types of buses and bus arbitration and it's types. Register, bus and memory transfer. Process or organization, general registers organization, stack organization and address in g modes.		
UNIT-II	ALU Unit	8 Hours
Arithmetic and logic unit: Lookahead carries adders. Multiplication: Signed operand multiplication, Booth's algorithm and array multiplier. Division and logic operations. Floating point arithmetic operation, Arithmetic & logic unit design. IEEE Standard for Floating Point Numbers.		
UNIT-III	Control Unit	8 Hours
Control Unit: Instruction types, formats, instruction cycles and sub cycles (fetch and execute etc.), micro-operations, execution of a complete instruction. Program Control, Reduced Instruction Set Computer, Complex Instruction Set Computer, Pipelining. Hardwire and microprogrammed control, Concept of horizontal and vertical microprogramming, Flynn's classification.		
UNIT-IV	Memory Unit	8 Hours
Memory: Basic concept and hierarchy, semiconductor RAM memories, 2D & 2 1/2D memory organization. ROM memories. Cache memories: concept and design issues & performance, address mapping and replacement Auxiliary memories: magnetic disk, magnetic tape and optical disks Virtual memory: concept implementation, Memory Latency, Memory Bandwidth, Memory Seek Time.		
UNIT-V	Input/Output	8 Hours
Peripheral devices, I/O interface, I/O ports, Interrupts: interrupt hardware, types of interrupts and exceptions. Modes of Data Transfer: Programmed I/O, interrupt initiated I/O and Direct Memory Access. ,I/O channels and processors. Serial Communication: Synchronous & asynchronous communication.		

Course Objective

- The objective of this course is to understand the types of organizations, structures and functions of computer, design of arithmetic and logic unit and float point arithmetic as well as to understand the concepts of memory system, communication with I/O devices and interfaces

Course Outcome

Understand the basic structure and operation of a digital computer system.
Analyze the design of arithmetic & logic unit and understand the fixed point and floating-point arithmetic operations.
Implement control unit techniques and the concept of Pipelining
Understand the hierarchical memory system, cache memories and virtual memory.
Understand different ways of communicating with I/O devices and standard I/O interfaces.

Program Outcome

1.Engineering knowledge

2. Problem analysis

3. Design/development of solutions

4. Conduct investigations of complex problems

5. Modern tool usage

6. The engineer and society

7. Environment and sustainability

8. Ethics

9. Individual and team work

10. Communication

11. Project management and finance

12. Life-long learning

CO-PO Mapping

COMPUTER ORGANIZATION AND ARCHITECTURE (ACSE0305)												
CO.K	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
ACSE0305.4	3	2	2	2	2	1	1	1	1	1	1	2

Program Specific Outcome

On successful completion of graduation degree, The computer Science & Engineering graduates will be able to:

PSO1: identify, analyze real world problems and design their ethical solutions using artificial intelligence, robotics, virtual/augmented reality, data analytics, block chain technology, and cloud computing.

PSO2: design and develop the hardware sensor devices and related interfacing software systems for solving complex engineering problems.

PSO 3: understand inter-disciplinary computing techniques and to apply them in the design of advanced computing.

PSO 4: conduct investigation of complex problem with the help of technical, managerial, leadership qualities, and modern engineering tools provided by industry sponsored laboratories.

CO- PSO Mapping

COMPUTER ORGANIZATION AND ARCHITECTURE (ACSE0305)				
CO.K	**PSO1**	**PSO2**	**PSO3**	**PSO4**
KCS-302.2	2	3	3	2

Program Educational Objectives

PEO 1: To have an excellent scientific and engineering breadth so as to comprehend, analyze, design and provide sustainable solutions for real-life problems using state-of-the-art technologies.

PEO 2: To have a successful career in industries, to pursue higher studies or to support entrepreneurial endeavors and to face the global challenges.

PEO 3: To have an effective communication skills, professional attitude, ethical values and a desire to learn specific knowledge in emerging trends, technologies for research, innovation and product development and contribution to society.

PEO 4: To have life-long learning for up-skilling and re-skilling for successful professional career as engineer, scientist, entrepreneur and bureaucrat for betterment of society.

Prerequisite and Recap

- Basics of Computer Organization & architecture
- Functional unit and their interconnection

Content

Memory:

- Basic concept and hierarchy
- semiconductor RAM memories
- 2D & 2 1/2D memory organization
- ROM memories
- Cache memories: concept and design issues & performance address mapping and replacement
- Auxiliary memories: magnetic disk, magnetic tape and optical disks
- Virtual memory: concept implementation

.

Unit Objective

- Study the various memory used by computer like hierarchical memory system, RAM, ROM, 2D & 2 1/2D memory organization.

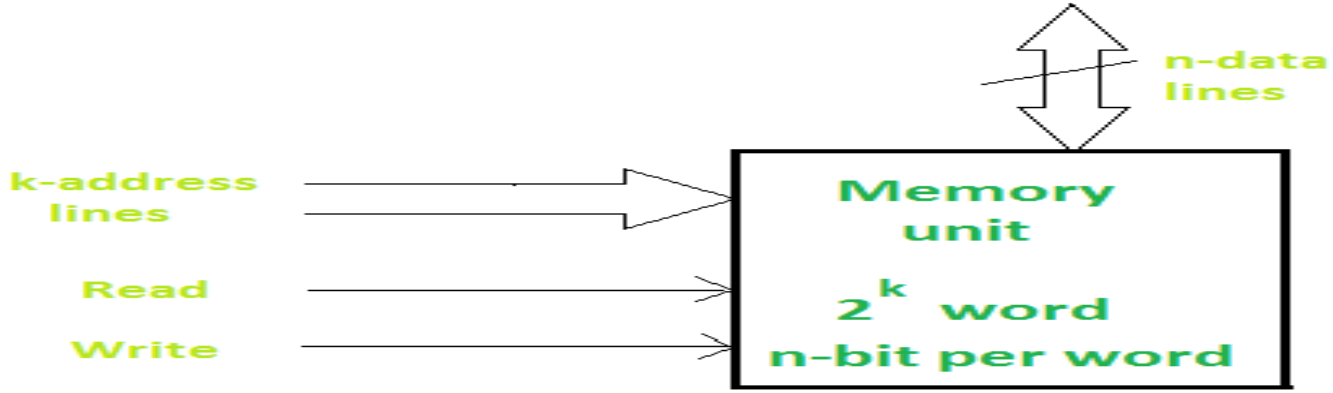
- Study of Cache memories and virtual memory , Auxiliary memories: magnetic disk, magnetic tape and optical disks .

Memory

- A **Memory Unit** is a collection of storage cells together with associated circuits needed to transfer information in and out of storage.
- The memory stores **binary information(1's and 0's)** in groups of **bits** called **words**. A storage element is called a **Cell**.
- A group of **eight bits** is called a **byte**. Most computer memories use words whose number of bits is a multiple of 8.
- The capacity of memories in commercial computers is usually stated as the total number of bytes that can be stored.

Memory

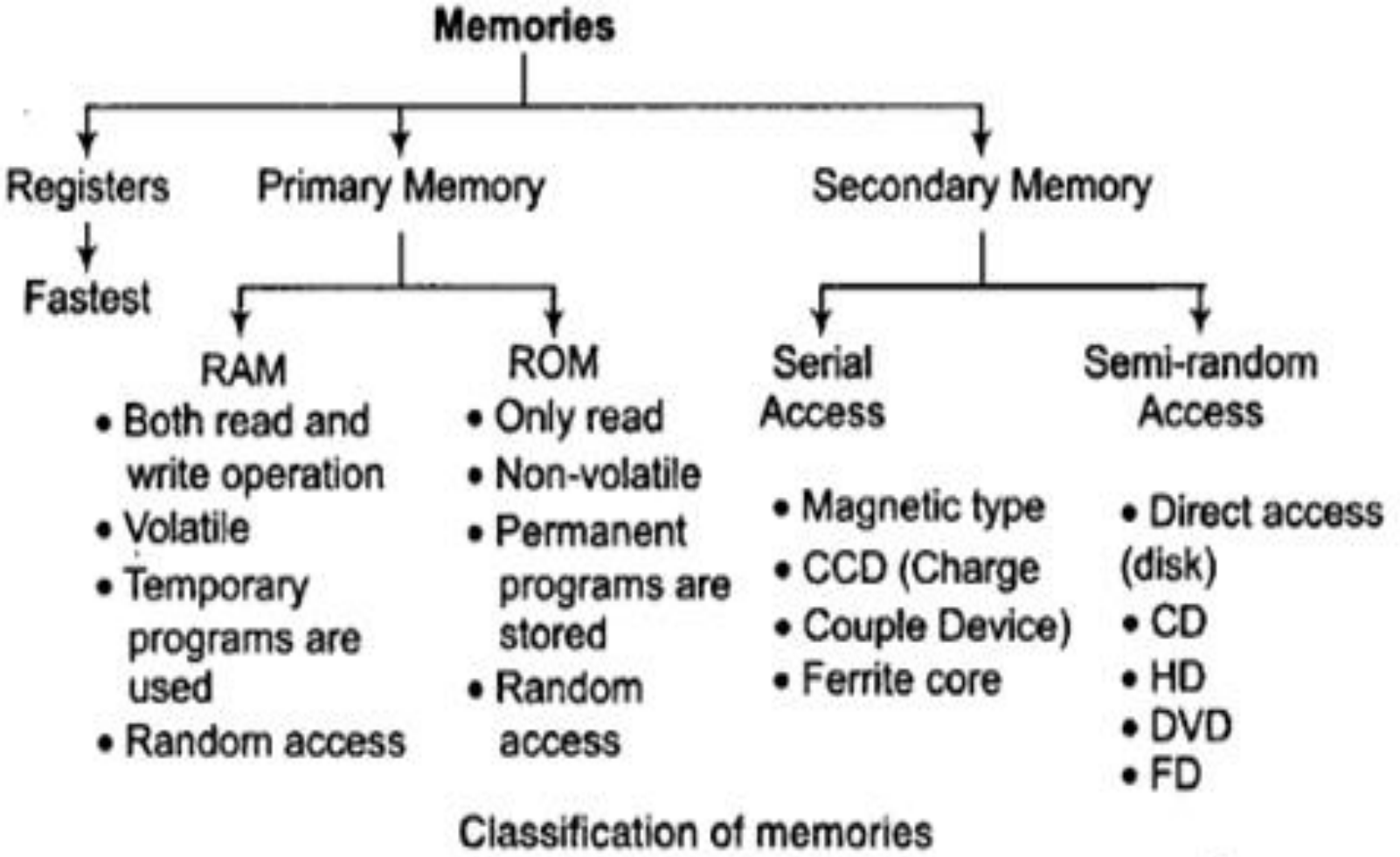
- A memory unit consists of data lines, address selection lines, and control lines that specify the direction of transfer. The block diagram of a memory unit is shown below:



•Data lines provide the information to be stored in memory. The control inputs specify the direction transfer.

•The k-address lines specify the word chosen. When there are k address lines, 2^k memory word can be accessed.

Classification of Memory



Classification of memories

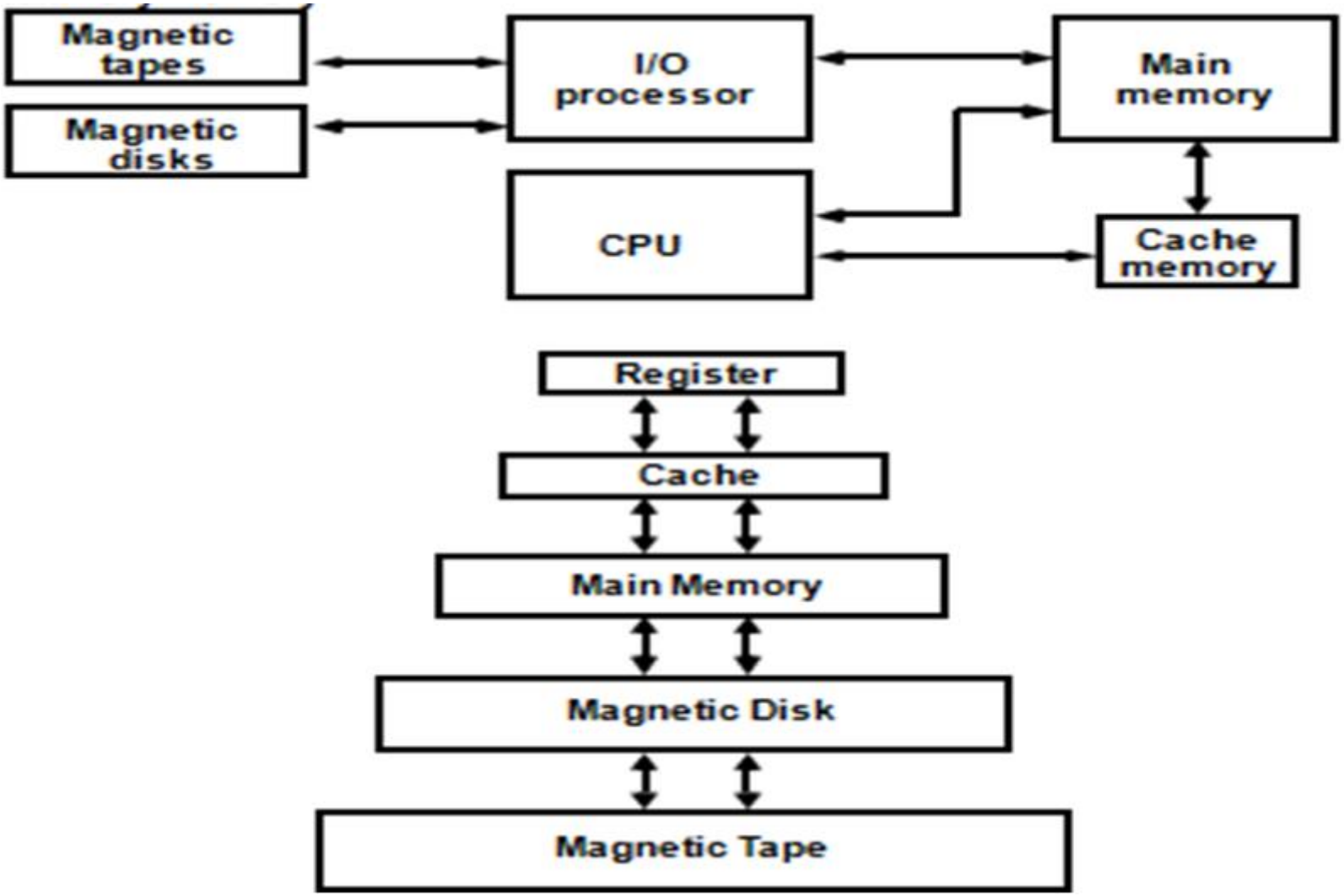
Memory Hierarchy

•The **memory hierarchy** system consists of all storage devices contained in a **computer** system from the slow Auxiliary **Memory** to fast and to smaller Cache **memory and Register**.

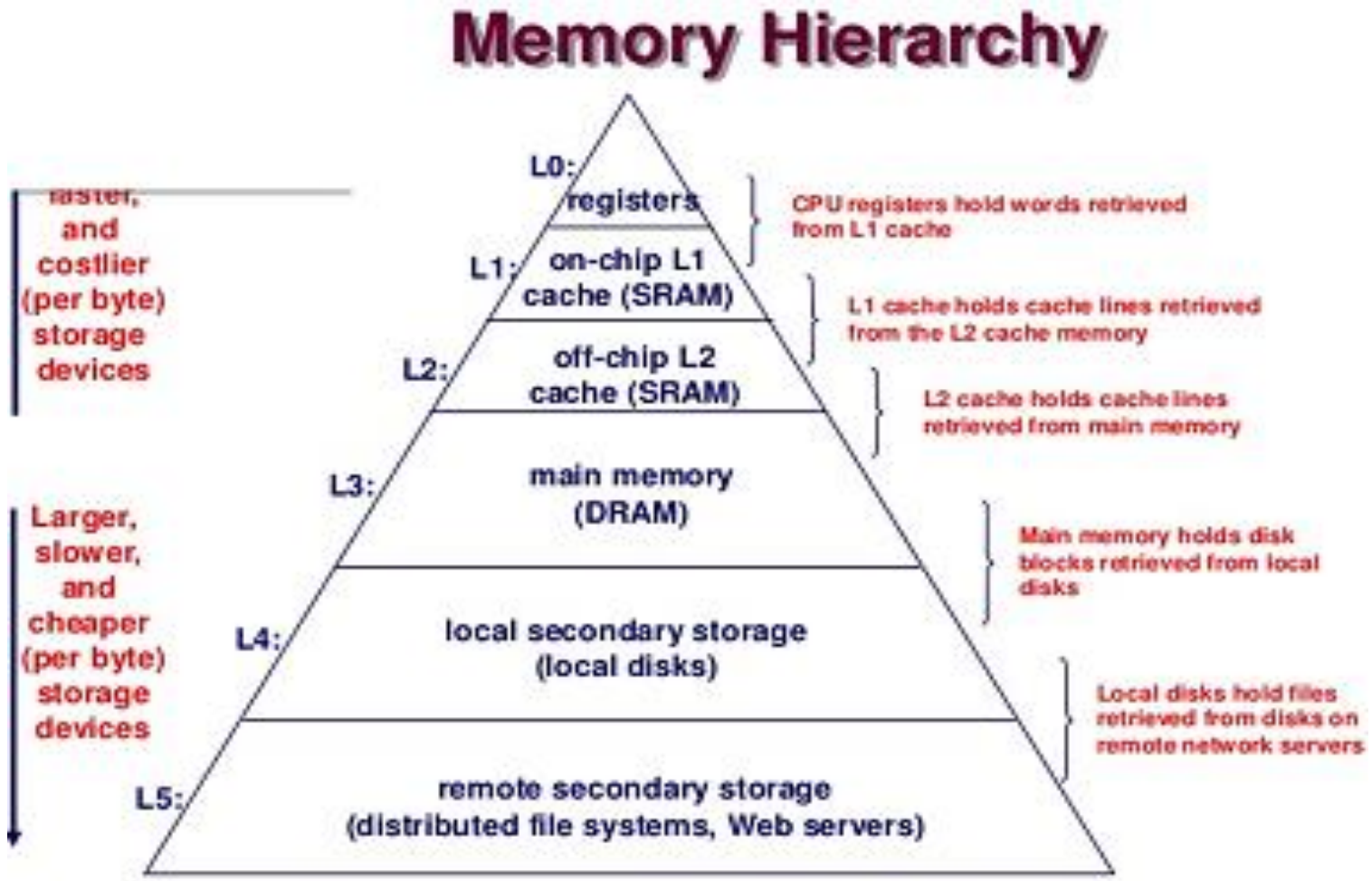
•Auxiliary **memory** access time is generally 1000 times that of the main **memory**, hence it is at the bottom of the **hierarchy**.

•The Memory Hierarchy was developed based on a program behavior known as locality of references.

Memory Hierarchy

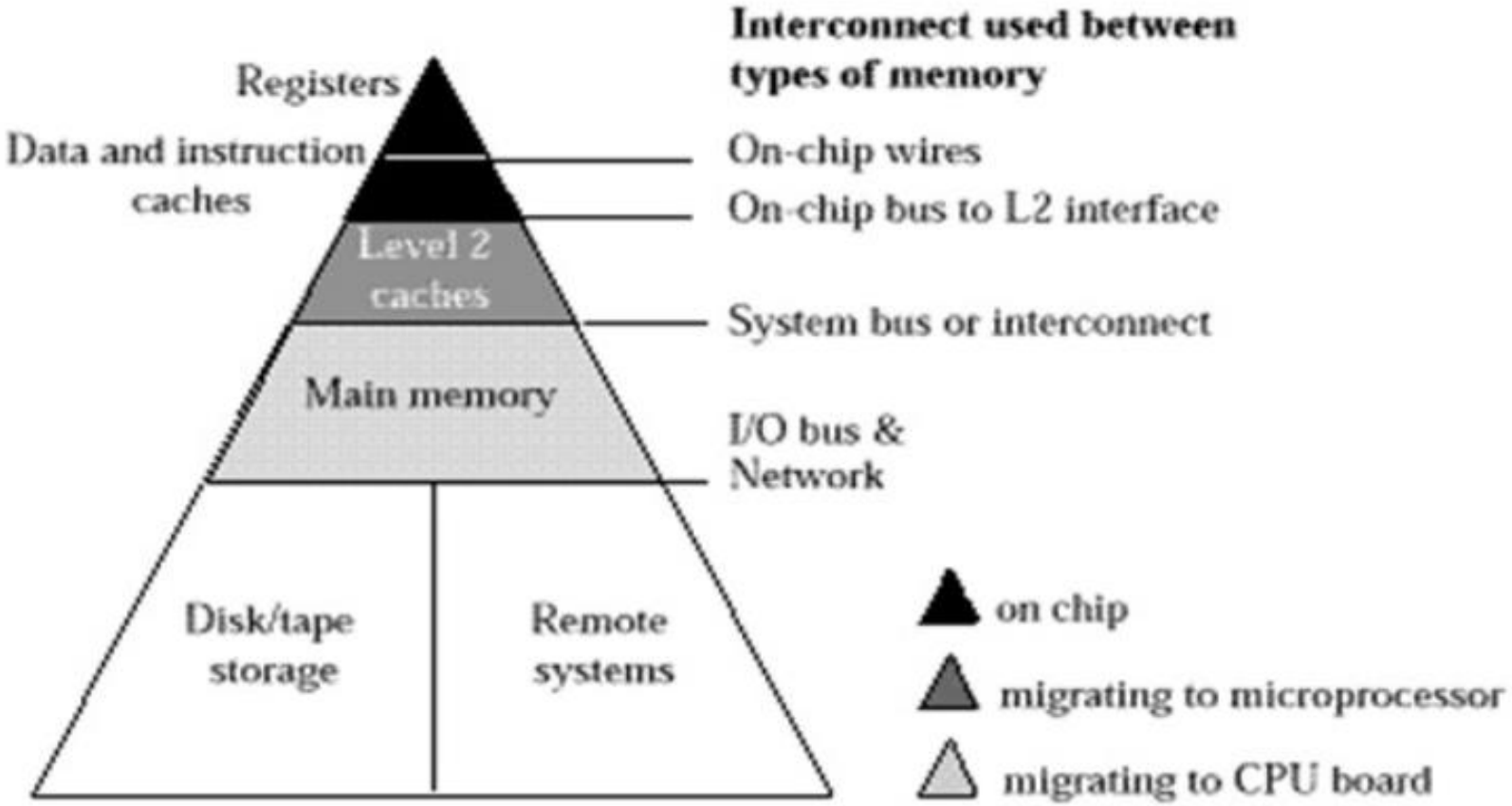


Memory Hierarchy



Memory Hierarchy

Memory Hierarchy



Semiconductor RAM

•RAM (Random Access Memory) is a part of computer's Main Memory which is directly accessible by CPU.

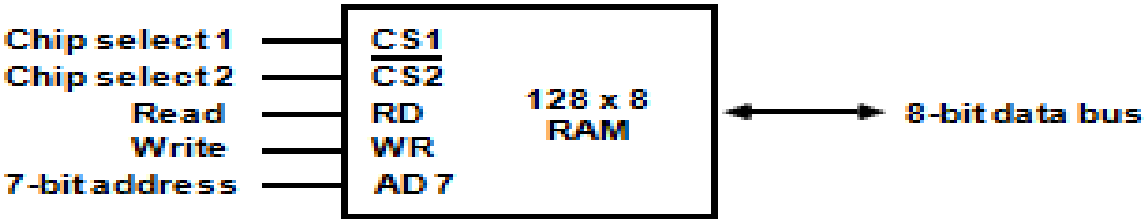
•RAM is used to Read and Write data into it which is accessed by CPU randomly. RAM is volatile in nature.

• RAM is used to store the data that is currently processed by the CPU.

Integrated RAM chips are available in two form:
SRAM(Static RAM)
DRAM(Dynamic RAM)

Semiconductor RAM

The block diagram of RAM chip is given below.



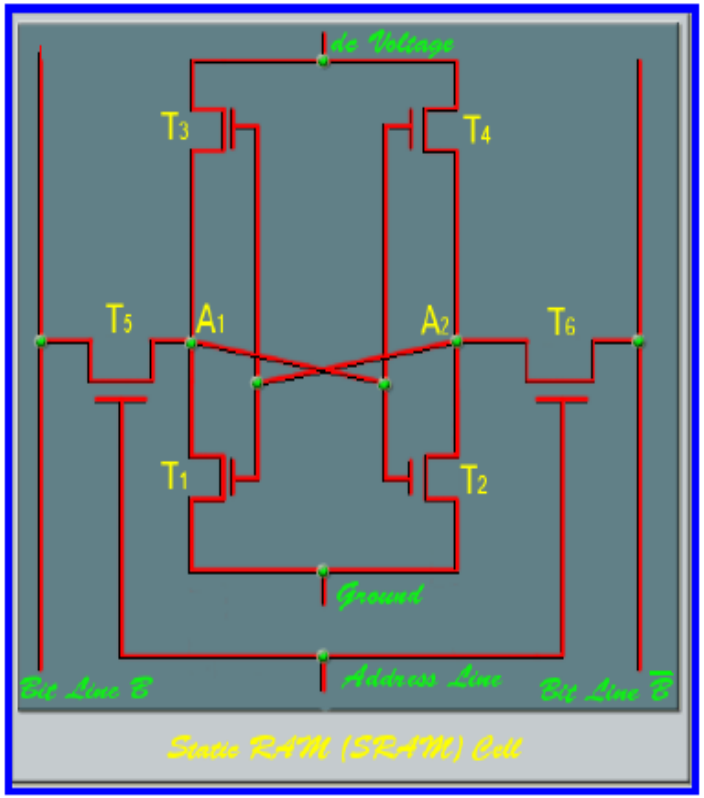
CS1	$\overline{CS2}$	RD	WR	Memory function	State of data bus
0	0	x	x	Inhibit	High-impedence
0	1	x	x	Inhibit	High-impedence
1	0	0	0	Inhibit	High-impedence
1	0	0	1	Write	Input data to RAM
1	0	1	x	Read	Output data from RAM
1	1	x	x	Inhibit	High-impedence

SRAM

Static RAM (SRAM):

In an SRAM, binary values are stored using traditional flip-flop constructed with the help of transistors. A static RAM will hold its data as long as power is supplied to it.

A typical SRAM constructed with transistors is shown in the figure.



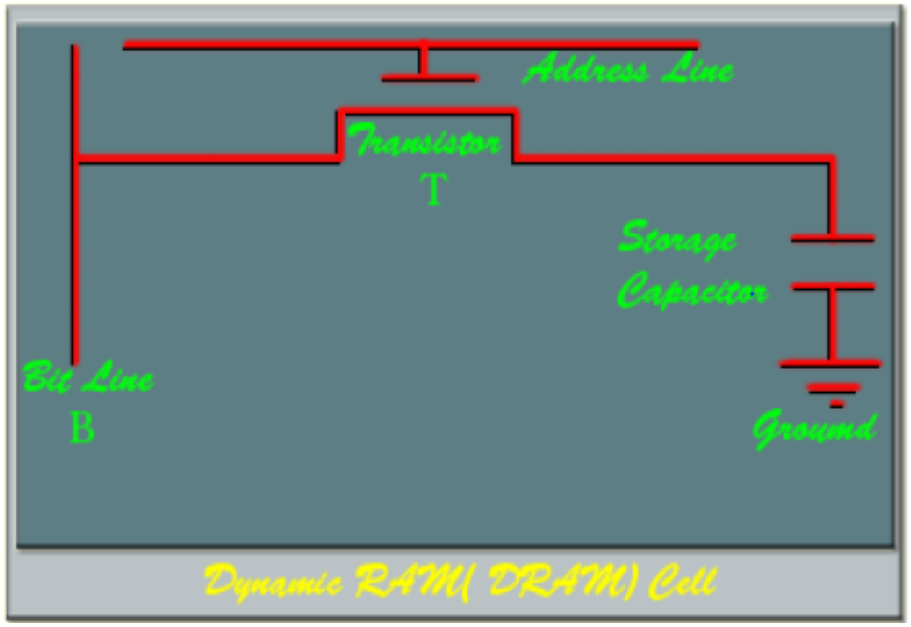
DRAM

Dynamic Ram (DRAM):

A DRAM is made with cells that store data as charge on capacitors. The presence or absence of charge in a capacitor is interpreted as binary 1 or 0.

Because capacitors have a natural tendency to discharge due to leakage current, dynamic RAM require periodic charge refreshing to maintain data storage. The term dynamic refers to this tendency of the stored charge to leak away, even with power continuously applied.

A typical DRAM structure for an individual cell that stores one bit information is shown in the figure.



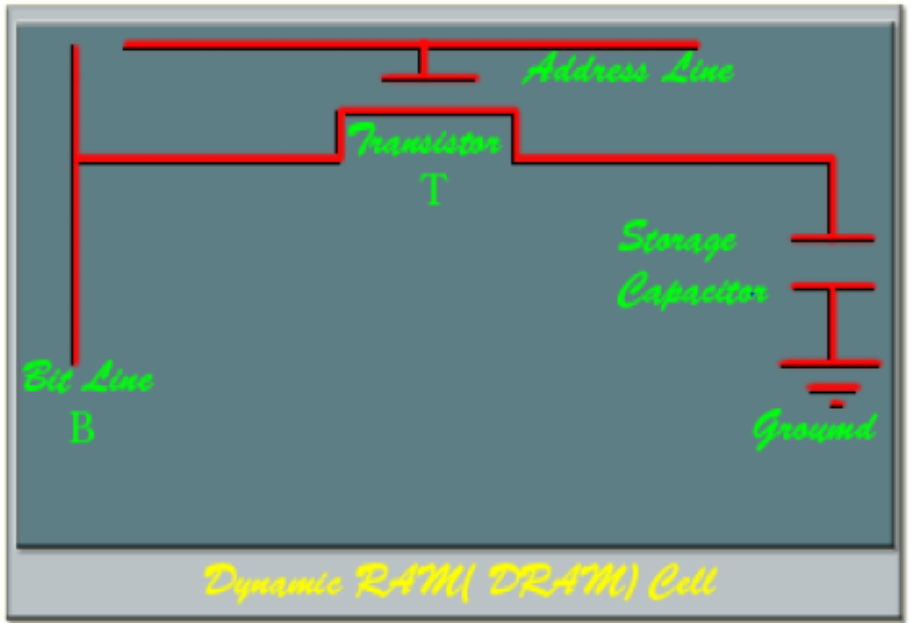
DRAM

Dynamic Ram (DRAM):

A DRAM is made with cells that store data as charge on capacitors. The presence or absence of charge in a capacitor is interpreted as binary 1 or 0.

Because capacitors have a natural tendency to discharge due to leakage current, dynamic RAM require periodic charge refreshing to maintain data storage. The term dynamic refers to this tendency of the stored charge to leak away, even with power continuously applied.

A typical DRAM structure for an individual cell that stores one bit information is shown in the figure.



SRAM Vs. DRAM

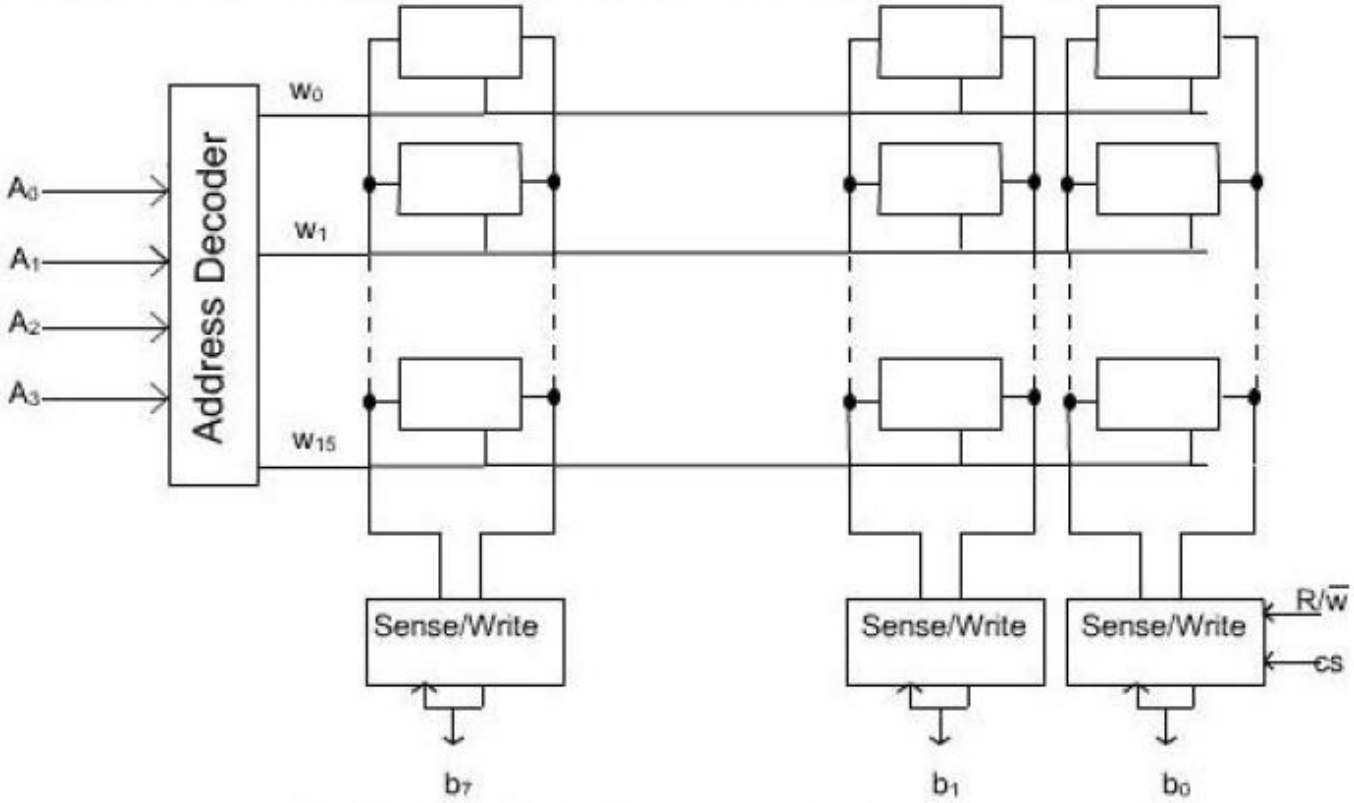
The block diagram of RAM chip is given below.

SRAM	DRAM
1. SRAM has lower access time, so it is faster compared to DRAM.	1. DRAM has higher access time, so it is slower than SRAM.
2. SRAM is costlier than DRAM.	2. DRAM costs less compared to SRAM.
3. SRAM requires constant power supply, which means this type of memory consumes more power.	3. DRAM offers reduced power consumption, due to the fact that the information is stored in the capacitor.
4. Due to complex internal circuitry, less storage capacity is available compared to the same physical size of DRAM memory chip.	4. Due to the small internal circuitry in the one-bit memory cell of DRAM, the large storage capacity is available.
5. SRAM has low packaging density.	5. DRAM has high packaging density.

Organization of Memory Chips

Internal Organization of Memory Chips

A memory cell is capable of storing 1-bit of information. A number of memory cells are organized in the form of a matrix to form the memory chip. One such organization is shown in the Figure.



16 memory location $w_0, w_1, \ldots\ldots\ldots\ldots w_{15}$

8 bits in each location $b_0, b_1, \ldots\ldots\ldots\ldots b_7$

Figure : 16 X 8 Memory Organization

2D and 2.5D Memory organization

2D Memory organization

In 2D organization memory is divided in the form of rows and columns. Each row contains a word. Now in this memory organization there is a decoder.

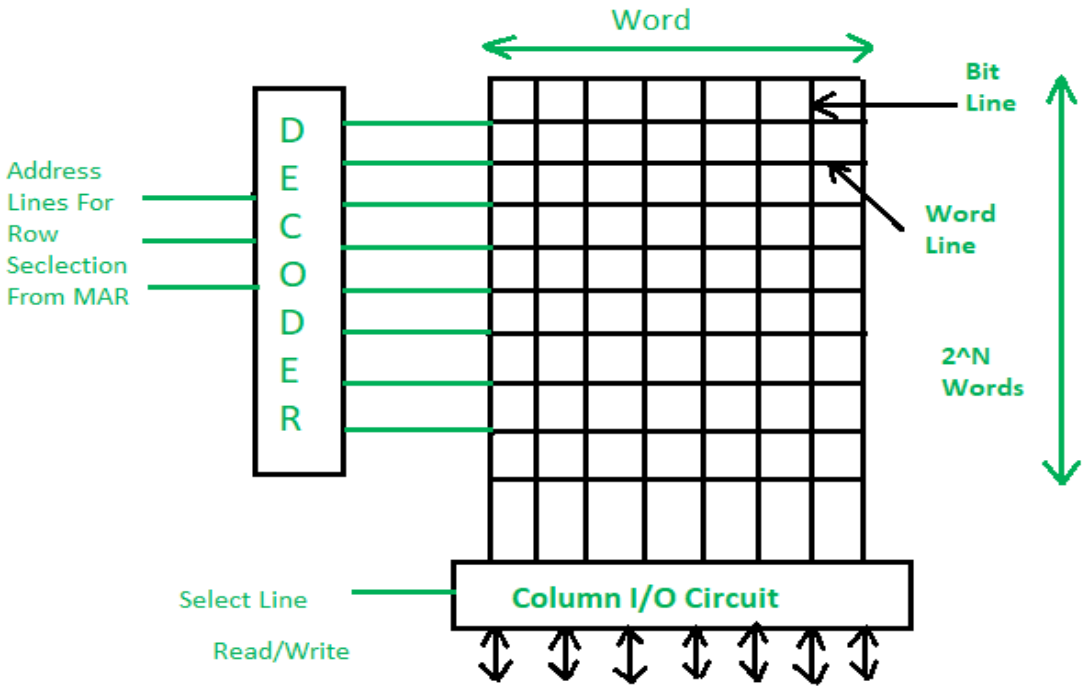
A decoder is a combinational circuit which contains n input lines and 2^n output lines.

One of the output lines will select the row which address is contained in the MAR.

The word which is represented by the row that will get selected and either read or write through the data lines.

2D and 2.5D Memory organization

2D Memory organization



2D Memory Organization

2D and 2.5D Memory organization

2.5 D Memory organization

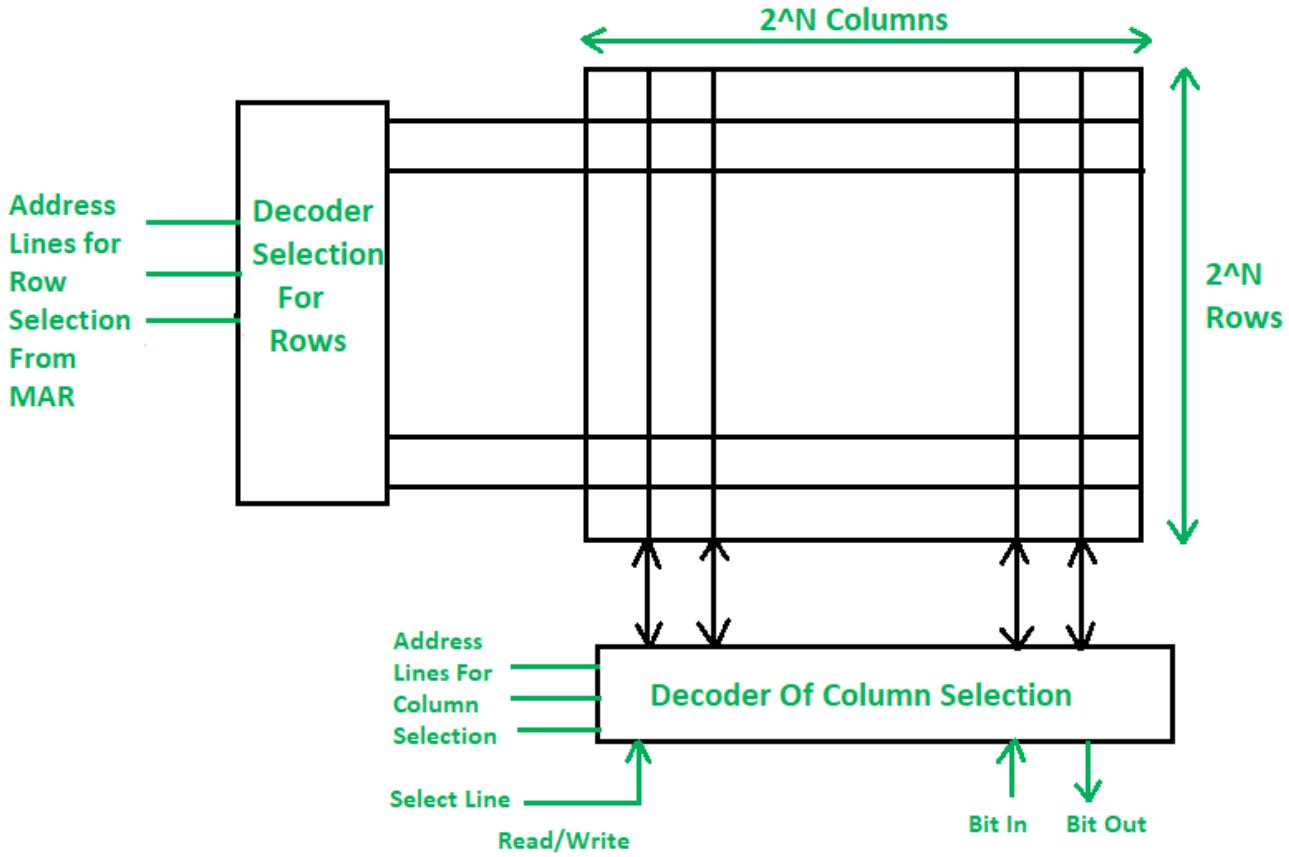
In 2.5D Organization the scenario is the same but we have two different decoders one is column decoder and another is row decoder.

Column decoder used to select the column and row decoder is used to select the row. Address from the MAR will go in decoders' input.

Decoders will select the respective cell. Through the bit outline, the data from that location will be read or through the bit in line data will be written at that memory location.

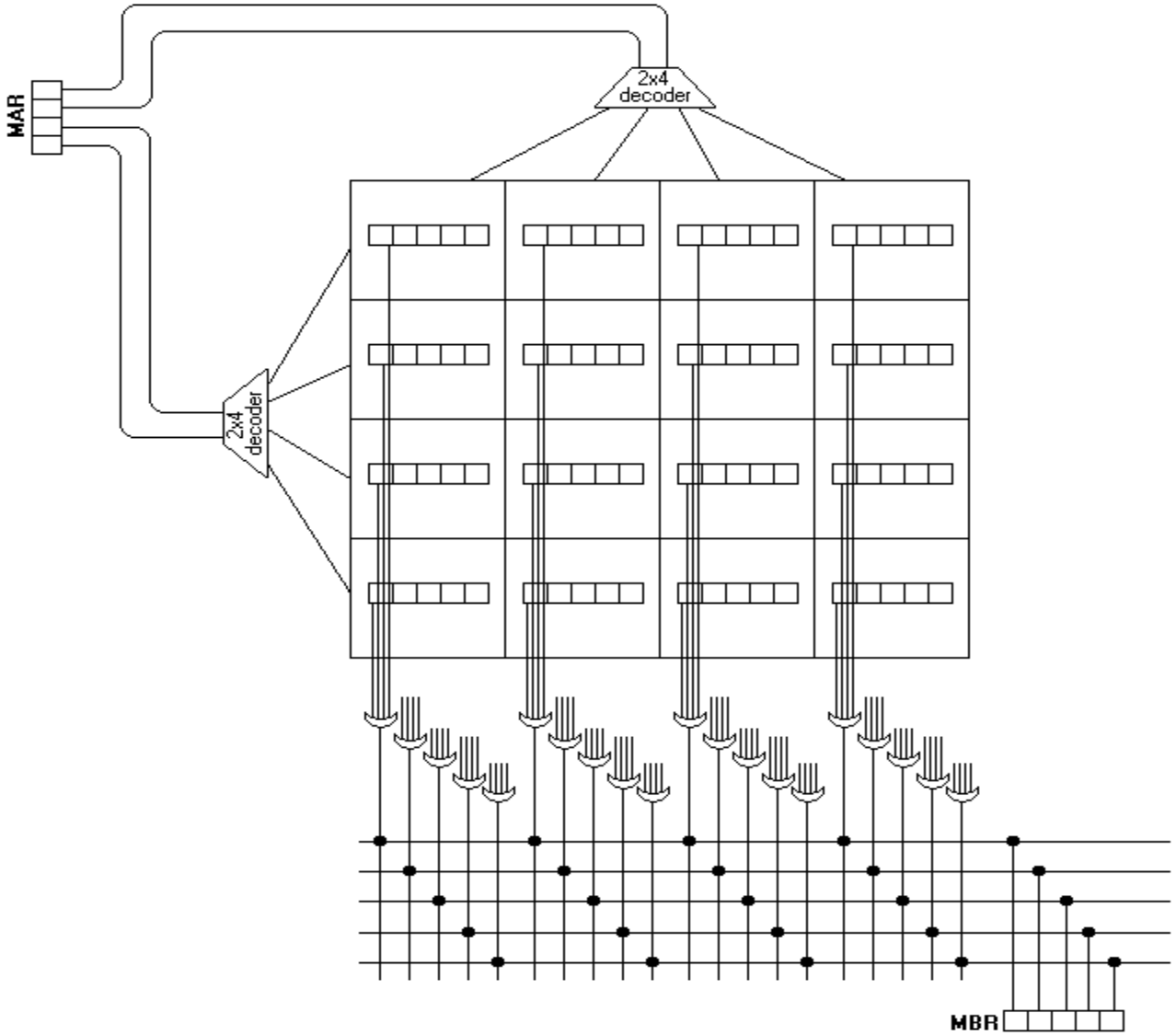
2D and 2.5D Memory organization

2.5D Memory organization



2.5D Memory Organization

2D and 2.5D Memory organization



2D and 2.5D Memory organization

Read and Write Operations

•If the select line is in Read mode then the Word/bit which is represented by the MAR that will be coming out to the data lines and get read.

•If the select line is in write mode then the data from memory data register (MDR) will go to the respective cell which is addressed by the memory address register (MAR).

•With the help of the select line the data will get selected where the read and write operations will take place.

•

2D and 2.5D Memory organization

Comparison between 2D & 2.5D Organizations

- In 2D organization hardware is fixed but in 2.5D hardware changes.
- 2D Organization requires more no. of Gates while 2.5D requires less no. of Gates.
- 2D is more complex in comparison to the 2.5D Organization.
- Error correction is not possible in the 2D organization but In 2.5D error correction is easy.
- 2D is more difficult to fabricate in comparison to the 2.5D organization.

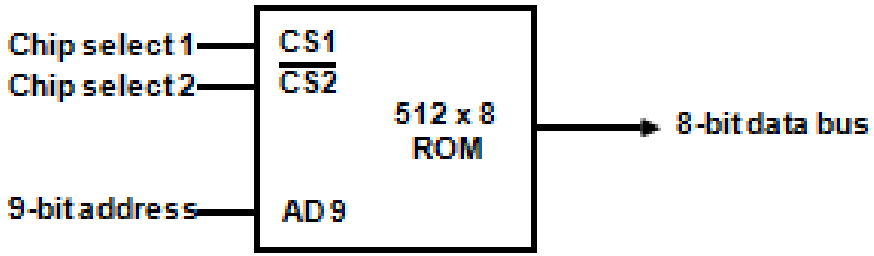
ROM Memory

ROM: Read only memories are **non volatile** in nature. The storage is permanent, but it is read only memory. We can not store new information in ROM.

Several types of ROM are available:

- **PROM:** Programmable Read Only Memory; it can be programmed once as per user requirements.
- **EPROM:** Erasable Programmable Read Only Memory; the contents of the memory can be erased and store new data into the memory. In this case, we have to erase whole information.
- **EEPROM:** Electrically Erasable Programmable Read Only Memory; in this type of memory the contents of a particular location can be changed without effecting the contents of other location.

Typical ROM chip



ROM Memory

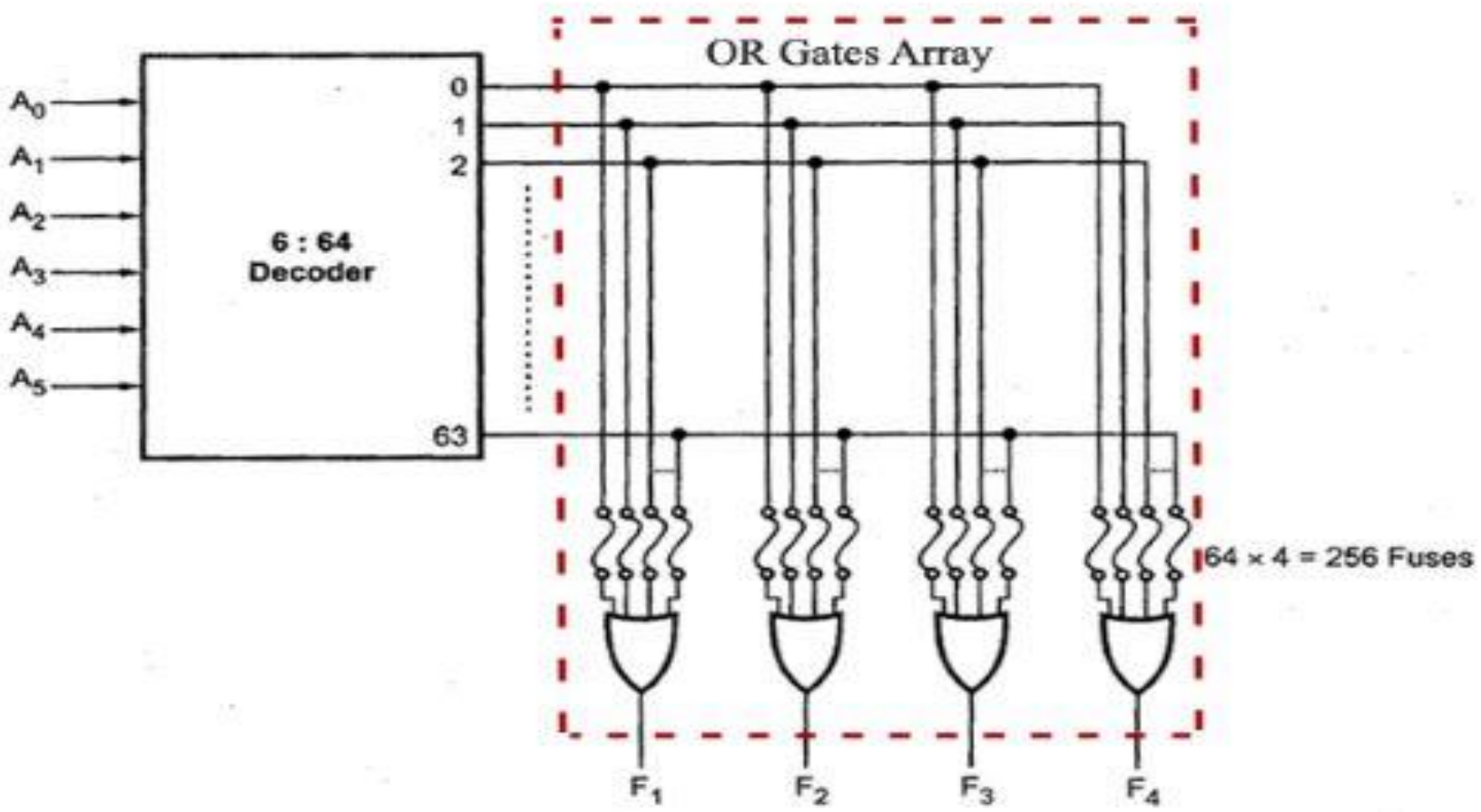
•Internal Structure of ROM:
•The internal structure comprises two basic components: decoder and OR gates.

•A decoder is a circuit that decodes an encoded form (such as binary coded decimal, BCD) to a decimal form. So, the input is in binary form, and the output is its decimal equivalent.

•All the OR gates present in the ROM will have outputs of the decoder as their output.

•Let us take an example of 64 x 4 ROM. The structure is shown in the following image.

ROM Memory



Internal construction of 64x4 ROM

RAM Vs.ROM

DRAM	SRAM
1. Constructed of tiny capacitors that leak electricity.	1.Constructed of circuits similar to D flip-flops.
2.Requires a recharge every few milliseconds to maintain its data.	2.Holds its contents as long as power is available.
3.Inexpensive.	3.Expensive.
4. Slower than SRAM.	4. Faster than DRAM.
5. Can store many bits per chip.	5. Can not store many bits per chip.
6. Uses less power.	6.Uses more power.
7.Generates less heat.	7.Generates more heat.
8. Used for main memory.	8. Used for cache.

Difference between SRAM and DRAM

Associative Memory

An associative memory can be considered as a memory unit whose stored data can be identified for access by the content of the data itself rather than by an address or memory location.

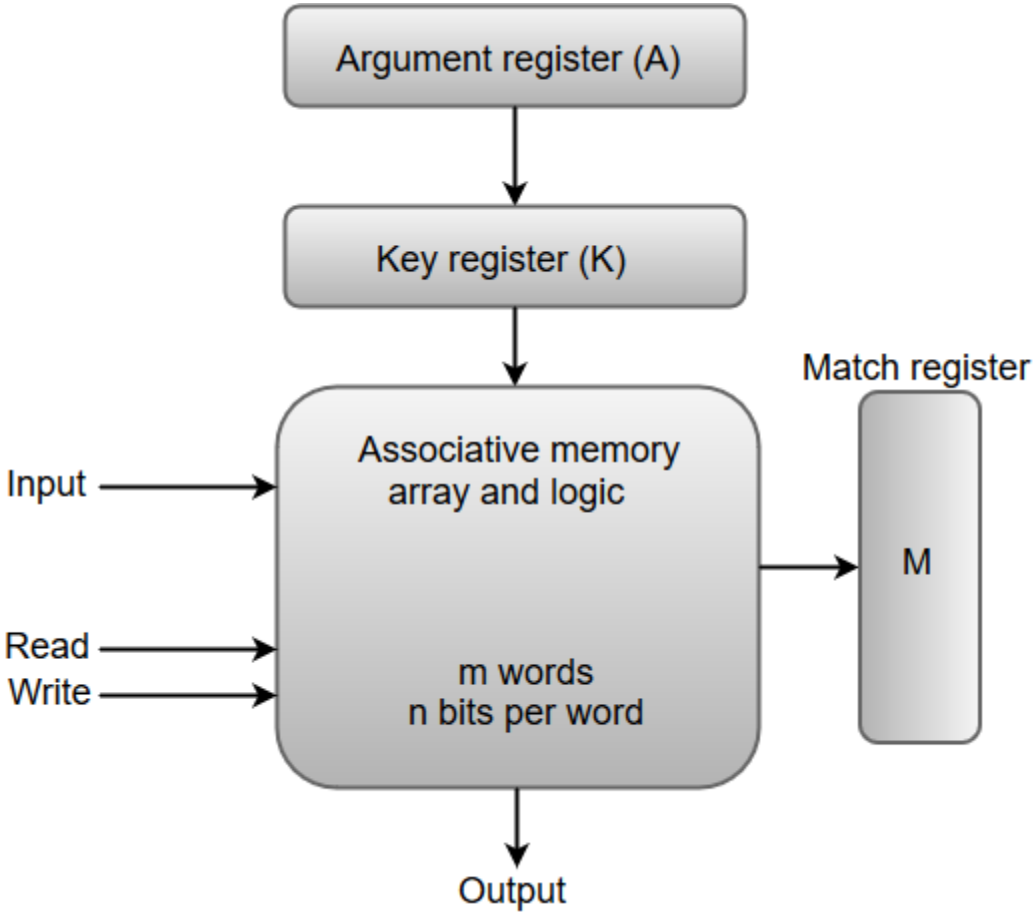
Associative memory is often referred to as **Content Addressable Memory (CAM)**.

When a write operation is performed on associative memory, no address or memory location is given to the word. The memory itself is capable of finding an empty unused location to store the word.

On the other hand, when the word is to be read from an associative memory, the content of the word, or part of the word, is specified. The words which match the specified content are located by the memory and are marked for reading.

The following diagram shows the block representation of an Associative memory

Associative Memory



Associative Memory

From the block diagram, we can say that an associative memory consists of a memory array and logic for 'm' words with 'n' bits per word.

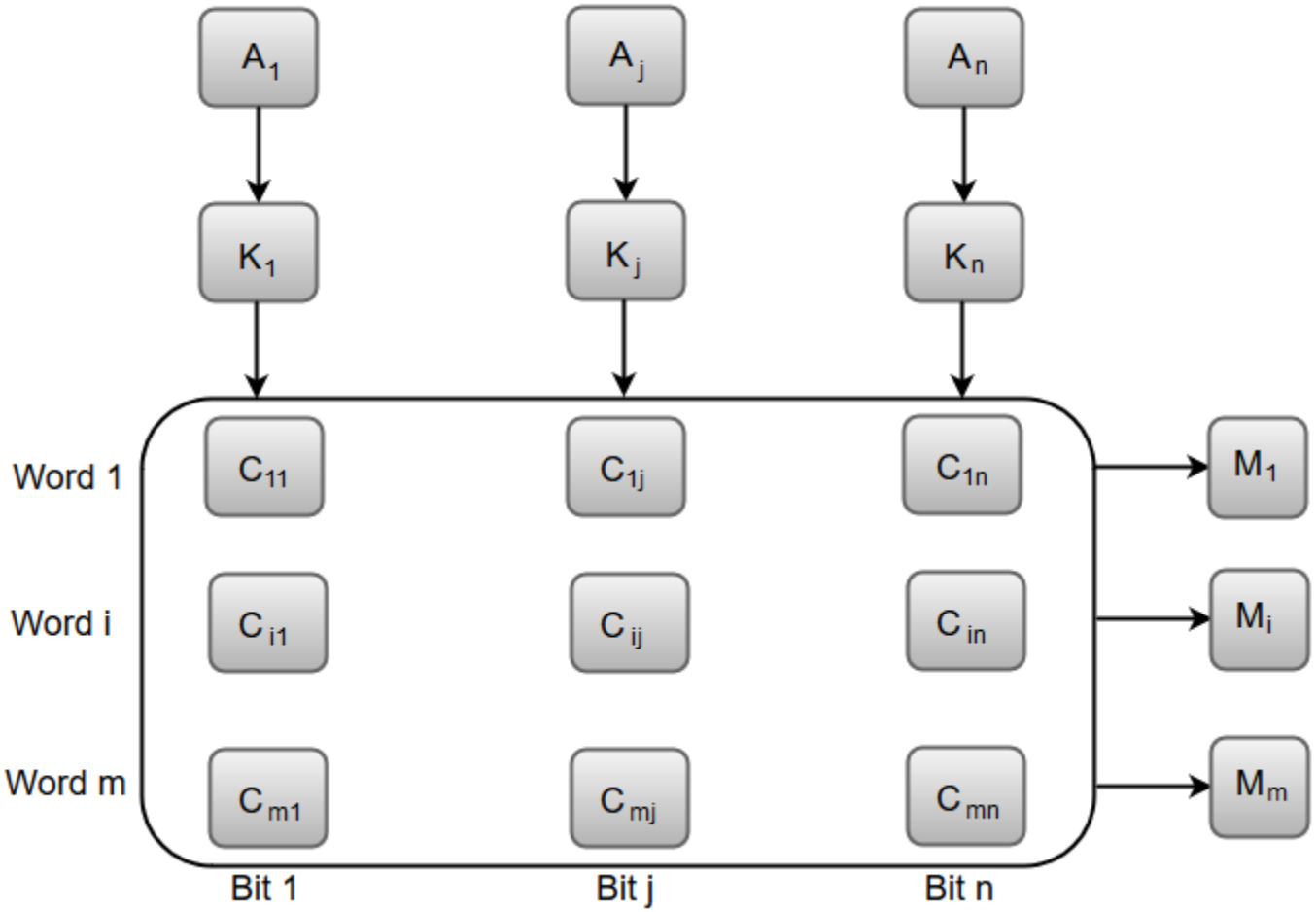
The functional registers like the argument register **A** and key register **K** each have **n** bits, one for each bit of a word. The match register **M** consists of **m** bits, one for each memory word.

The words which are kept in the memory are compared in parallel with the content of the argument register.

The key register (K) provides a mask for choosing a particular field or key in the argument word. If the key register contains a binary value of all 1's, then the entire argument is compared with each memory word. Otherwise, only those bits in the argument that have 1's in their corresponding position of the key register are compared. Thus, the key provides a mask for identifying a piece of information which specifies how the reference to memory is made.

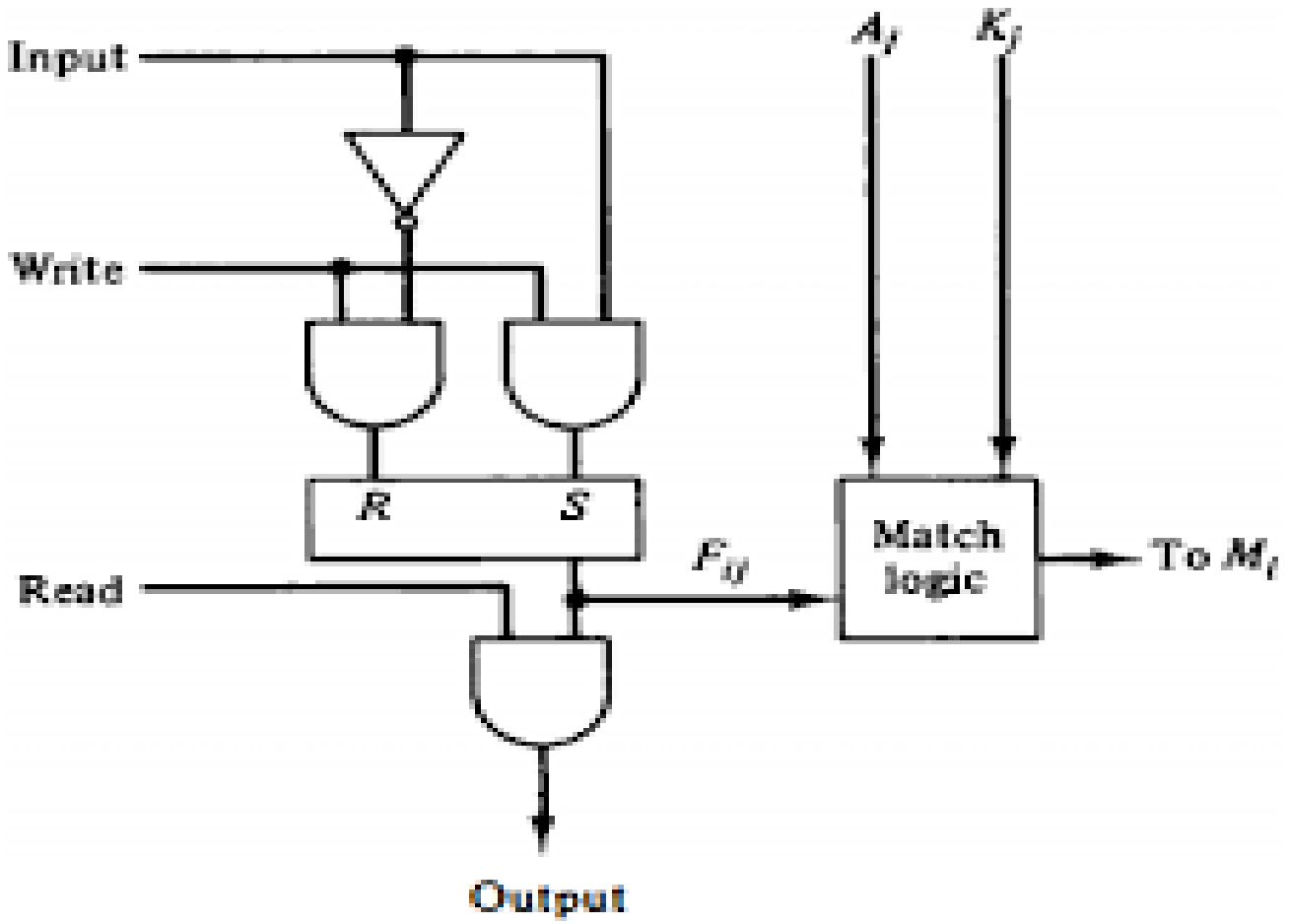
Associative Memory

Associative memory of m word, n cells per word:



Associative Memory

Figure 8 One cell of associative memory.



Cache Memory

•**Cache Memory** is a special very high-speed and small memory. It is used to speed up and synchronizing with high-speed CPU.

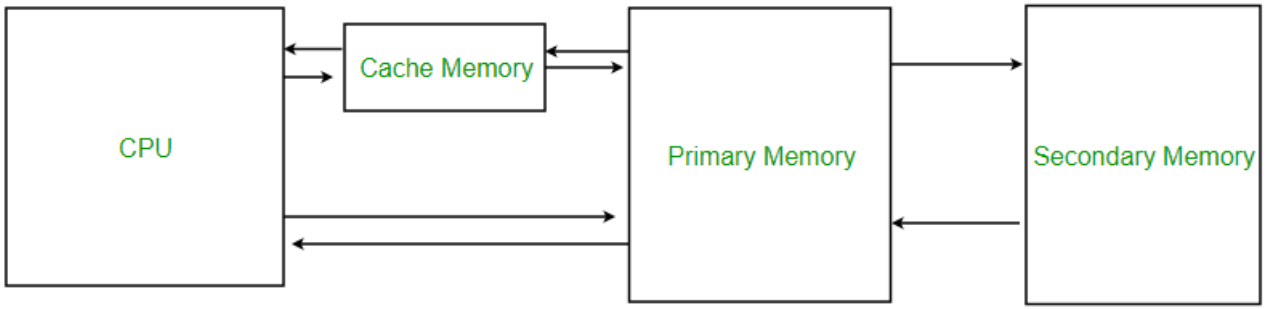
•Cache memory is costlier than main memory or disk memory but economical than CPU registers.

•Cache memory acts as a buffer between RAM and the CPU. It holds frequently requested data and instructions so that they are immediately available to the CPU when needed.

•Cache memory is used to reduce the average time to access data from the Main memory.

•The cache stores copies of the data from frequently used main memory locations.
•There are various different independent caches in a CPU, which store instructions and data.

Cache Memory



Position of cache memory

Cache Performance: When the processor needs to read or write a location in main memory, it first checks for a corresponding entry in the cache.

- If available in the cache, a **cache hit** otherwise **cache miss**

- For a cache miss, the cache allocates a new entry and copies in data from main memory, then the request is fulfilled from the contents of the cache.

Hit ratio = hit / (hit + miss)

Cache Memory

Types of Cache

There are two types of cache memory-

- **Primary Cache**

A primary cache is always located on the processor chip. This cache is small and its access time is comparable to that of processor registers. It is referred to as the level 1 (L1) cache.

- **Secondary Cache**

Secondary cache is placed between the primary cache and the rest of the memory. It is referred to as the level 2 (L2) cache. Often, the Level 2 cache is also housed on the processor chip.

Cache Memory

Locality of reference

Since size of cache memory is less as compared to main memory. So to check which part of main memory should be given priority and loaded in cache is decided based **on locality of reference**.

- Temporal Locality
  - A memory location that is referenced is likely to be accessed again in the near future
  - Data is brought into cache expecting it to be used again
- Spatial Locality
  - Memory locations near the last access are likely to be accessed in the near future
  - Lines are copied into cache so adjacent addresses are available

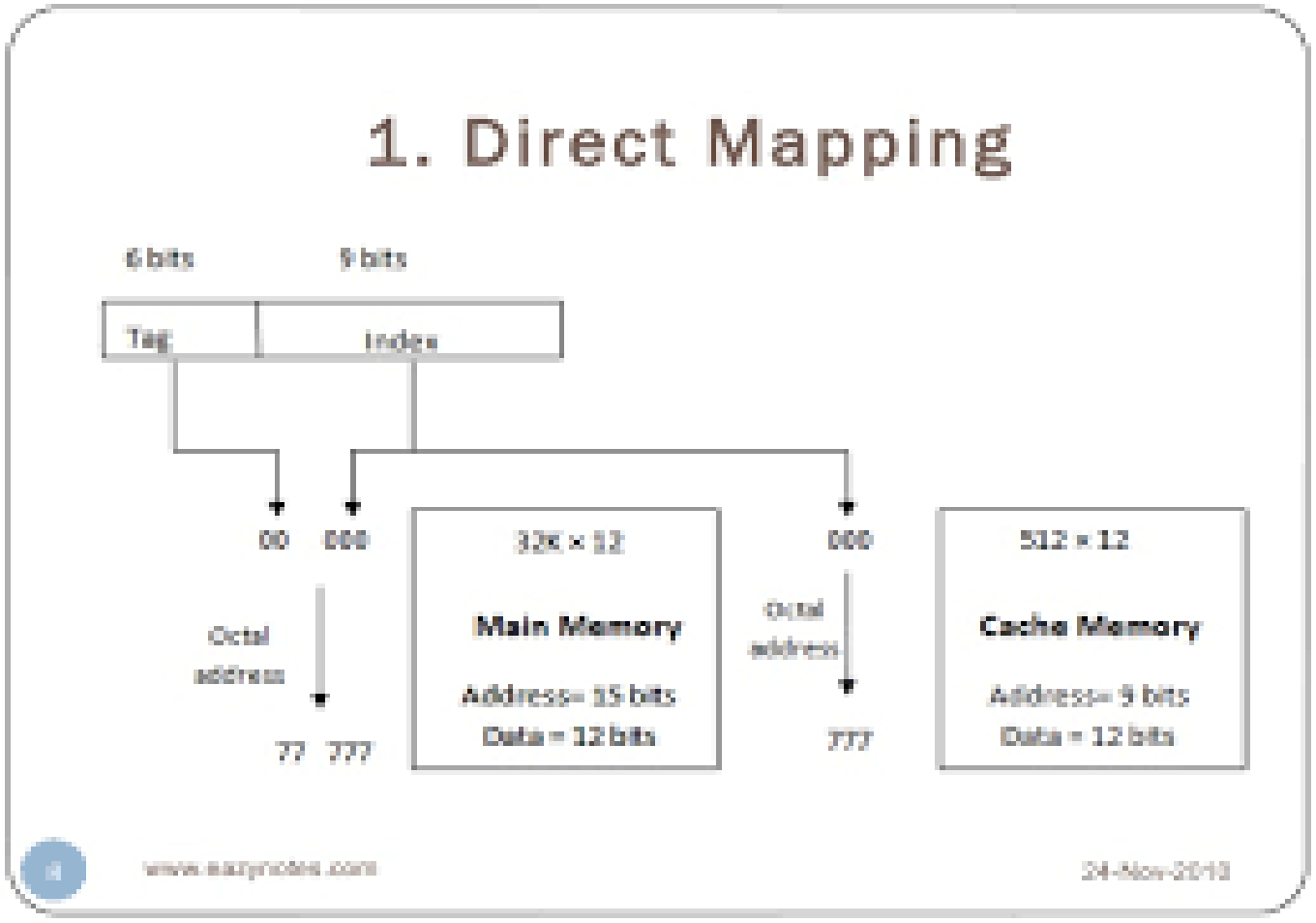
Cache Memory Mapping

Cache Mapping: The transformation of data from main memory to cache memory is referred to as a mapping process

There are three different types of mapping used for the purpose of cache memory which are as follows:

1. **Direct mapping**
2. **Associative mapping**
3. **Set-Associative mapping**.

Cache Memory Mapping



Cache Memory Mapping

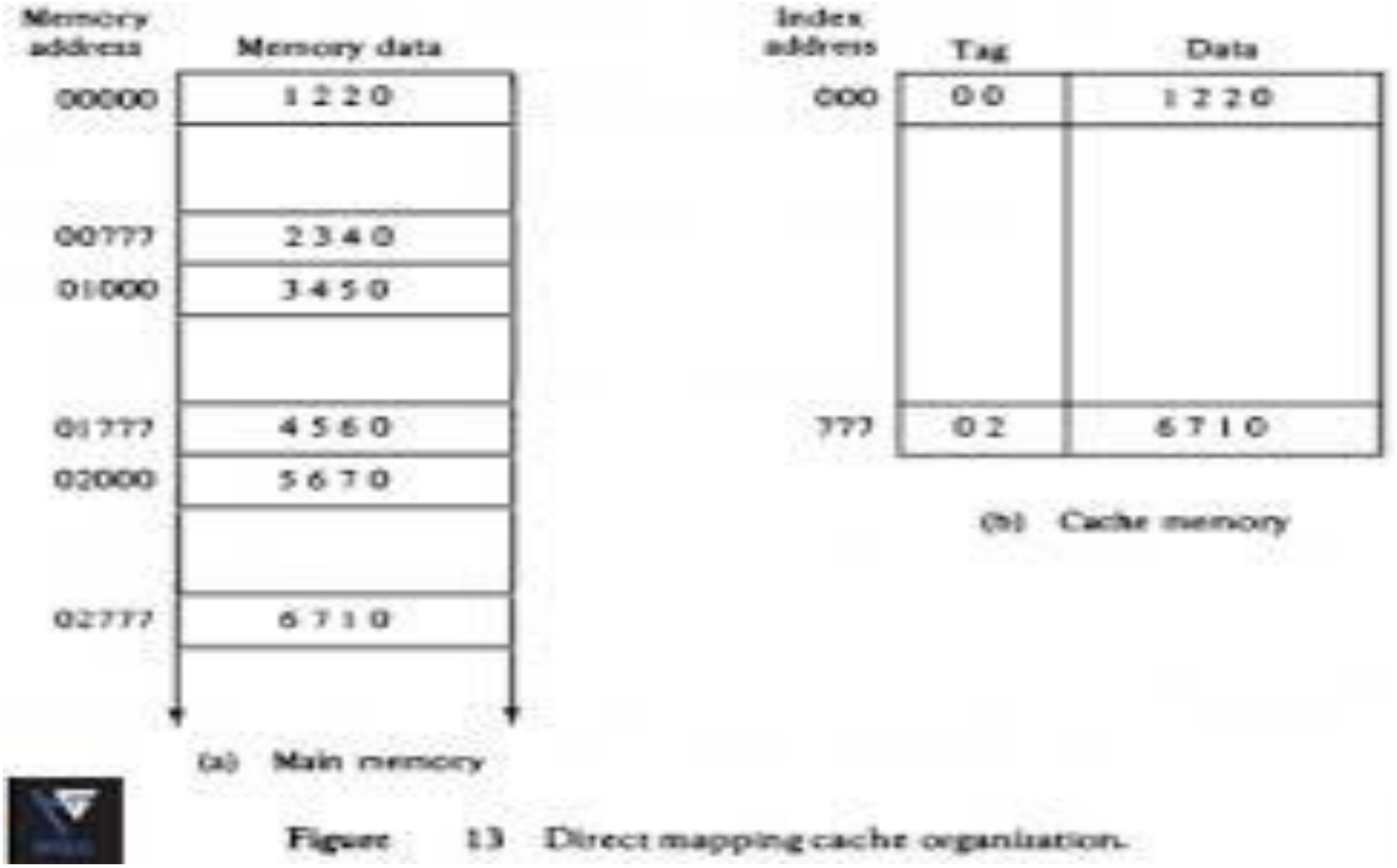
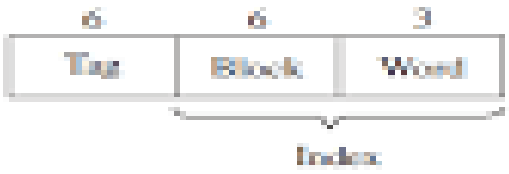


Figure 13 Direct mapping cache organization.

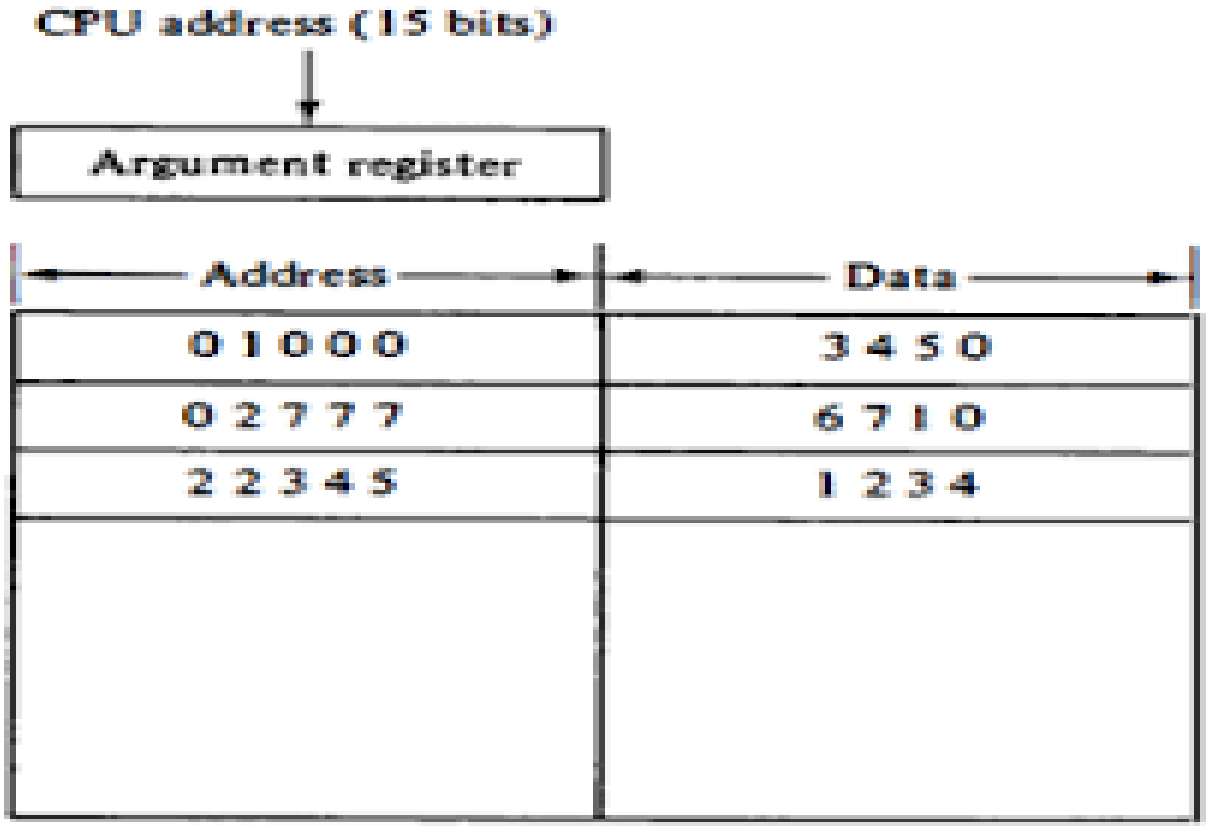
Cache Memory Mapping

	Index	Tag	Data
Block 0	000	01	3450
	007	01	6578
Block 1	010		
	017		
Block 63	770	02	
	777	02	0710



Direct mapping cache with block size of 8 words

Associative Mapping



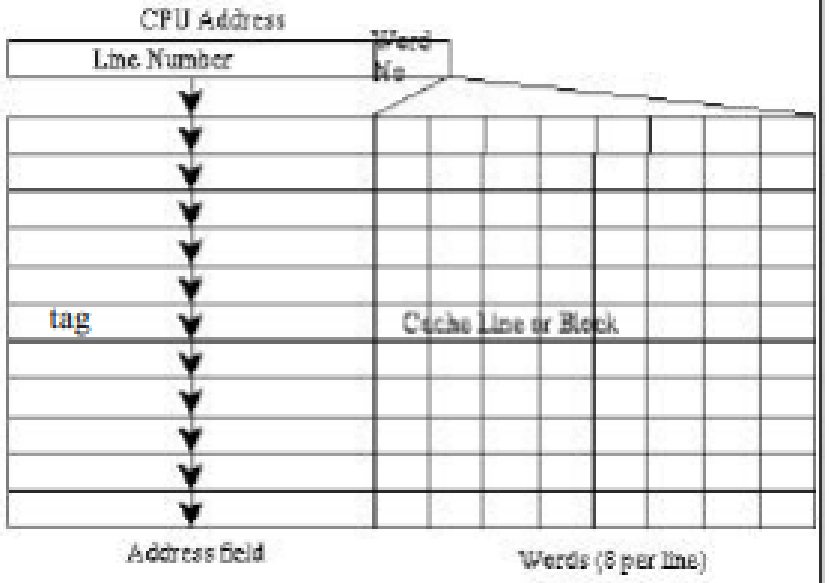
Set Associative Mapping

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

Cache Memory Mapping

Fully associative caches

- Cache line contains more than one word (8 here)
- CPU address must identify
  - which (if any) line has the word
  - which word is required
- Most significant part of address identifies the cache line
- less significant part identifies the word in the line
  - here, 29, 3 bits

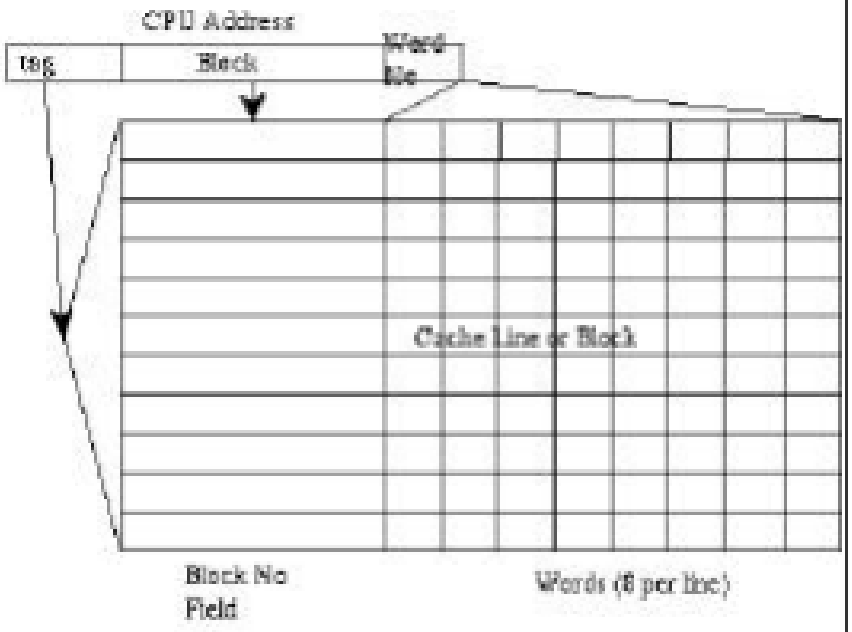


- An associative memory search identifies which if any line holds the address
  - all the cache block address *tags* are compared with the CPU address simultaneously
  - this is essential to fast operation

Cache Memory Mapping

Direct mapping caches

- Each block can only be inserted in one place
- No associative mapping
- CPU address is in 3 parts:
  - tag identifies exactly 1 cache block
  - block no is compared with the block no field in the cache line
  - word (least significant) identifies the word in the cache line

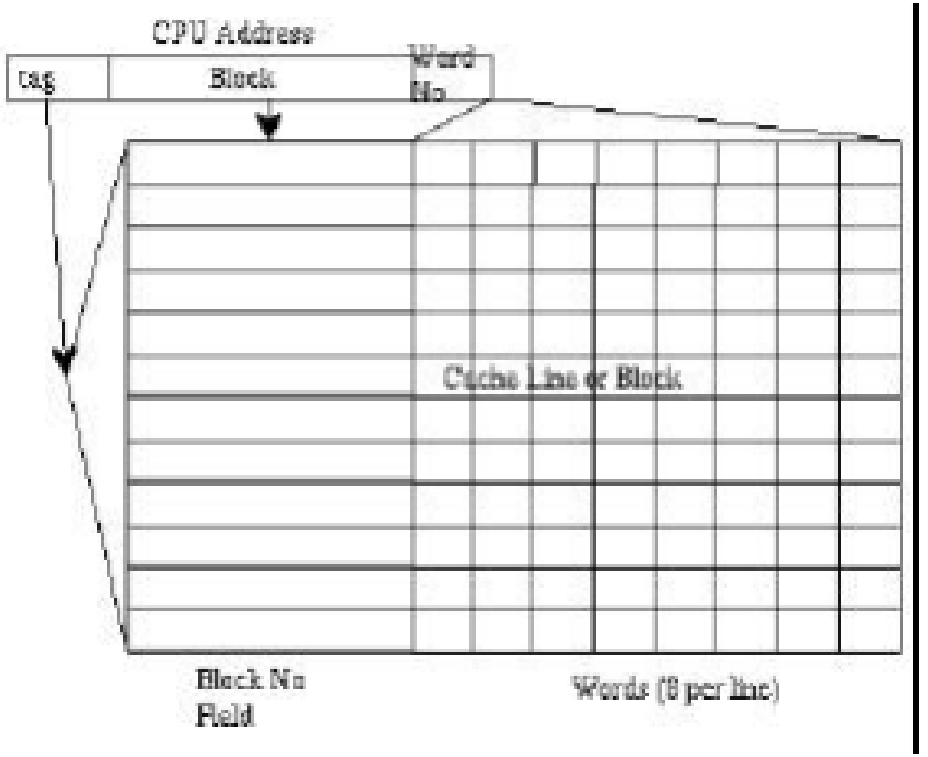


- For a 128-line cache tag field would be 7 bits,
- block field 22 bits, and word field 3 bits.
- Simpler hardware then associative cache

Cache Memory Mapping

Set-Associative Cache

- 2-way set-associative cache
- Tag address selects one out of 2 possible lines
- Block number is associatively compared with these 2 block ids
- For a 128 line cache, we have a 6-bit (not 7) tag field
- Block field is 23 bits
- This is a compromise between direct and fully associative caches.



Virtual Memory

Virtual Memory

- Almost all modern processors support *virtual memory*
- Virtual memory allows a program to treat its memory space as single contiguous block that may be considerably larger than main memory
- A memory management unit takes care of the mapping between virtual and physical addresses

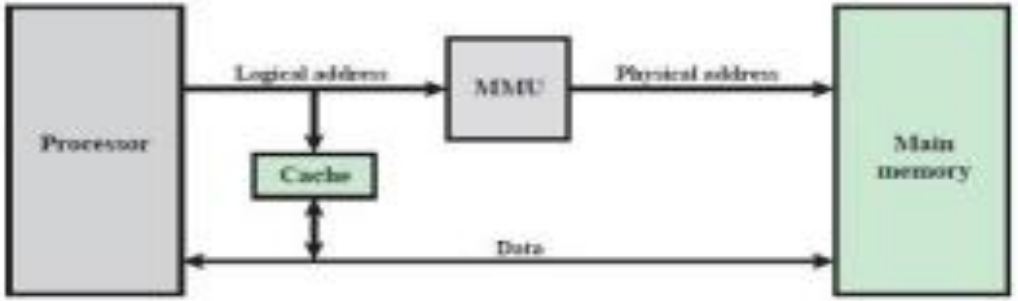
Virtual Memory

Logical Cache

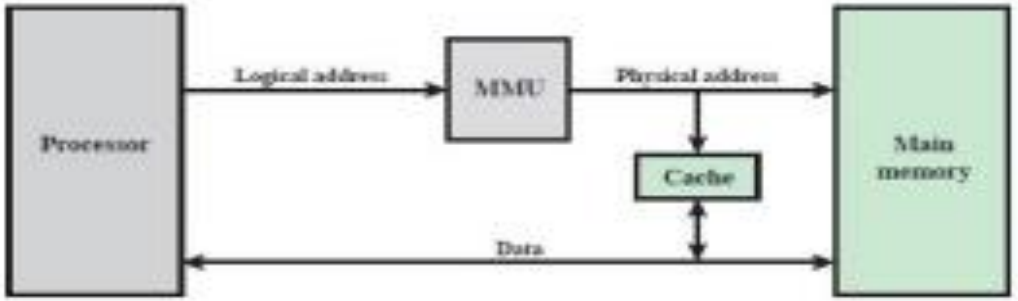
- A logical (virtual) cache stores virtual addresses rather than physical addresses
- Processor addresses cache directly without going through MMU
- Obvious advantage is that addresses do not have to be translated by the MMU
- A not-so-obvious disadvantage is that all processes have the same virtual address space - a block of memory starting at 0
  - The same virtual address in two processes usually refers to different physical addresses
  - So either flush cache with every context switch or add extra bits

Virtual Memory

Logical and Physical Cache

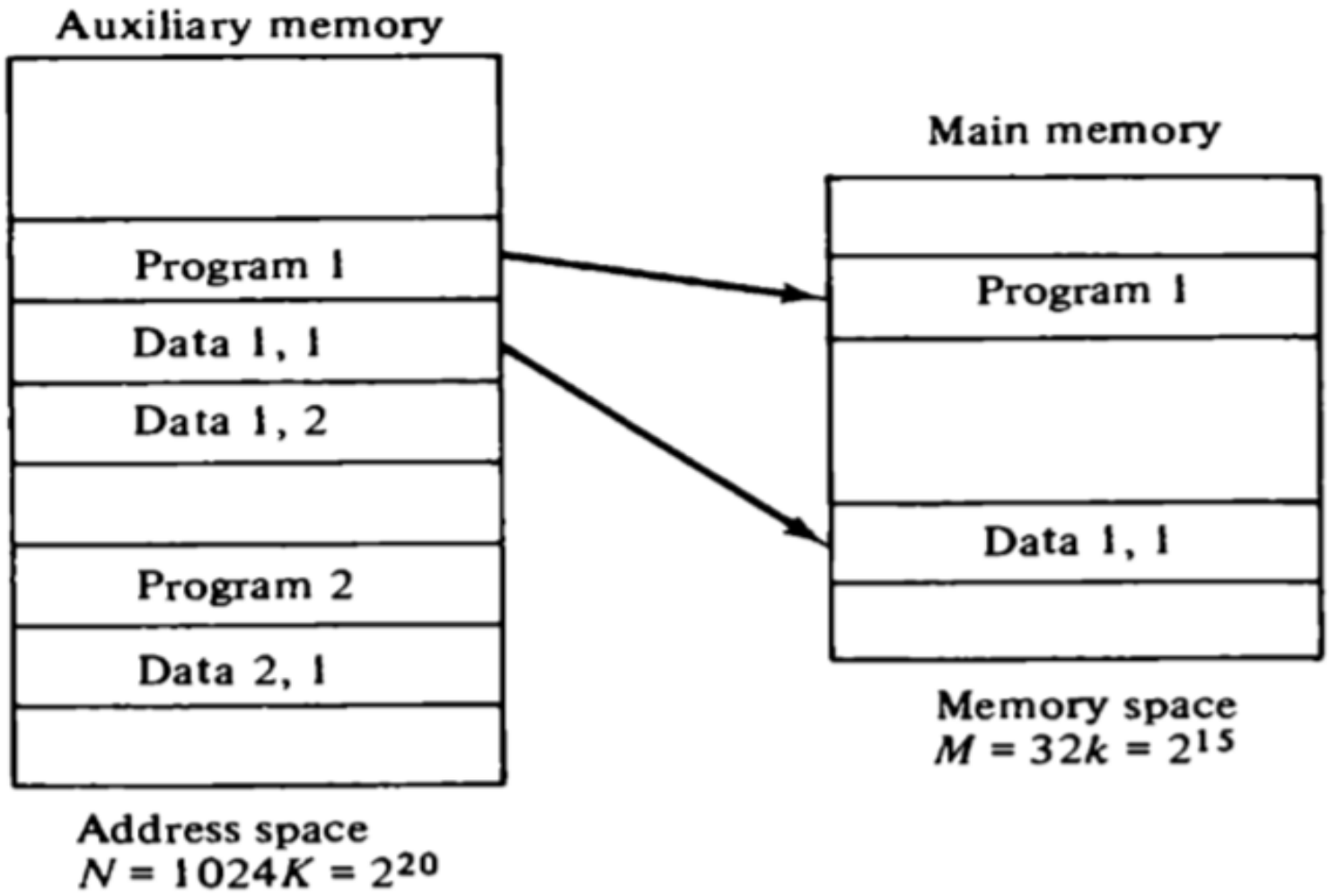


(a) Logical Cache



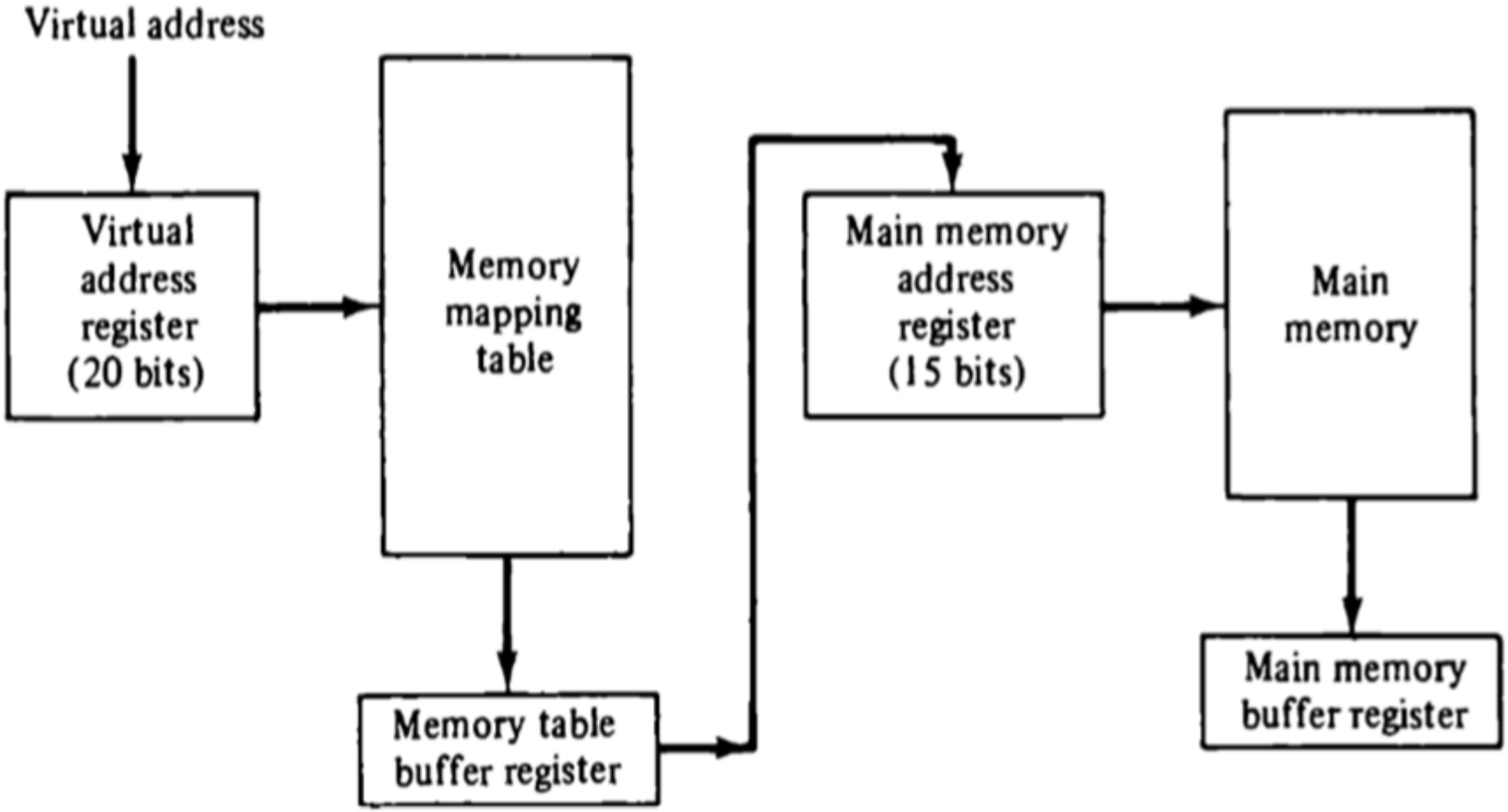
(b) Physical Cache

Virtual Memory



Virtual Memory

Memory table for mapping a virtual address.



Virtual Memory

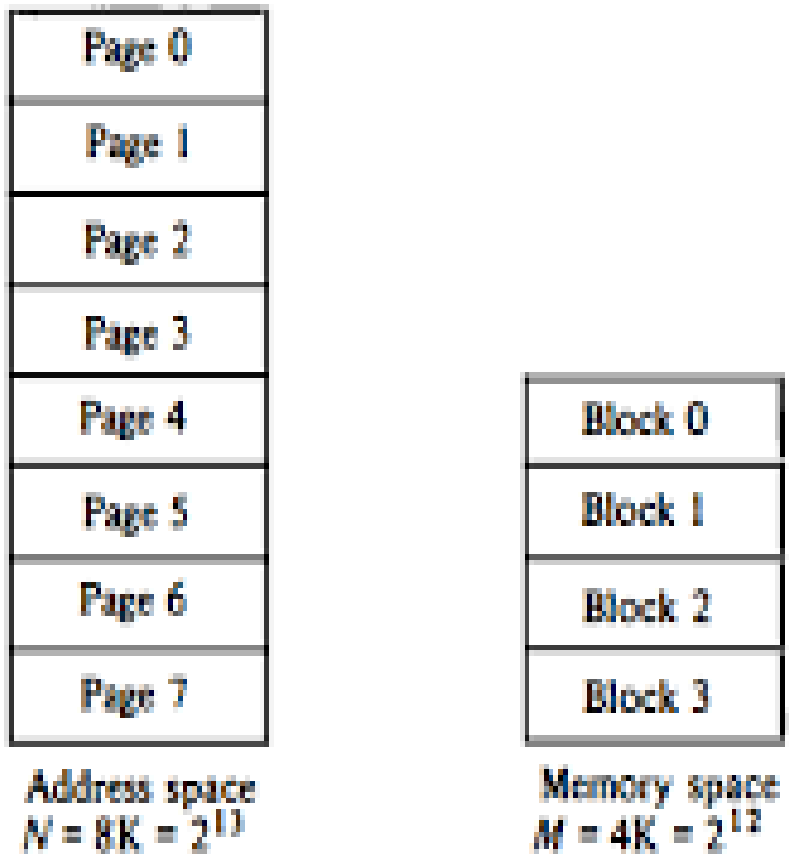


Figure 18 Address space and memory space split into groups of 1K words.

Virtual Memory

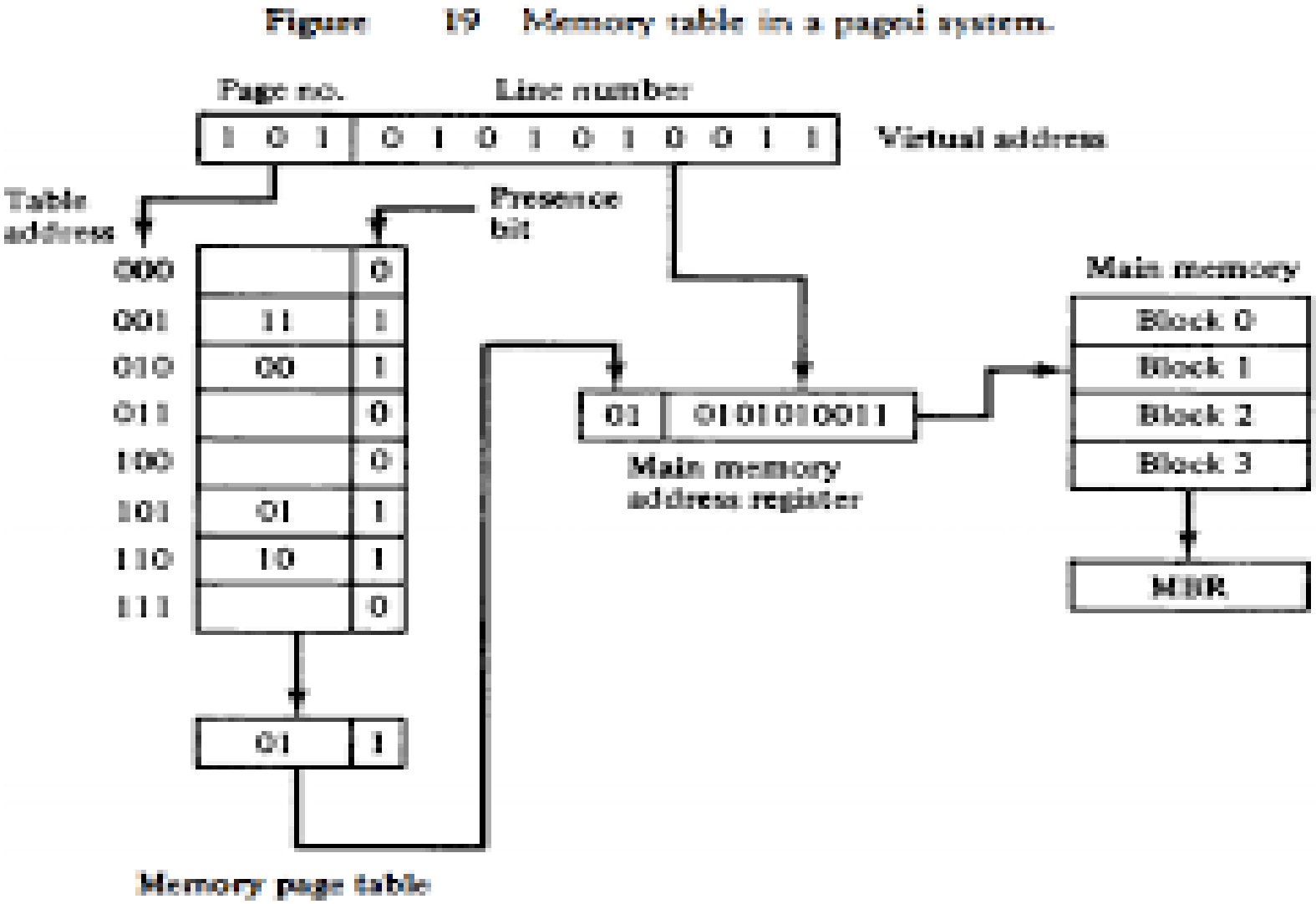
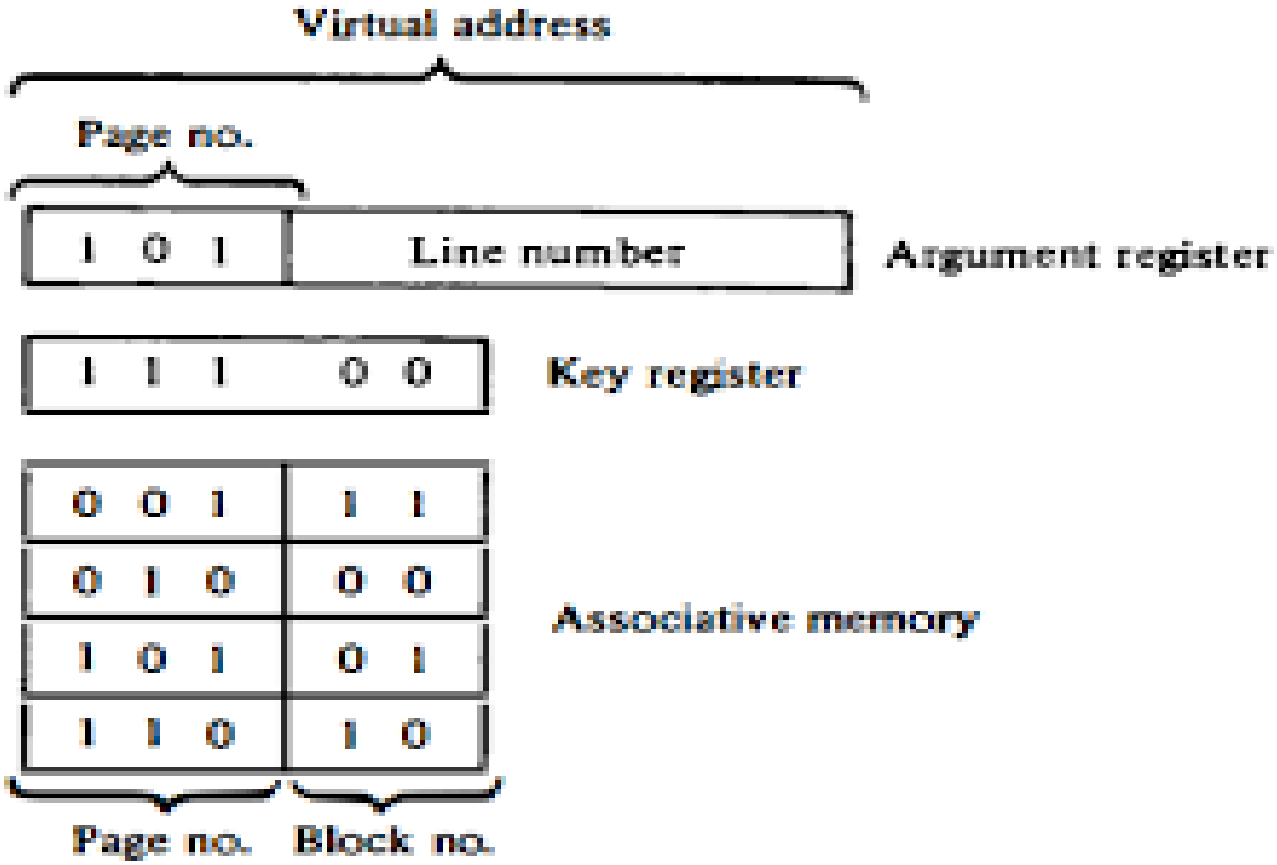


Figure 19 Memory table in a paged system.

Virtual Memory

Figure 20 An associative memory page table.



Page Replacement Algorithm

Page replacement is a process of swapping out an existing page from the frame of a main memory and replacing it with the required page.

Page Replacement Algorithms-

Page replacement algorithms help to decide which page must be swapped out from the main memory to create a room for the incoming page.

FIFO Page Replacement Algorithm
LIFO Page Replacement Algorithm
LRU Page Replacement Algorithm
Optimal Page Replacement Algorithm

A good page replacement algorithm is one that minimizes the number of page faults.

Page Replacement Algorithm

FIFO Page Replacement Algorithm-

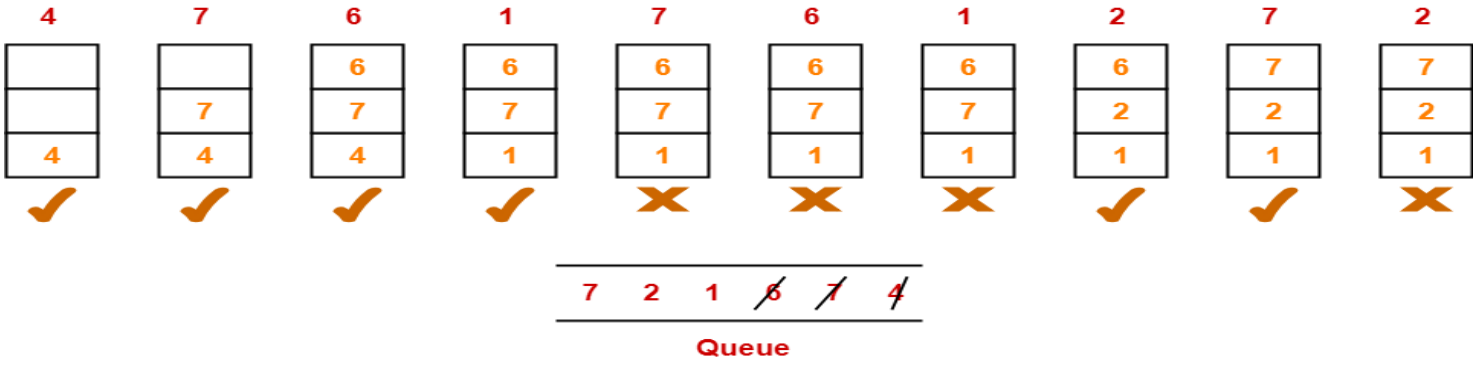
As the name suggests, this algorithm works on the principle of "**First in First out**".
It replaces the oldest page that has been present in the main memory for the longest time. It is implemented by keeping track of all the pages in a queue.

Example
A system uses **3 page frames** for storing process pages in main memory. It uses the **FIFO** page replacement policy. Assume that all the page frames are initially empty. What is the total number of page faults, hit ratio and miss ratio and page reference string given below-
4 , 7, 6, 1, 7, 6, 1, 2, 7, 2

FIFO Replacement Algorithm

Solution-

Total number of references = 10



Total number of page faults occurred = 6

Total number of page hits

= Total number of references – Total number of page misses or page faults

= 10 – 6 = 4

Thus, Hit ratio = 4 / 10 = 0.4 or 40%, Miss ratio = 6 / 10 = 0.6 or 60%

LIFO Page Replacement Algorithm

LIFO Page Replacement Algorithm-

This algorithm works on the principle of "**Last in First out**".

It replaces the newest page that arrived at last in the main memory.

It is implemented by keeping track of all the pages in a stack.

Example
A system uses **3 page frames** for storing process pages in main memory. It uses the **LIFO** page replacement policy. Assume that all the page frames are initially empty. What is the total number of page faults, hit ratio and miss ratio and page reference string given below-
4 , 7, 6, 1, 7, 6, 1, 2, 7, 2

LIFO Page Replacement Algorithm

Solution-

Total number of references = 10

Total number of page faults occurred =

Total number of page hits

= Total number of references – Total number of page misses or page faults

=

Thus, Hit ratio = Miss ratio =

Optimal Replacement Algorithm

Optimal Page Replacement Algorithm-

This algorithm replaces the page that will not be referred by the CPU in future for the longest time.

It is practically impossible to implement this algorithm.

However, it is the best known algorithm and gives the least number of page faults.

Hence, it is used as a performance measure criterion for other algorithms.

Example

A system uses **3 page frames** for storing process pages in main memory. It uses the **Optimal** page replacement policy. Assume that all the page frames are initially empty. What is the total number of page faults, hit ratio and miss ratio and page reference string given below-

4 , 7, 6, 1, 7, 6, 1, 2, 7, 2

Optimal Page Replacement Algorithm

Solution-

Total number of references =

Total number of page faults occurred =

Total number of page hits

= Total number of references – Total number of page misses or page faults

=

Thus, Hit ratio = , Miss ratio =

Auxiliary Memories

Auxiliary memory (also referred to as *secondary storage*) is the non-volatile memory lowest-cost, highest-capacity, and slowest-access storage in a computer system.
It is where programs and data kept for long-term storage or when not in immediate use. It is not directly accessible by the CPU. **For example:** Magnetic disks and tapes, Optical Disk.

Secondary Storage Media
There are the following main types of storage media:

1. Magnetic storage media:

Magnetic media is coated with a magnetic layer which is magnetized in clockwise or anticlockwise directions. When the disk moves, the head interprets the data stored at a specific location in binary 1s and 0s at reading.
Examples: hard disks, floppy disks and magnetic tapes.

Auxiliary Memories

Floppy Disk: A floppy disk is a flexible disk with a magnetic coating on it. It is packaged inside a protective plastic envelope.

Hard disk: A hard disk consists of one or more circular disks called platters which are mounted on a common spindle. Each surface of a platter is coated with a magnetic material.

2. Optical storage media

In optical storage media information is stored and read using a laser beam. The data is stored as a spiral pattern of pits and ridges denoting binary 0 and binary 1.

Examples: CDs and DVDs

Compact Disk: A Compact Disc drive(CDD) is a device that a computer uses to read data that is encoded digitally on a compact disc(CD). A compact disk or CD can store approximately 650 to 700 megabytes of data.

Auxiliary Memories

DVD:

It stands for Digital Versatile Disk or Digital Video Disk. It looks just like a CD and use a similar technology as that of the CDs but allows tracks to be spaced closely enough to store data that is more than six times the CD's capacity.

It is a significant advancement in portable storage technology. A DVD holds 4.7 GB to 17 GB of data.

Blue Ray Disk:

This is the latest optical storage media to store high definition audio and video. It is similar to a CD or DVD but can store up to 27 GB of data on a single layer disk and up to 54 GB of data on a dual layer disk.

While CDs or DVDs use red laser beam, the blue ray disk uses a blue laser to read/write data on a disk.

Auxiliary Memories

Example of Auxiliary Memories

DVD

CD

Hard Disk

Daily Quiz

- Sketch memory hierarchy .
- Sketch 2.5D RAM organization.
- Write some differences between SRAM and DRAM.
- What do mean by virtual memory in COA.
- Which page replacement algorithm is good & why ?

Weekly Assignment

- Explain memory hierarchy with suitable diagram.
- Analysis different mapping scheme of cache memory.
- Discuss 2 D RAM and 2+1/2D RAM with suitable diagram.
- Write short notes on-
  a) Cache Memory b) Associative Memory c) Auxiliary Memory
- Explain the method to improve the performance of cache memory.

Faculty Video Links, You tube Courses Details

You tube/other Video Links

- https://www.youtube.com/watch?v=NVUWlO5zsk0
- https://www.youtube.com/watch?v=zwovvWfkuSg
- https://www.youtube.com/watch?v=m1dA7D6c3C0
- https://www.youtube.com/watch?v=o2_iCzS9-ZQ
- https://www.youtube.com/watch?v=pJ6qrCB8pDw&list=PLIY8eNdw5tW-BxRY0yK3fYTYVqytw8qhp

MCQ

1 Which of the following is the fastest means of memory access for CPU?
a) Registers b) Cache c) Main memory d) Virtual Memory

2. Size of the ____memory mainly depends on the size of the address bus.
a) Main b) Virtual c) Secondary d) Cache

3. What is the location of the internal registers of CPU?
a) Internal b) On-chip c) External d) Motherboard

4. MAR stands for _____

5. Which of the following is non-volatile storage?
a) Backup b) Secondary c) Primary d) Cache

6. Which of the following is used in main memory?
a) SRAM b) DRAM c) PRAM d) DDR

7. RAID stands for _____

Solution 1 b. 2a. 3 b. 4 Memory address register. 5 b 6 b

7 Redundant array of independent disks

Old Question Papers

5 Attempt any **two** parts of the following : **10×2=20**

(a) A ROM chip of 1024 × 8 bits has four select inputs and operates from a 5-volt power supply. How many pins are needed for the IC package? Draw a block diagram and label all inputs and output terminals in the ROM.

(b) Define the terms address space and memory space. An address space is specified by 24 bits and the corresponding memory space by 16 bits. Find the following :

(i) How many words are there in the address space?

(ii) How many words are there in the memory space?

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

(c) Write short notes on any **two** of the following :

(i) Auxiliary memory

(ii) Memory Hierarchy

(iii) Virtual Memory

(iv) Cache Memory.

Old Question Papers

(f) Convert the following decimal numbers to the bases indicated :

(i) 7625 to octal

(ii) 1983 to Hexadecimal

(iii) 174.5 to Binary

(iv) 6279 to octal

(v) 3001 to Hexadecimal

2. Attempt **any four** parts : **(4x5=20)**

(a) What is stack organization ? Compare Register stack and Memory stack.

(b) Explain addressing modes. Define the role of programme counter in addressing mode.

(c) What is CISC ? Explain it with its characteristics.

(d) What is the radix of number if the solution to the quadratic equation :

$x^2 - 10x + 31 = 0$

is $x = 5$ and $x = 8$.

(e) Show the multiplication process using Booth's algorithm when the following numbers are multiplied :

$(-12) * (-18)$

(f) Show the block diagram of the hardware that implements the following register transfer statements.

$y\ T_2 : R_2 \leftarrow R_1,\ R_1 \leftarrow R_2.$

3. Attempt **any two** parts : **(2x10=20)**

(a) What is Microinstruction ? How is it different from microprogram sequence ? Explain with the help of example.

(b) An encoded microinstruction format is to be used. Show how a 9 - bit microoperation field can be divided into sub-fields to specify 46 different actions.

(c) How a processor execute instructions ? Define the internal functional units of a processor and how they are interconnected ?

4. Attempt **any two** parts : **(2x10=20)**

(a) What are semiconductor RAM memories ? Show the read operation and write operation in static memories with examples.

(b) Explain the concept of Virtual memory. How address mapping is performed in virtual memory ?

(c) What is difference between 2D and $2\frac{1}{2}$ D memory organization ? Explain it with the help of suitable examples.

5. Attempt **any two** parts. (Write the short notes) : **(2x10=20)**

(a) Direct Memory Access (DMA).

(b) Synchronous and Asynchronous communication.

(c) Interrupts with their types and exceptions.

- o O o -

Expected Questions for University Exam

- Describe CAM in memory unit with major characteristics with suitable diagram.
- A computer uses RAM chips of 1024*1 capacity.
  - i) How many chips are needed & how should their address lines be connected to provide a memory capacity of 1024*8 ?
  - ii) How many chips are needed to provide a memory capacity of 16KB ?
- Calculate the page fault for a given string with the help of LRU & FIFO page replacement algorithm, Size of frames = 4 and string 1 2 3 4 2 1 5 6 2 1 2 3 7 6 3 2 1 2 3 6
- Explain Memory hierarchy with suitable diagram.

- Define Direct and Set associative mapping in a cache memory.

Summary

In previous slides we discuss in details

Memory Unit:

- Basic concept and hierarchy
- semiconductor RAM memories
- 2D & 2 1/2D memory organization
- ROM memories
- Cache memories: concept and design issues & performance address mapping and replacement
- Auxiliary memories: magnetic disk, magnetic tape and optical disks
- Virtual memory: concept implementation