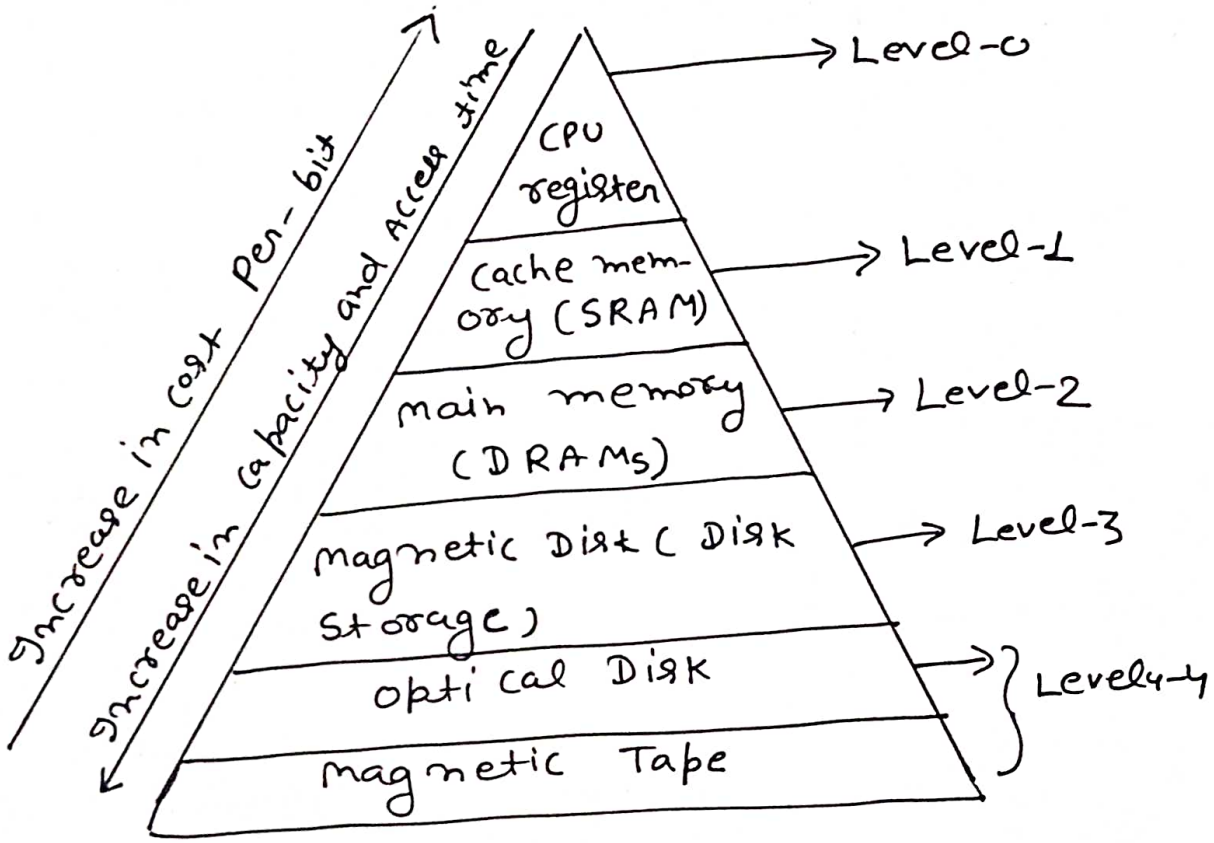
UNIT-4

MEMORY

BASIC CONCEPT AND HIERARCHY:-

Memory Hierarchy:-



(memory Hierarchy in a computer system)

⇒ memory unit is an essential component in any digital computer.

⇒ It is needed for Storing Programs and data.

⇒ The memory unit that communicates directly with the CPU is called the main memory.

M.S

⇒ Devices that Provide back-up storage are called "Auxiliary memory". The most common auxiliary devices used in computer Systems are magnetic Tapes and magnetic Disks. They are used for storing system Programs, large data files and other back-up information.

⇒ The total memory capacity of computer can be visualized as being a Hierarchy of computer.

⇒ The memory Hierarchy system consists of all Storage devices employed in a computer.

⇒ The memory Hierarchy System of all Storage devices employeed in a computers of system from the slow to an even smaller and faster cache memory accessible to High-Speed Processing Logic.

⇒ Magnetic Tapes are used to store removable files.
⇒ magnetic disks are used for Back-up Storage.

M·S

⇒ Magnetic Disks are used for Back-up Storage.

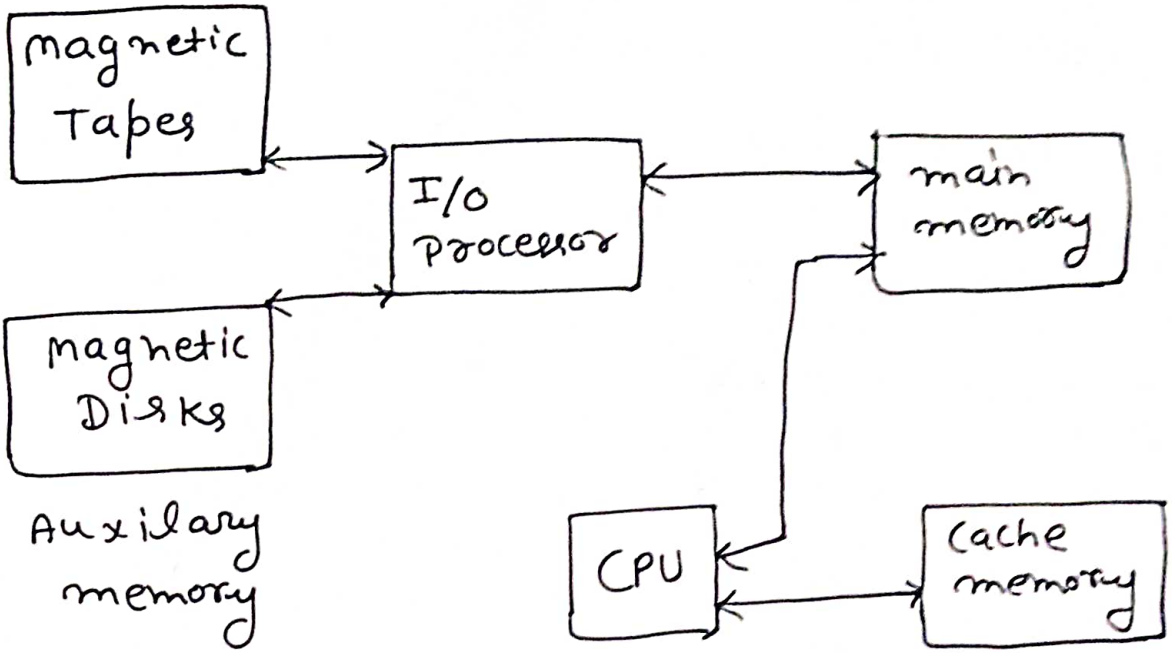
⇒ The main memory occupies a central position by being able to communicate with the CPU and with Auxiliary memory devices through and I/O Processor.

⇒ when Programs not residing in main memory are needed by the CPU. They are bought in from Auxiliary memory Programs not currently needed in main memory are transferred into Auxiliary memory.

⇒ A special very- High speed memory called a Cache, sometimes used to increase the speed of Processing by making current Programs and data available to CPU at a rapid rate.

Memory Hierarchy in a computer System:-

M.S



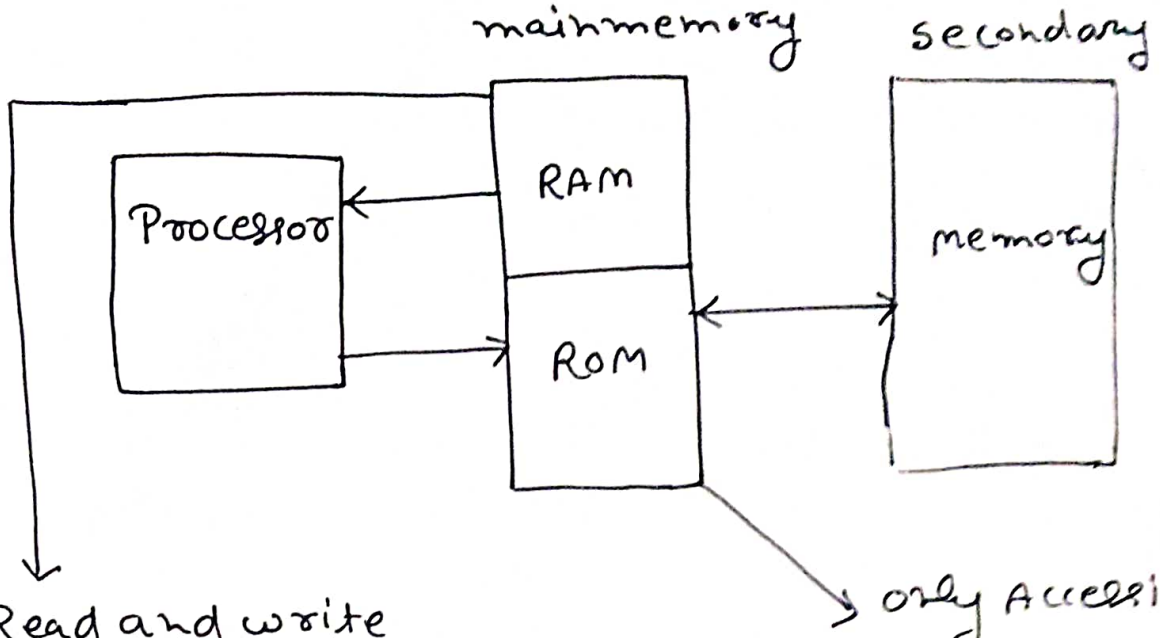
(memory hierarchy in a computer system)

⇒ The overall goal of using a memory Hierarchy is to obtain the highest possible average Access Speed while minimizing the total cost of the entire memory system.

⇒ multi Programming enable the CPU to Process a number of independent Program concurrently.

⇒ while the I/O Processor manages the data transfers between Auxiliary memory and main memory. the cache organization concerned with the transfer of information between the main memory and CPU.

M·S



Read and write operations are allowed.
Stores file/information required by Processor for execution of different Processes.

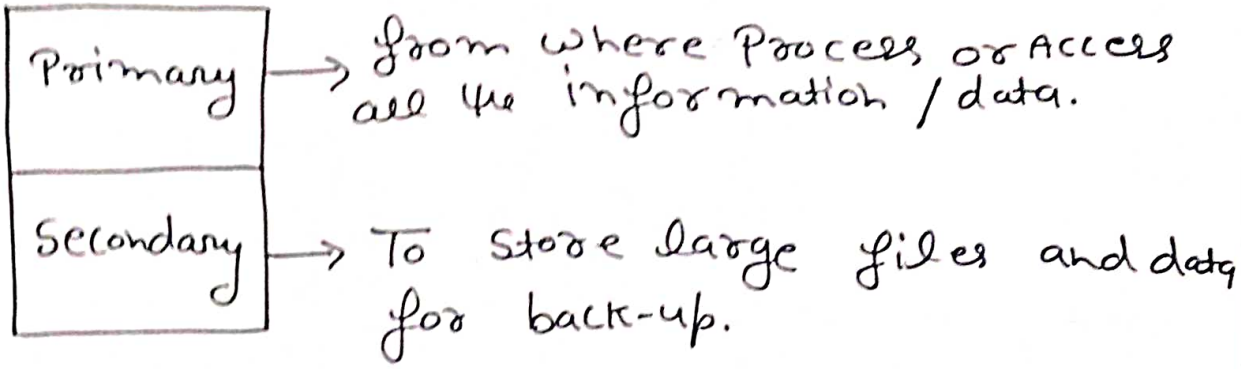
only Accessing of information. No write operation is allowed.
Generally store system files. which mendatory for system/computer execution.

Cache memory:- To increases Processing Speed we use a special type High speed memory unit that Stores the frequently used data by Processor. This is called cache memory. It is an intermediate memory that exist between Processor and main memory.

⇒ Auxiliary memory has
(a) Large storage capacity
(b) In expensive
(c) Low Access speed.

⇒ Cache memory has
(a) Small storage capacity.
(b) expensive
(c) High Access speed.

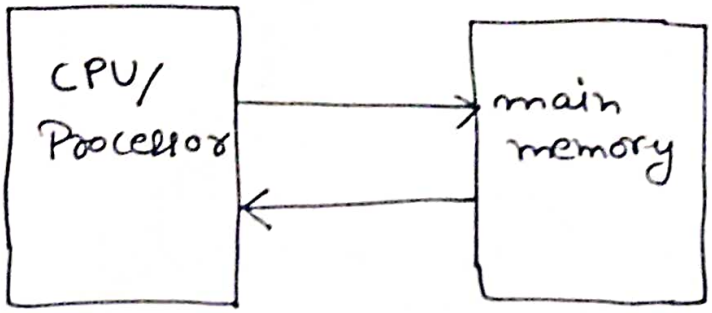
Memory:- The storage unit in computer do store information and data.

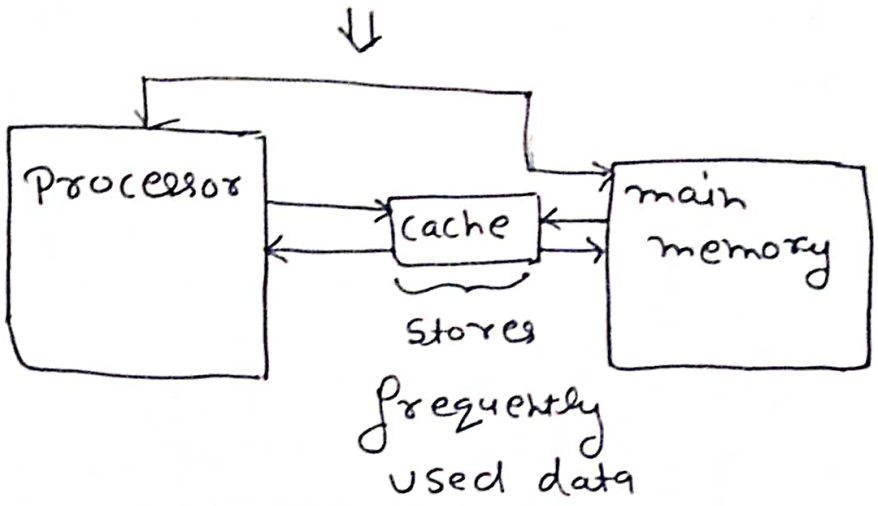


Primary memory / main memory:- mainly of two types.

(1) RAM ⇒ Random Access memory.
(2) ROM ⇒ Read only memory.

M·S





Cache memory is high speed memory

Size of cache memory is less than main memory.

if the required data is found in cache than it is called cache hit. Otherwise it is called cache miss.

for ex we searched for information 10 times and we found the required data 8 time and 2 times the data was not found.

Hit = 8

miss = 2

$$\frac{\text{Number of Hit}}{\text{Number of miss}} = \text{Hit Ratio}$$

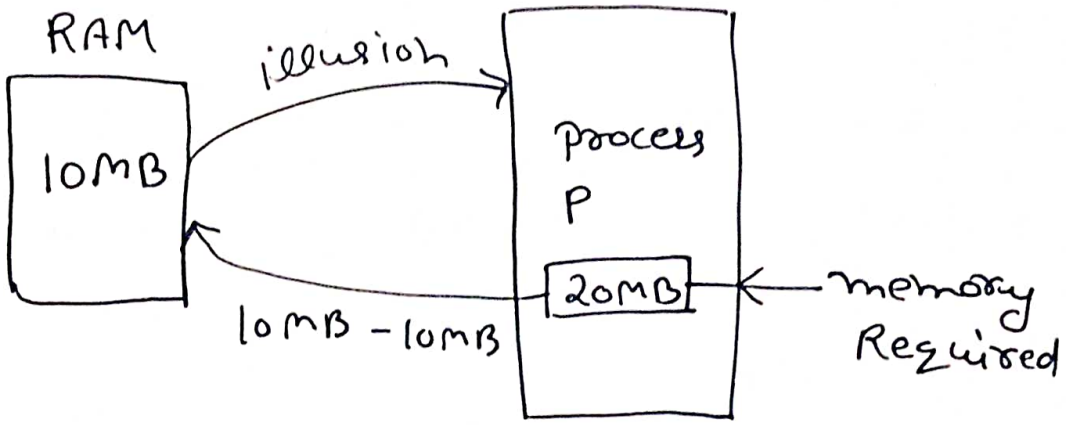
M·S

Factors for memory:

① Accessing Speed.
② Storage Capacity - Size
③ Cost

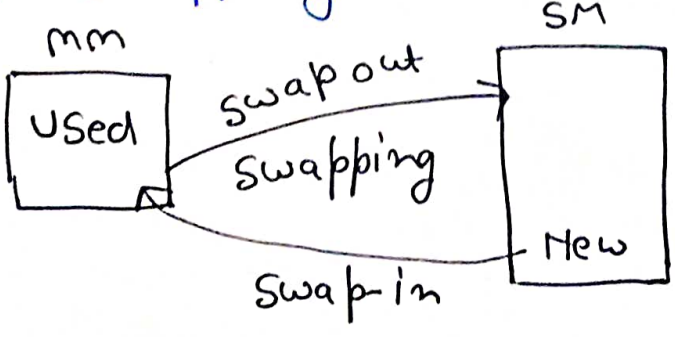
When we arrange different types of memories according above factors, then that arrangement is called memory Hierarchy.

Virtual memory:- virtual memory illusion of very large size memory.



Concept of virtual memory
⇒ swapping

implementation of virtual memory required swapping.



Virtual memory:- Virtual memory is a storage scheme that provides user an illusion of having a very big main memory. This is done by treating a part of secondary memory as the main memory.

In this scheme user can load the bigger size processes than the available main memory by having the illusion that the memory is available to load the process.

Instead of loading one big process in the main memory the operating system loads the different parts of more than one process in the main memory.

How virtual memory works:- In this scheme whenever some pages needs to be loaded in the main memory for the execution and the memory is not available for those many pages. then in that case instead of stopping the pages from entering in the main memory, the operating system search for the RAM area that are least used in that recent times or that are not

M.S

referenced and copy that into the secondary memory to make the space for the new pages in the main memory.

Advantages of virtual memory:-

① The degree of multiprogramming will be increased.

② User can run large application with less real RAM.

③ There is no need to by more memory RAM's.

Disadvantages of virtual memory:-

① The system becomes slower since swapping takes time.

② It takes more time in switching between applications.

③ The user will have the lesser hard disk space for its use.

Auxilary memory:- Types of secondary memory (SM)

⇒ Pen drive

⇒ CD (compact disk)

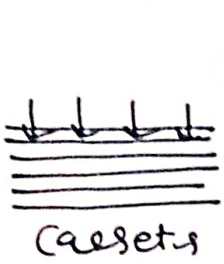
M·S

⇒ DVD (Digital versatile disk)

⇒ Magnetic Tape

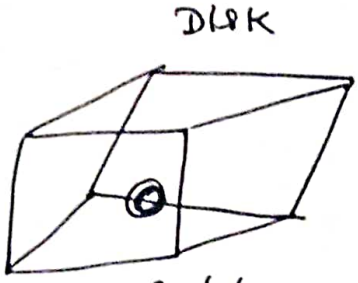
⇒ Magnetic Disk

⇒ External Hard disk.

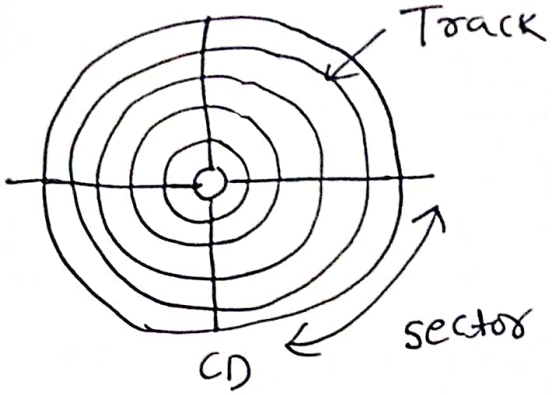


Casets

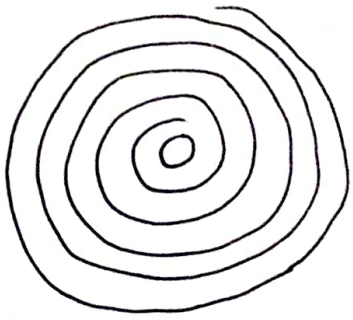
(Magnetic Tape)



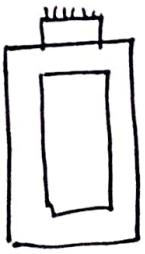
Floppy



CD



DVD



circuit-Base-chip

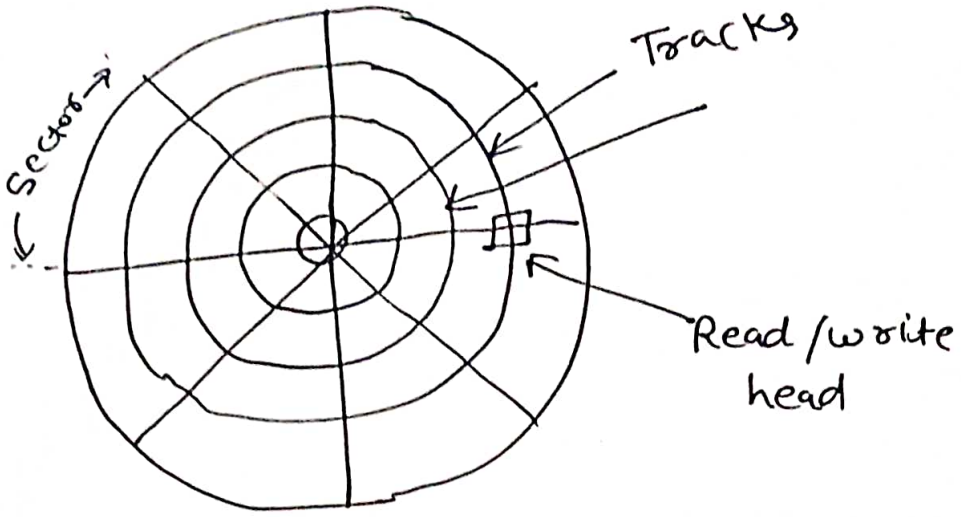
Auxilary memory :- Anxilary memory is known as the lowest-cost highest-capacity and slowest access storage in a computer system. It is where Programs

M.S

Auxilary memory:- An auxilary memory is known as the lowest-cost highest-capacity and slowest access storage in a computer system. It is where Programs and data are kept for long-term storage or when not in immediate use. The most common examples of auxiliary memories are magnetic tapes and magnetic Disks.

Magnetic Disks:- A magnetic disk is a type of memory constructed using a circular plate of metal or Plastic coated with magnetized materials. usually. Both sides of the disks are used to carry out read/write operations. However several disks may be stacked on one spindle with read/write head available on each surface.

M.S



⇒ The memory bits are stored in the magnetized surface in spots along the concentric circles called tracks.

⇒ The concentric circles (tracks) are commonly divided into section called sectors.

Magnetic Tape :- magnetic tape is a storage medium that allows data archiving collection and back-up for different kinds of data. The magnetic tape is constructed using a Plastic strip coated with a magnetic recording medium.

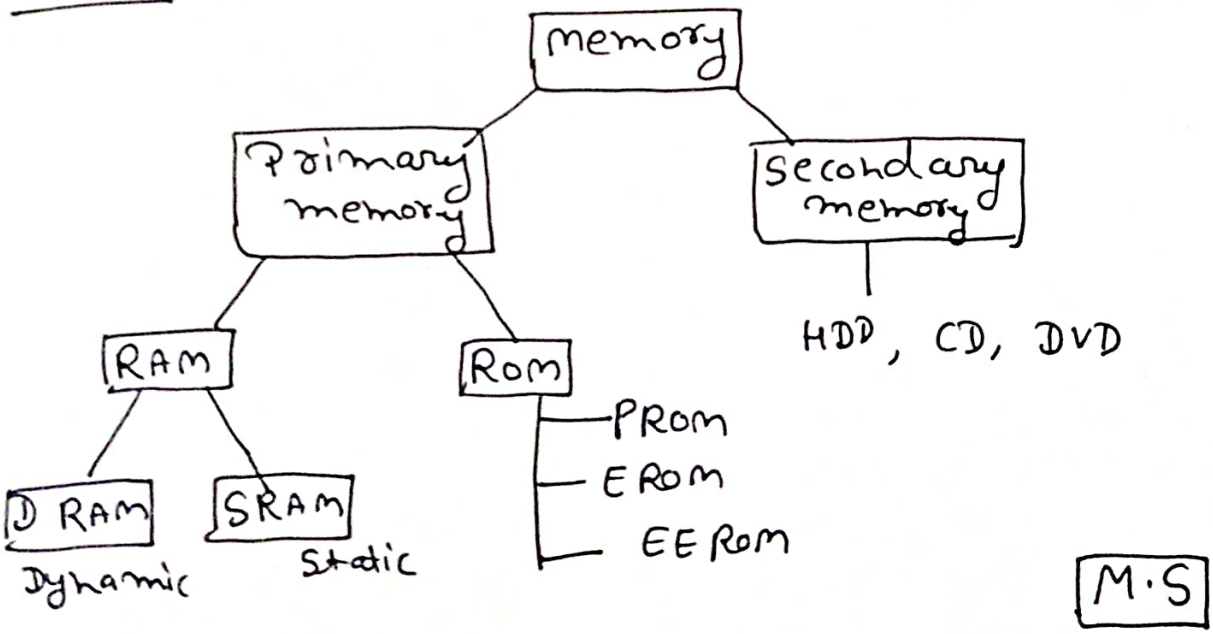
M.S

The bits are recorded as magnetic spots on the tape along several tracks. usually seven or nine bits are recorded simultaneously to form a character together with a parity bit.

Magnetic tape units can be halted started to move forward or can be rewound. However they cannot be started or stopped fast enough between individual characters for this reason, information to as records.

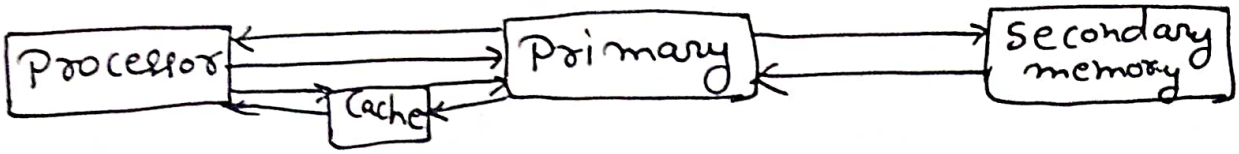
Memory and Types:- computer memory is a storage device/place that is used to store information.

Types:- main two types of memory.



M·S

Processor communicate with Primary memory.



Random Access memory – RAM Read & write
Read only memory - Read (System files
Cache memory (Special memory).

Primary memory :- It is also known as main memory. The memory unit that communicate directly with the CPU called Primary memory.

RAM (Random access memory) :- RAM Stands for Random access memory. It is volatile memory. The tasks currently Performed by the CPU are Stored in RAM.

Note :-

(1) ~~Ram~~ RAM is Part of CPU
(2) RAM data directly accessed by the CPU.
(3) It is very fast memory.

Types of RAM :-

(1) SRAM
(2) DRAM

M.S

Secondary memory!- It is also known as auxilary memory. The memory unit which is Provide back-up storage called secondary memory.

Ex Pendrive, HDD, floppy disk.

Difference between RAM & ROM!-

RAM	ROM
① RAM Stands for Random Access memory.	① ROM stands for Read only memory.
② RAM is a volatily memory.	② ROM is a non volatile memory.
③ volatile memory means the stored data are lost, if the Power goes off.	③ Non-volatile memory means the information stored in it is not lost even if the Power supply goes off.
④ RAM data can be read erased or modified.	④ ROM data is only read only.
⑤ RAM Speed is High.	⑤ ROM Speed is slow as compare to RAM.
⑥ RAM capacity is High.	⑥ ROM capacity is Low.
⑦ RAM is costly.	⑦ ROM is Cheaper as compare to RAM

M.S

(8) RAM Speed used to store data that has to be currently processed by CPU temporarily.	(7) ROM is used to store data that in needed to boot strap the computer.
(9) Static RAM and Dynamic RAM is the type of RAM.	(8) MI ROM PROM, EP ROM, EEROM are the type of ROM.

Difference between SRAM and DRAM:-

SRAM	DRAM
(1) SRAM cosists of internal flip-flops.	(1) DRAM consists of MOS transistor and capacitors.
(2) Information Stored in the form of bits.	(2) Information Stored in the form of charge.
(3) No chance of Lekage.	(3) Possibility of charge leakage.
(4) SRAM's are costlier than DRAM.	(4) DRAM's are Less costlier than SRAM.
(5) more Power Consumption than DRAM.	(5) DRAM's Less Power Consumption than SRAM
(6) Low Storage density than DRAM.	(6) ~~It is read~~ High Storage density than SRAM.

M·S

Disadvantages of fifth generation Computers:-

⇒ It tends to be sophisticated and complex tools.

⇒ It Pushes the Limit of transistor density

⇒ In these computers, very Low-Level Languages are required.

⇒ They may be harmful in terms of making Human brains doomed and dull.

COMPUTER MEMORY:-

Memory is Just like a Human brain. It is used to Store data and instructions.

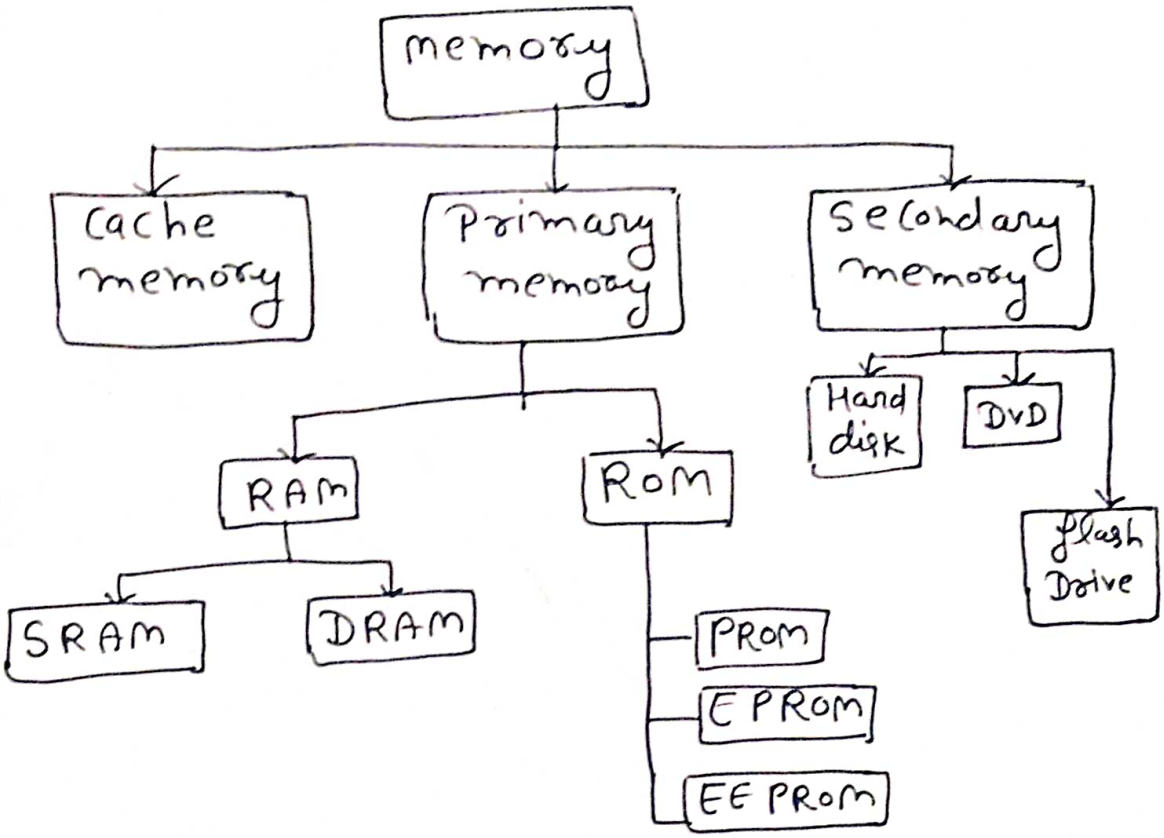
⇒ Computer memory is the Storage Space in the computer, where data is to be Processed and instructions required for Processing and Stored.

⇒ The memory is divided into Large number of small Parts called cells.

⇒ Each cell has a unique address which varies from 0 to memory Size minus.

⇒ Whenever we talk about the memory of a computer System, we usually talk about the main or Primary memory.

Types of computer memory :-



Cach.

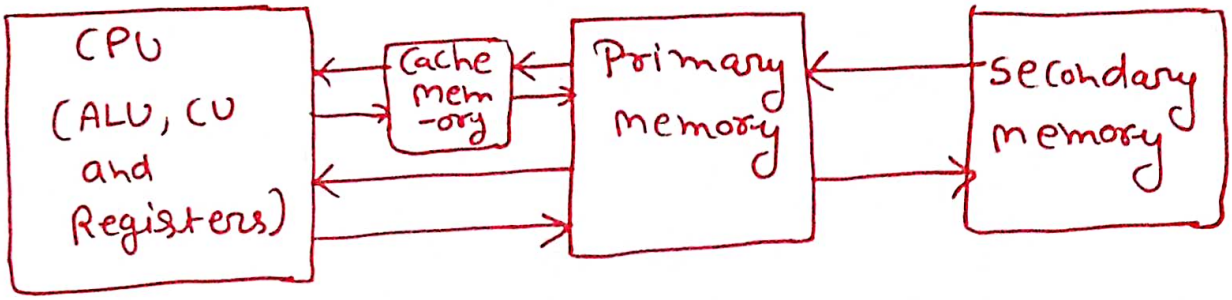
Cache memory :-

⇒ Cache memory is very High Speed Semiconductor memory which can Speed up the CPU.

⇒ It acts as a buffer between the CPU and main memory.

⇒ Cache memory is used to reduce the average time to access data from the main memory.

⇒ It is used to hold those Parts of data and Program which are most frequently used by the CPU.



Advantages of Cache memory :-

⇒ Cache memory is faster than main memory.
⇒ It consumes less access time as compared to main memory.
⇒ It stores the program that can be executed within a short period of time.
⇒ It stores data for temporary use.

Disadvantages of Cache memory :-

⇒ Cache memory has Limited capacity.
⇒ It is very expensive.

Primary memory (main memory) :-

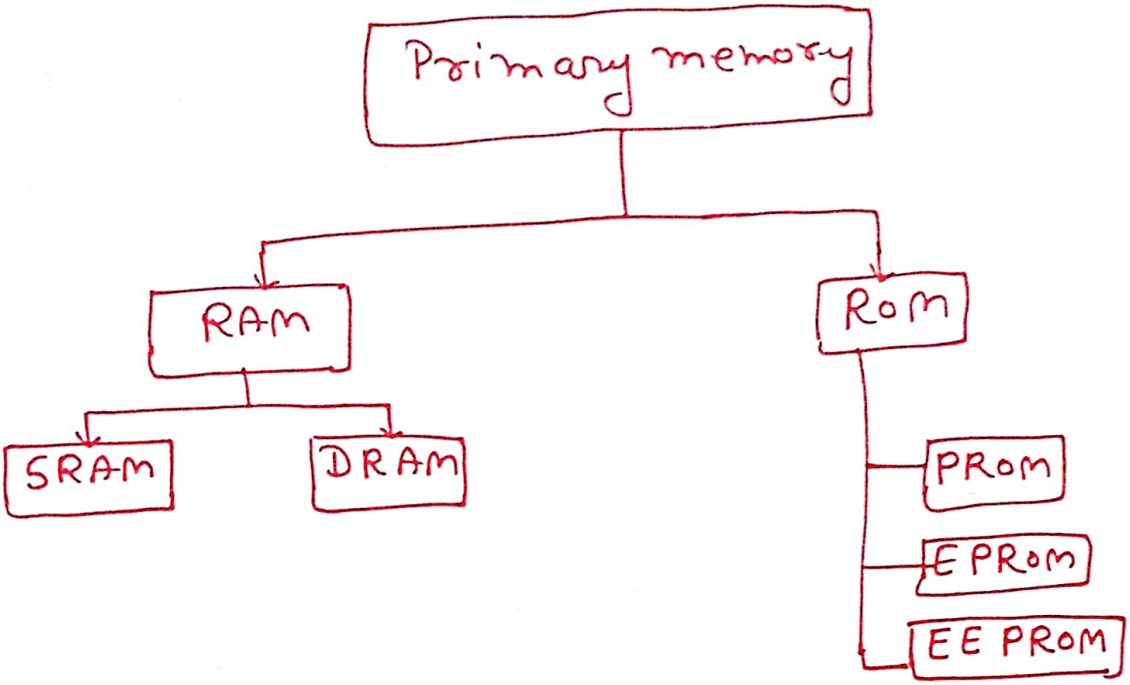
⇒ Primary memory is an essential component of a computer system.
⇒ Program and data are loaded into the Primary memory before Processing

Primary memory is computer memory that is accessed directly by the CPU.

⇒ Primary memory is also known as main memory or may also refer to Internal memory and primary storage.

⇒ Computer memories that are directly accessed by the Processor using data bus are called Primary memory.

There are two types of Primary memory



(1) RAM (Random Access memory) :-

⇒ RAM is the main memory or the Primary Storage of the computer and is also known as the working area of the computer.

⇒ RAM is valatile in nature. That means Data is retained in RAM as long as

the computer data is on, but it is lost when the computer is turned off. when the computer is rebooted, the OS and other files are re-loaded, the OS and other files are reloaded into RAM, usually from an HDD and SSD.

⇒ The features of the RAM are that the computer can read the data from RAM and can also write Data into the RAM.

There are two types of RAM:-

(1) SRAM (Static Random Access memory):-

⇒ SRAM is costly because the computer can read and write the data from SRAM and it also consumes less power.

⇒ Data is stored using the six transistor memory cell.

⇒ Static RAM is mostly used as a cache memory for the Processor (CPU).

Characteristics of Static RAM:-

(1) Long Life
(2) No need to refresh
(3) faster
(4) used as cache memory.
(5) Large size.

(2) DRAM (Dynamic Random Access memory) :-

⇒ It is less costly than the SRAM and that's why the DRAM consumes more energy and is also slower than the SRAM.

⇒ It is a type of RAM which allows you to store each bit of Data in a separate capacitor within a particular integrated circuit.

⇒ It is standard computer memory of any modern desktop computer.

Characteristics of Dynamic RAM :-

(1) Short Data lifetime.
(2) Needs to be refreshed continuously
(3) Slower as compared to SRAM
(4) Used as RAM
(5) Smaller in size.
(6) Less expensive.

ROM (Read only memory) :-

⇒ Read only memory (ROM) is non-volatile in nature. meaning it also holds its memory even when power is removed.

⇒ As it name shows that its read-only. so we cannot write into it.

⇒ It is a type of storage medium that permanently stores.

There are two types of ROM :-

(1) PROM (Programmable Read only memory) :-

⇒ Its limitation is that PROM Programs the data only once after that cannot be changed so due to this features it is named as a one time programming device.

(2) EPROM (Erasable Programmable Read only memory)

It's next versions of the ROM and in it, the data can be Programmed many times but when this rewriting and erasing is Performed the chip becomes useless.

(3) EEPROM (Electrically erasable ROM) :-

This is the next version of ROM and EPROM and in this, the data can be erased using Electrical Signals. And the data can be erased many times in EEPROM about ten thousands time data can be erased and rewritten.

Characteristics of ROM :-

(1) Non-volatile in nature.
(2) Cannot be accidentally Changed.
(3) Easy to test.
(4) more reliable than RAM's
(5) static and do not require refreshing.

Secondary memory :-

⇒ Secondary memory or auxiliary memory is to Permanently store the data and instruction for future use.

⇒ The secondary memory is non-volatile and has Larger storage capacity than primary memory.

⇒ It is slower and cheaper than the main memory. But it cannot be accessed directly by the CPU.

⇒ The CPU can't directly access the secondary memory. first the secondary memory data is transferred to Primary memory then the CPU can access it.

Example: examples of secondary memory devices include Hard Disk, Drive (HDD), CD/ DVD, memory card etc.

⇒ SSD (Solid State Drive) is the Latest Secondary Storage Device which support very fast data transfer speed as compared to HDD's.

Characteristics of Secondary memory :-

⇒ These are magnetic and optical memories.

- It is known as the backup memory.

⇒ It is non-volatile memory.
⇒ Data is permanently stored even if power is switched off.
⇒ It is used for storage of Data in a computer.
⇒ Computer may run without the secondary memory.
⇒ Slower than Primary memories.

Difference between RAM & ROM

RAM	ROM
(1) RAM stands for Random Access memory.	(1) ROM stands for Read only memory.
(2) Volatile (maintains its data while the device is powered) in nature.	(2) Non-volatile (does not Lose content when power is Lost) in nature.
(3) Stores information temporary.	(3) Stores information permanently.
(4) Requires flow of electricity to retain Data.	(4) Does not require flow of electricity to remain Data.
(5) Large size with Higher Capacity.	(5) Small size with Less Capacity.
(6) Used for Both read and write.	(6) Used only for reading.
(7) Used for both read and write.	(7) Used only for reading.

2D and 2.5D (Dimension) memory / RAM :-

The internal structure of memory is made of cells that contain memory bit. The group of 8-bit make-up a byte (8 bit = 1 byte).

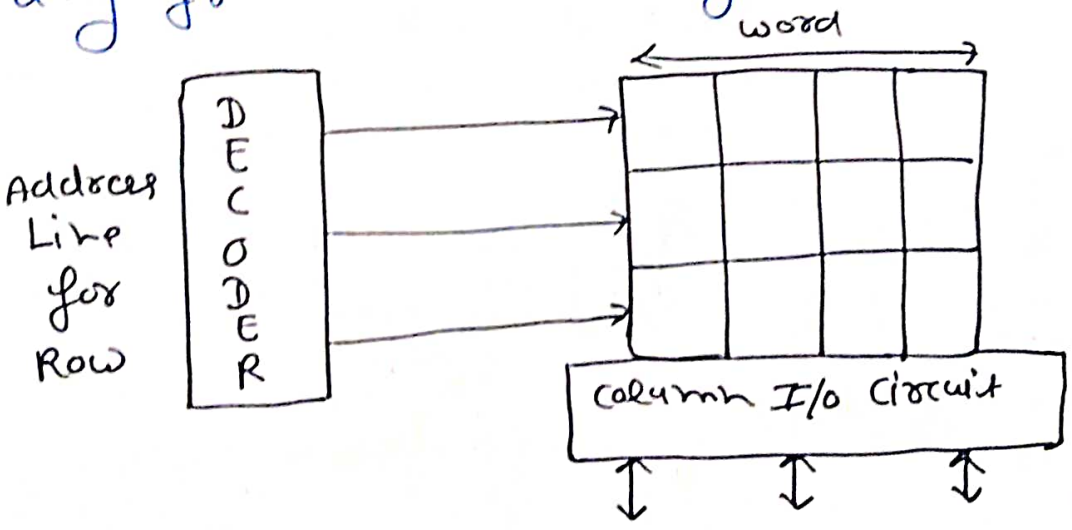
The memory is in the form of multi-dimensional array of rows and columns. memory can be organized in two ways.

① 2-D - organization.

② 2.5D - organization.

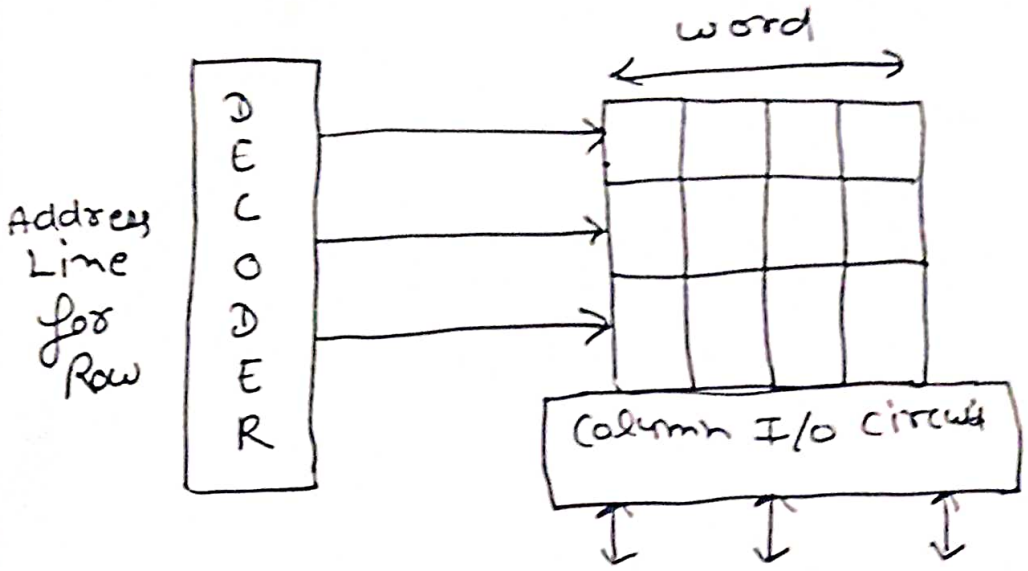
2-D organization :- In 2-D organization memory is divided in the form of matrix. each row contains a word (set of 4-bits).

In this memory organization. A decoder is used for the Address calculation of any from the memory in Row wise manner.

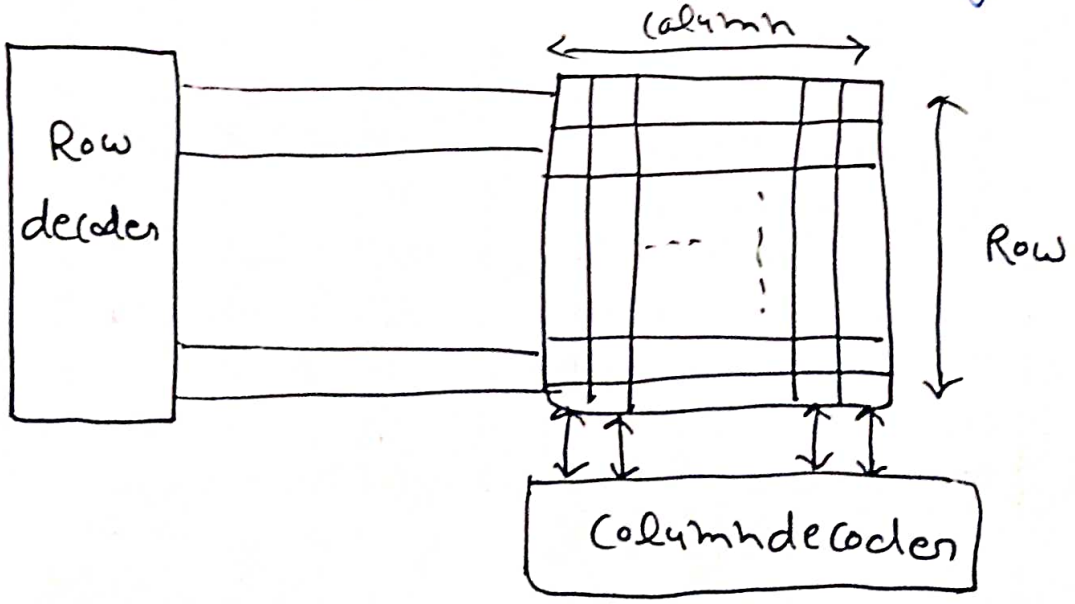


M·S

calculation of any from the memory in Row wise manner.



2.5D organization:- In 2.5-D organization All of scenario is same but we have two different decoders are for Row and another for column. to calculate or find the Addresses of the words in Both column (reference) or Row reference.



Comparison between 2-D & 2.5-D organizations:-

① In 2-D hardware is fixed but in 2.5 hardware may change.

② 2-D required more number of gates while 2.5 D required less.

③ 2-D is more complex than 2.5D.

Page Replacement Algorithms:

⇒ In a computer operating system that uses Paging for virtual memory management. Page Replacement Algorithms decide which memory Page-out (Swap-out - write to disk) when a Page of
⇒ memory needs to be allocated.

⇒ When a Program starts execution one or more Pages are transferred in to main memory and the Page-table is set to indicate their position.

The Program is executed from main memory until it attempts to reference a Page that is still in auxiliary (Secondary) memory. This condition is called Page fault.

M·S

Page fault:- If the required page is not present in (main) memory in(main) memory. The Page fault occurs which requires I/O operations.

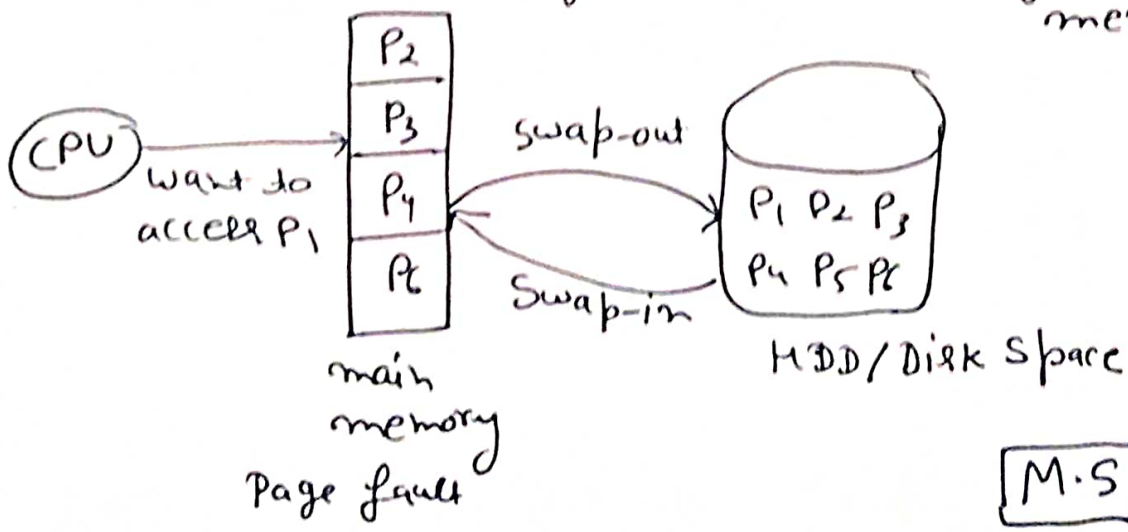
When Page fault occurs the execution of Program is suspended until the required Page is brought into main memory.

* A new is then transfered from auxiliary memory to main memory.

* If main memory is full. It would be necessary to remove a Page from memory block to make room for the new Page.

* The Policy for choosing Pages to remove is determined from Page Replacement algorithm is used.

Page Replacement selection of Page to be removed from main memory.



M.S

The goal of a replacement policy is to try to remove the page least likely to be referenced in the immediate future.

most commonly used page Replacement are:

① FIFO (first in first out)
② LRU (Least-recently used)
③ OPT (optimal
④ MRU (most-recently used)

① FIFO (first in first out):-

Page-Replacement Algorithm:-

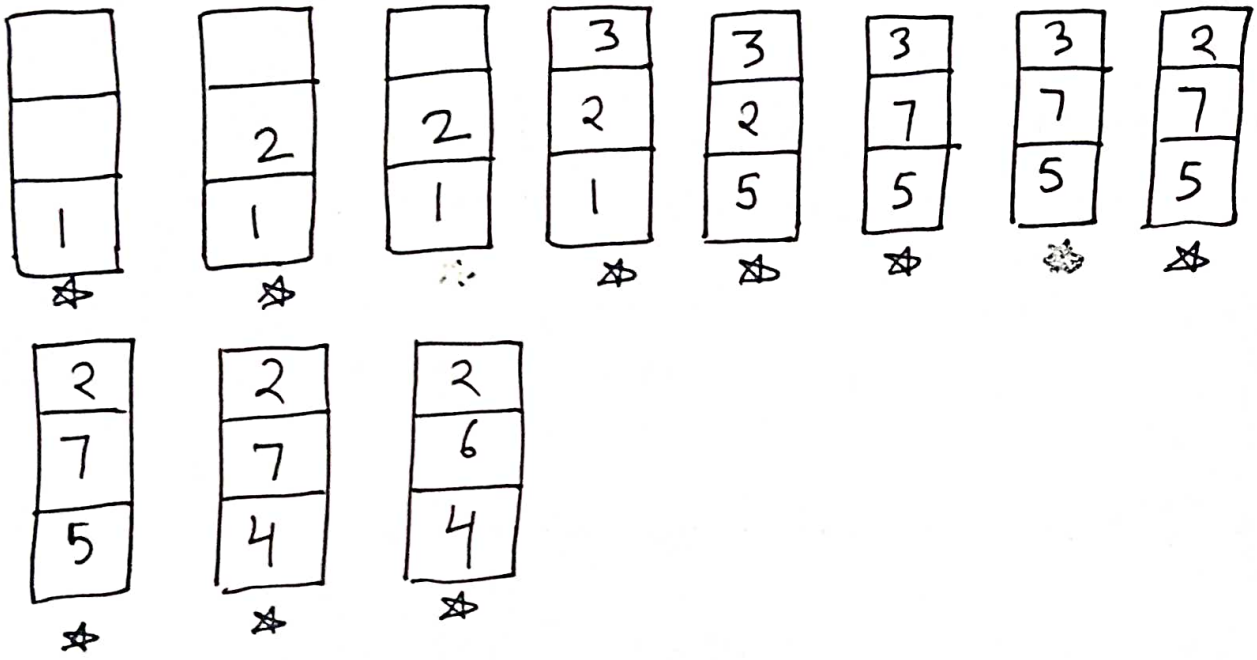
⇒ FIFO is easy to implement.
⇒ It selects the page for replacement that has been in memory the longest time.
⇒ oldest page in memory is selected for replacement.
⇒ It keeps a list & Replace page from fail and add page at the head.

M·S

Reference String:-

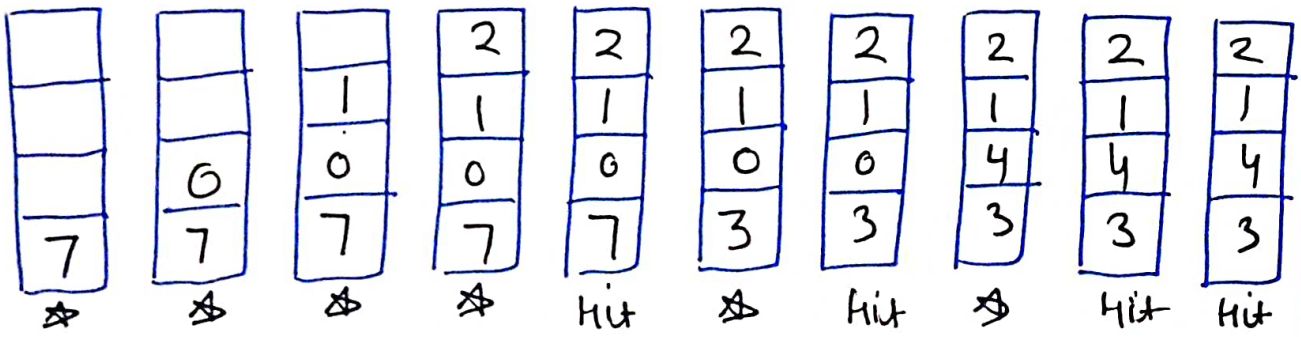
1, 2, 1, 3, 5, 7, 3, 2, 5, 4, 6

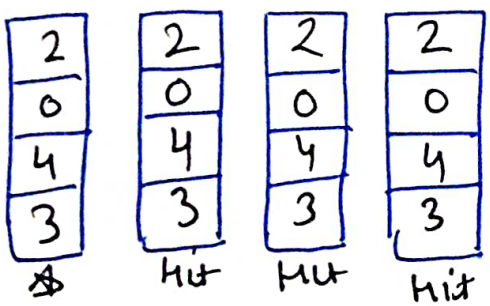
Page frame = 3



Ex: 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 3

number of frame is 4. find the number of page fault (FIFO).





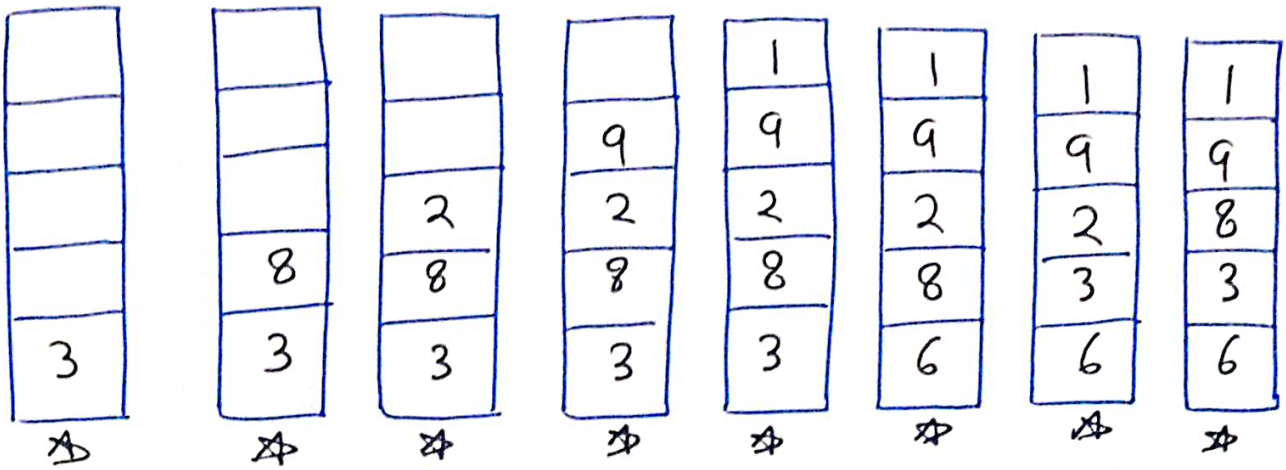
Number of Page fault = 7

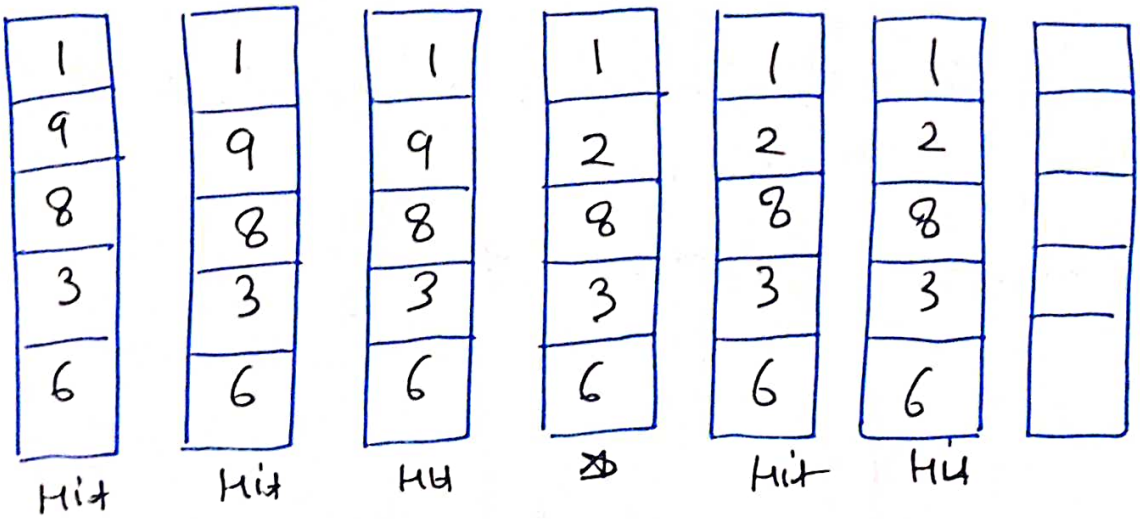
Page Hit = 7 Ans

Ex-3 3, 8, 2, 3, 9, 1, 6, 3, 8, 9, 3, 6.

2, 1, 3

Consider a main memory with five Page frames and the following sequence of Page references (FIFO).





Number of Page fault = 9

Hits = 6 Ans

M·S

LRU (Least Recently used) Page Replacement Algorithm:

⇒ LRU selects a page for replacement which has not be used for the longest time in main memory.

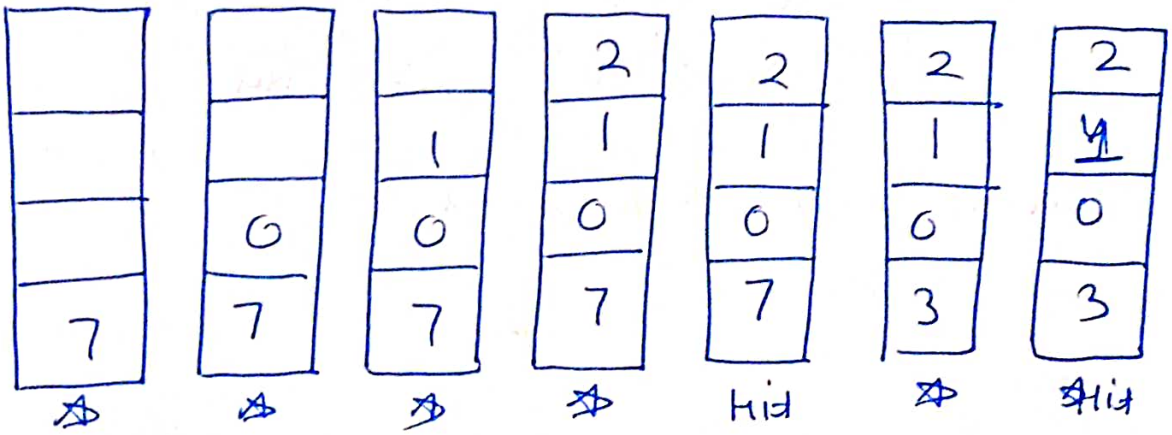
⇒ Keeps a list and replace Pages by looking back into time.

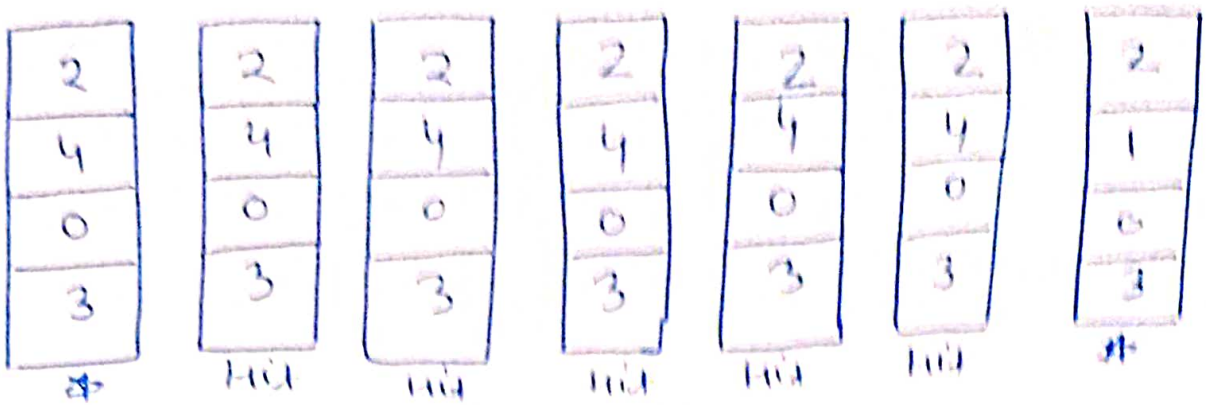
⇒ It is implemented by associating a counter with every Page that is in main memory.

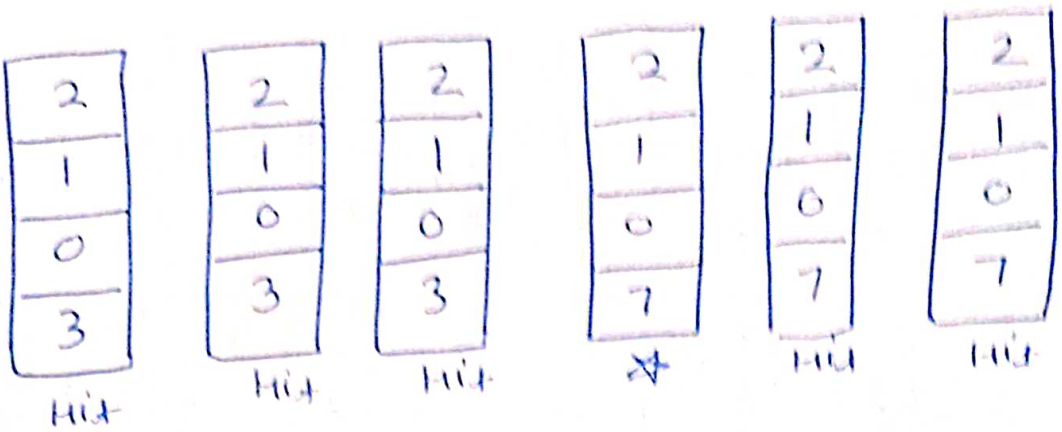
Ex Reference String:

7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1

number of frames = 4.



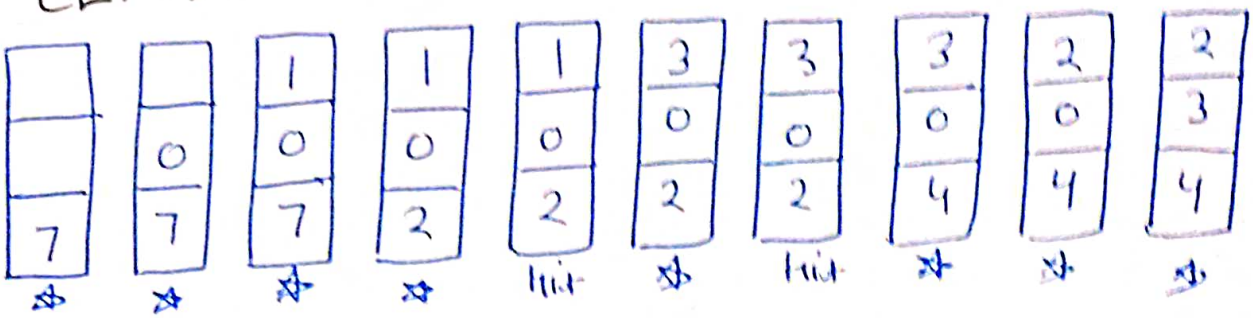


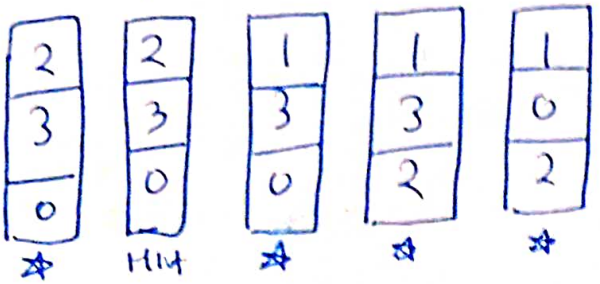


Number of fault = 8

Hit = 12 Ap

Ex2 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 1, 2, 0 Number of frame = 3 (LRU).





Number of Page fault = 12

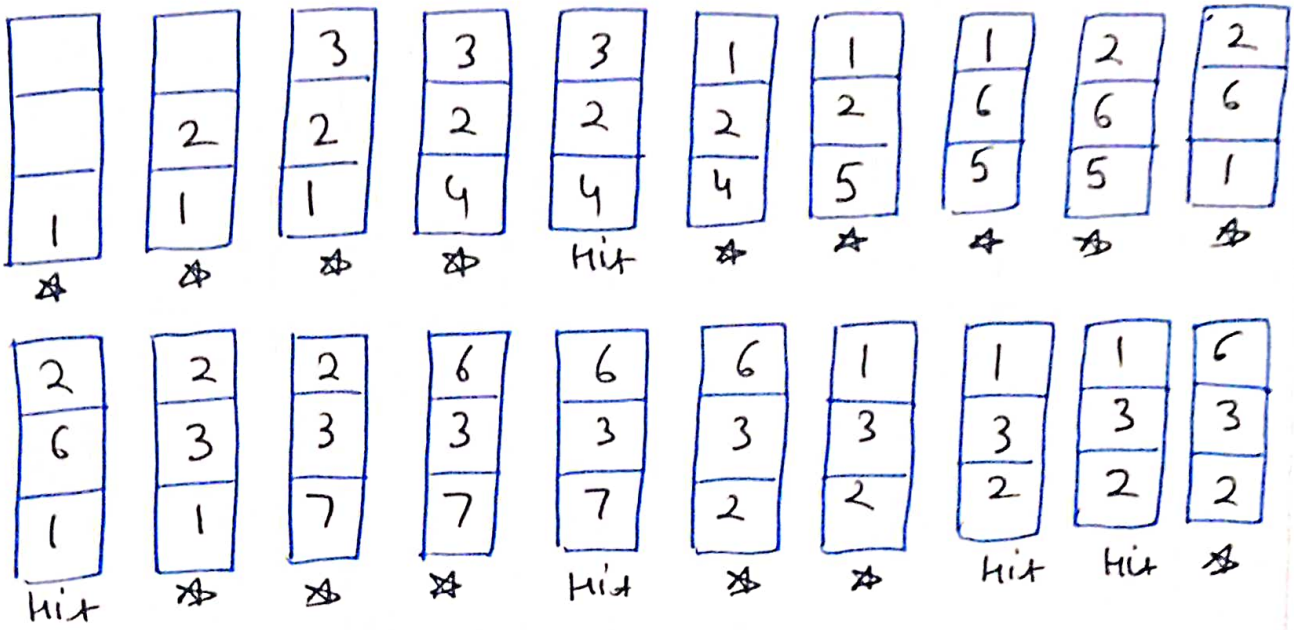
Hit = 3

Ap

Ex-3 Reference string :

1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6,
3, 2, 1, 2, 3, 6,

Number of frames = 3 (LRU).



Number of fault = 15

Hit = 5 Ans

(3) Optimal Page Replacement Algorithm :-

⇒ It has lowest Page fault rate of all algorithms.

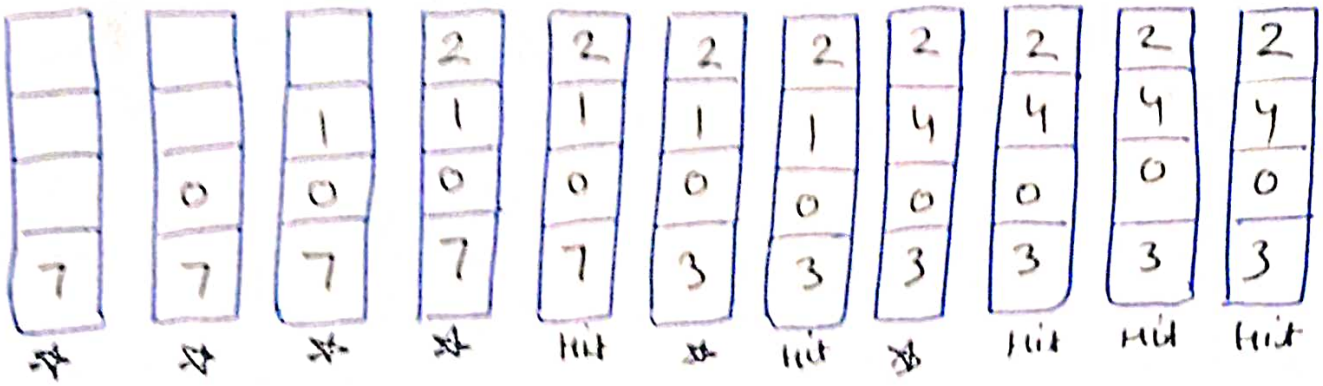
⇒ optimal Policy replace the Page that will not be used for the longest Period of time.

⇒ It uses the time when a Page is to be used next.

M·S

Eg1 Reference string:

7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0,
1, 7, 0, 1. Number of frame = 4 (OPT)



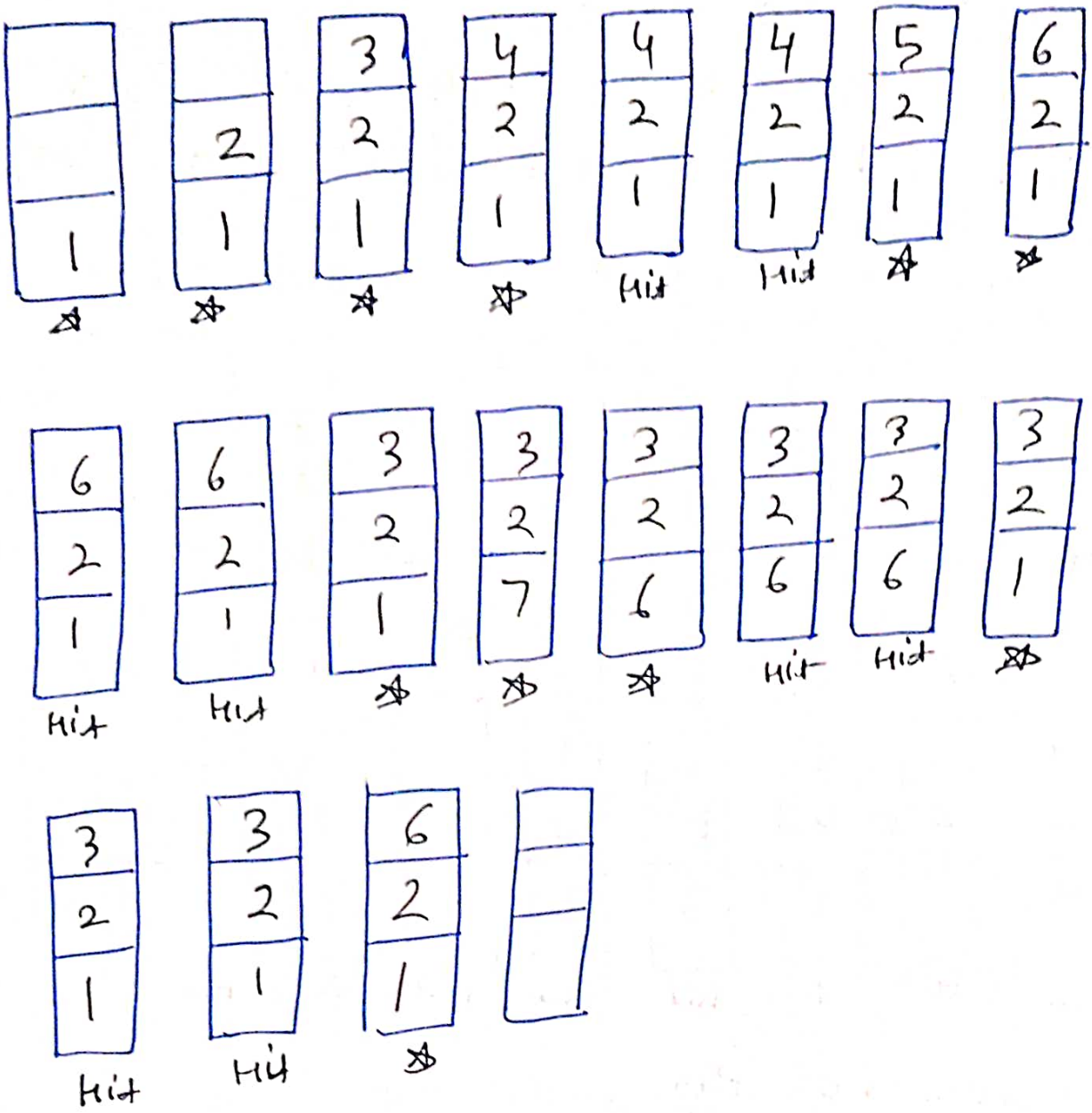
Number of Page fault = 8

Number of Hit = 12 Ans

Eg2 Reference String

1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3,
7, 6, 3, 2, 1, 2, 3, 6

Nomber of frame = 3 (OPT)



Number of Page fault = 11

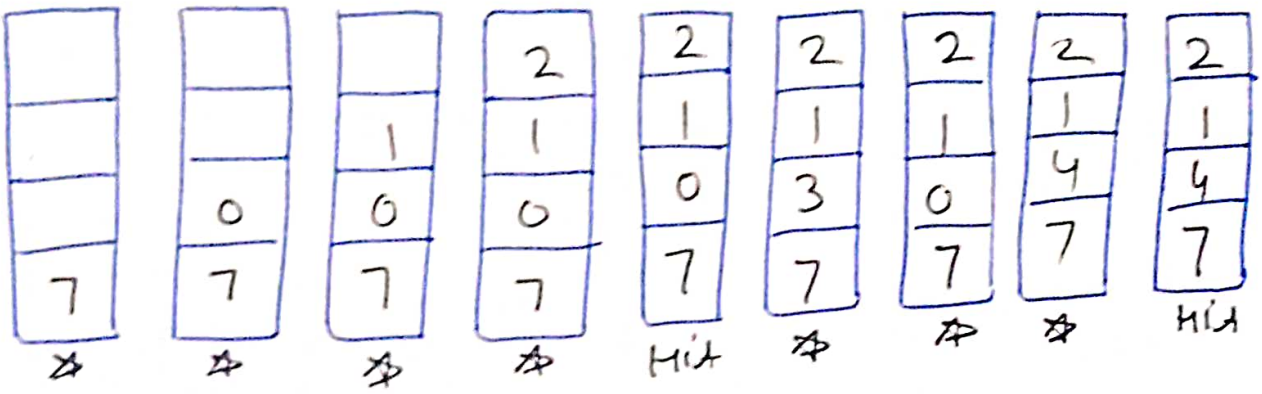
Hit = 8 Ans

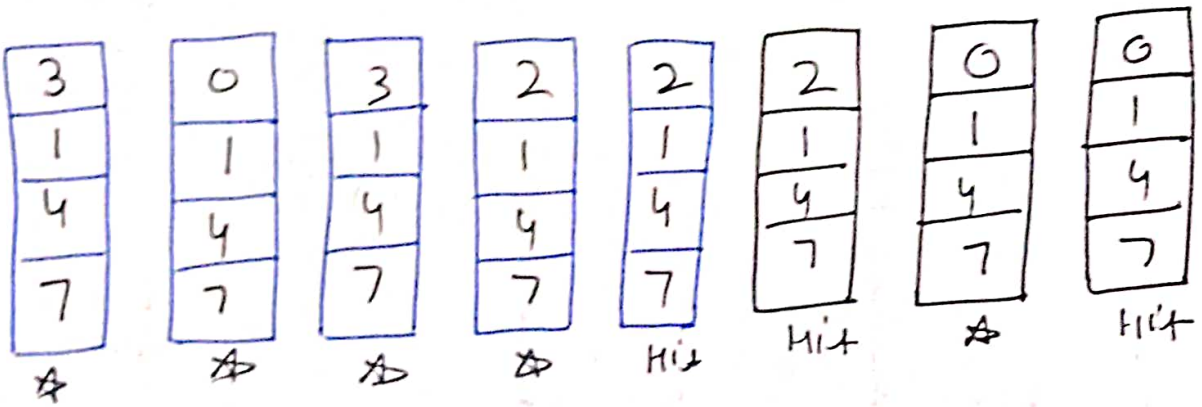
(4) most recently (frequently) used Page Replacement Algorithms (MRU):-

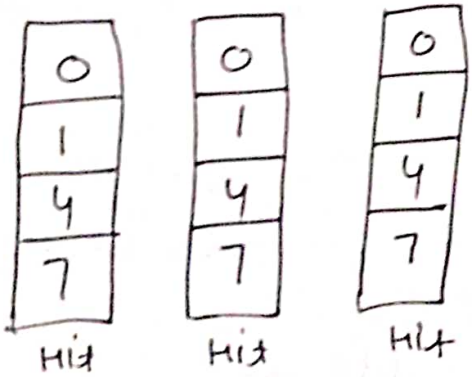
Replace the most Recently (frequently) used Page in Past.

Reference string:

7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0 1, 7, 0, 1 Number of frame = 4







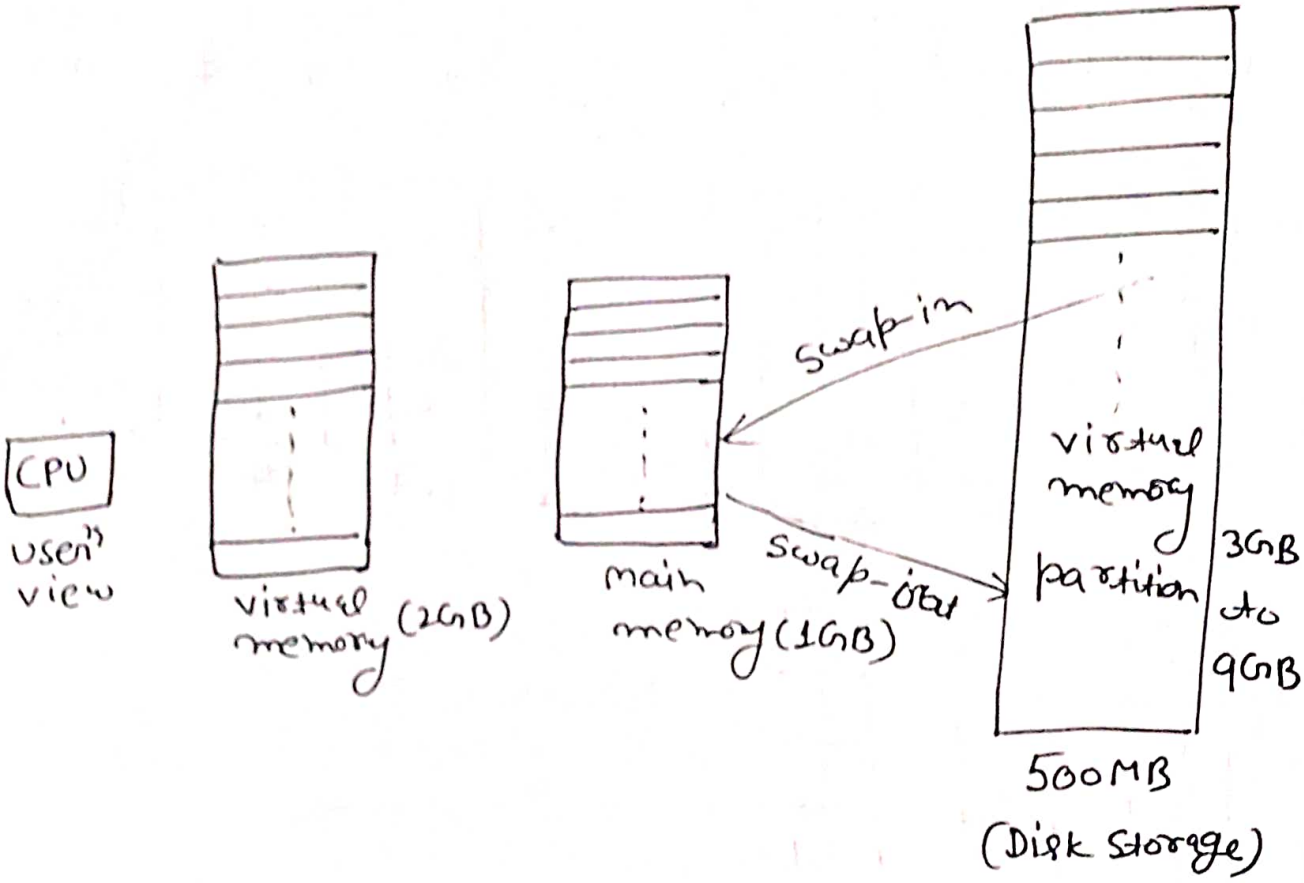
Number of Page fault = 12

Hit = 8 Ans

M·S

Virtual memory implementation using Paging:

Virtual memory is a concept that gives an illusion to the user that he has sufficient main memory (RAM) to execute any Program/application of any size, however computer actually have relatively smaller main memory.



Virtual Address (Logical Address):-

Each address in virtual memory is called address space.

M-5

Address space!- Set of all virtual address is called address space.

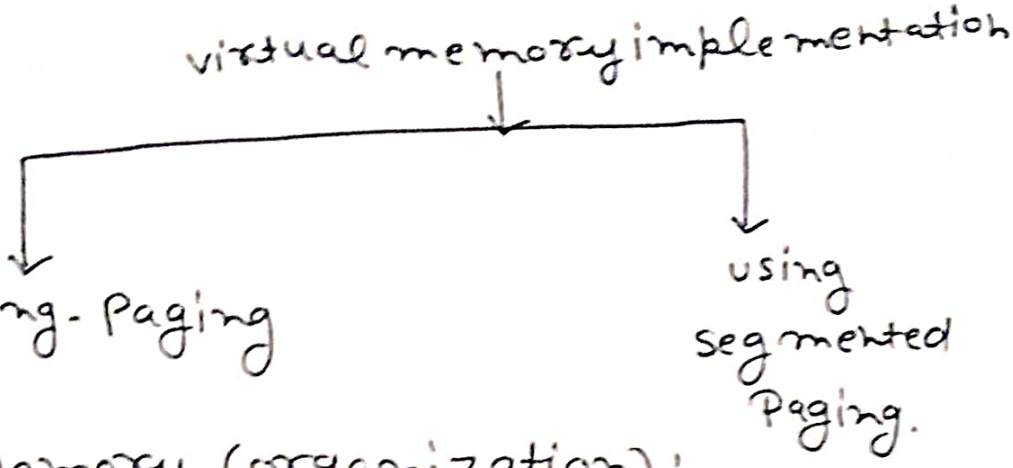
memory Address (Physical Address)!- Each Address in main memory is called memory Address.

memory space!- The set of all memory addresses is called memory space.

swapping!- swapping is a mechanism in which a process temporarily moved out from main memory to secondary storage (disk) and another process moved in from secondary storage (disk) to main memory. After some time the first process again brought back to main memory.

Address mapping!- Address mapping specify how to convert virtual address to memory Address.

M·S



Cache memory (organization) :-

Cache memory :- The term cache means a safe place for hiding or storing things.

⇒ Cache memory (CM) is a small, fast memory which holds copies Recently accessed instructions and data.

⇒ when the Processor makes a request for memory Reference the request is first Sought in the cache.

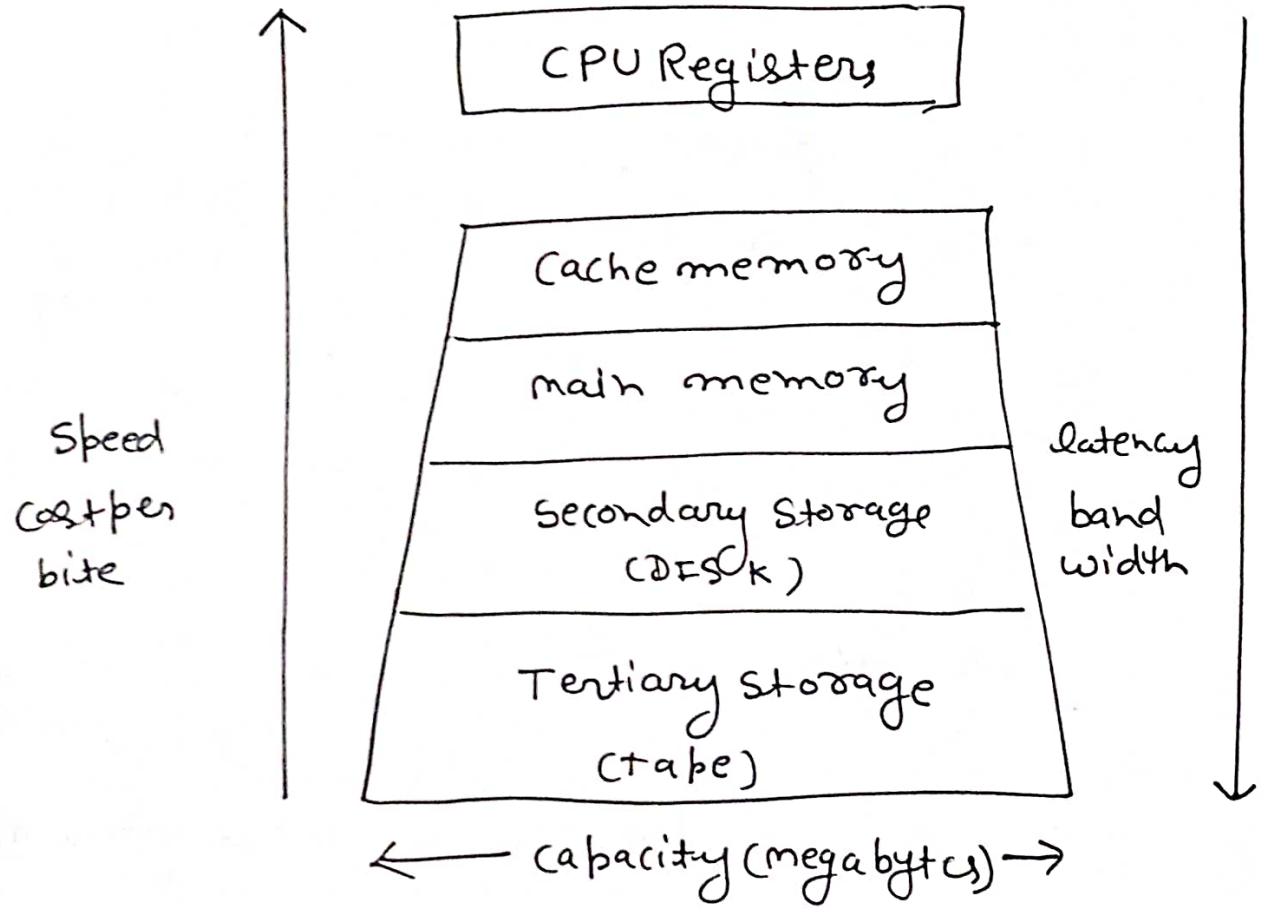
Cache memory

⇒ if we get that memory Reference which is Requested we call it. CACHE HIT otherwise CACHE MISS.

M.S

=> In the case of cache MISS, requested element is brought from a subsequent memory level from memory Heirarchy and Placed in cache.

=> In the cache of cache MISS, requested element is brought from a sub sequent memory level from memory.



(memory Hierarchy)

M.S

⇒ A block of elements are transferred from main memory to cache memory by expecting that the next requested element (Spatial locality) and this has to happened under one main memory Access time.

Some main points of cache organization:-

⇒ Cache is organized not in bytes, but as blocks of cache lines with each line containing some no of bytes (16-64)

⇒ Cache lines do not have fixed addresses which enables the cache system to populate each cache line with a unique (non-contiguous) address.

three methods for filling a cache lines.

① fully Associated — The most flexible.

② Direct mapping - The most basic.

③ Set Associative — A combination of the two-way set.

M.S

Cache memory mapping techniques:-

Cache memory mapping means how data is copied (mapped) from main memory to cache memory.

mapping techniques:-

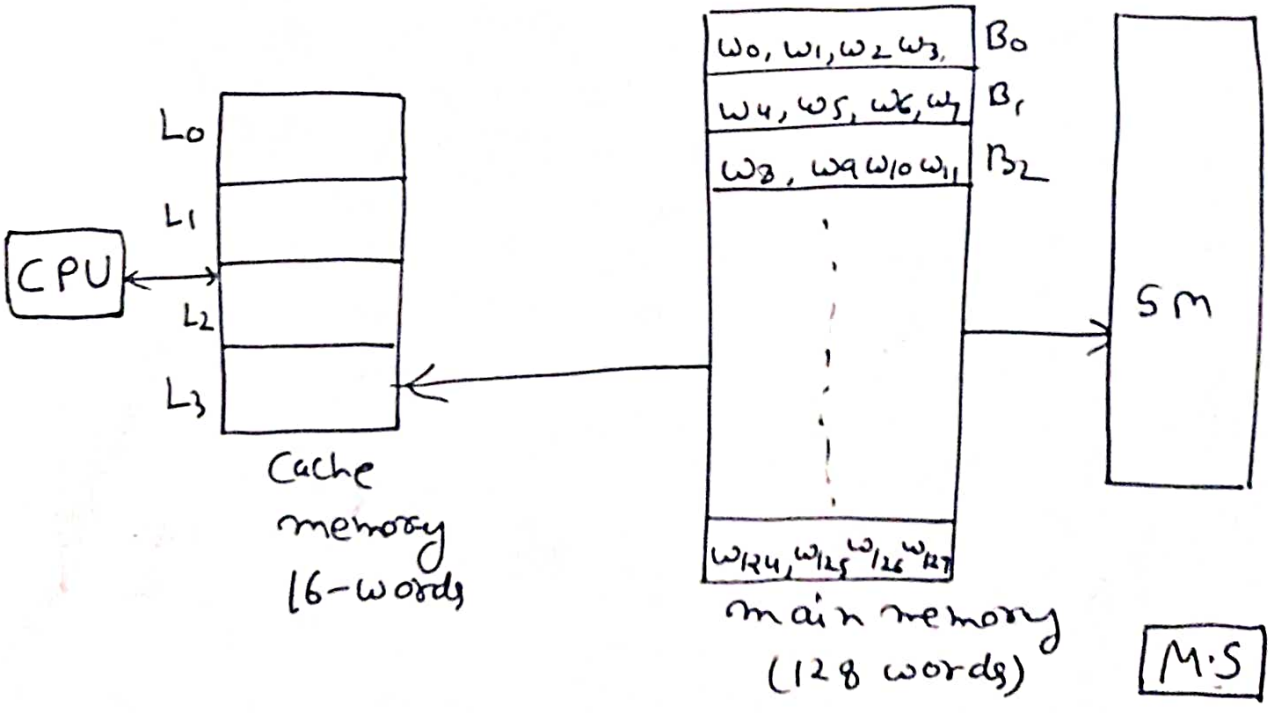
Direct mapping:- In this mapping main memory blocks are copied to a fixed block of cache memory. but oneat a time.

Let Cache memory block No = C_b

main memory block No = m_b

Number of block in Cache = C

Number of block in main memory = n



In which Line

$k \bmod n$

where k is block number

n = number of Lines

$B_0 = 0 \bmod 4 = 0$

↳ Remainder

$B_0 \rightarrow$ Line Number zero $\rightarrow L_0$

$B_1 \rightarrow 1 \bmod 4 = 1 \Rightarrow L_1$

$B_5 \rightarrow 5 \bmod 4 = 1 \Rightarrow L_1$

$B_2 \rightarrow 2 \bmod 4 = 2 \Rightarrow L_2$

$B_6 \rightarrow 6 \bmod 4 = 2 \Rightarrow L_2$

1 word = 1 byte

Block size = 4 words

total No of words = 128 words

Block = 4 words

No of block = $\frac{128}{4}$

= 32 blocks

Number of Line = 4

total No of words = 16

1 Line word = $\frac{16}{4}$

= 4 words

formula:

$$\text{Cache memory Block Number} = \left(\begin{matrix}\text{main memory} \\ \text{Block Number}\end{matrix}\right) \bmod \left(\begin{matrix}\text{Number of} \\ \text{cache block}\end{matrix}\right)$$

$$C_b = m_b \bmod C$$

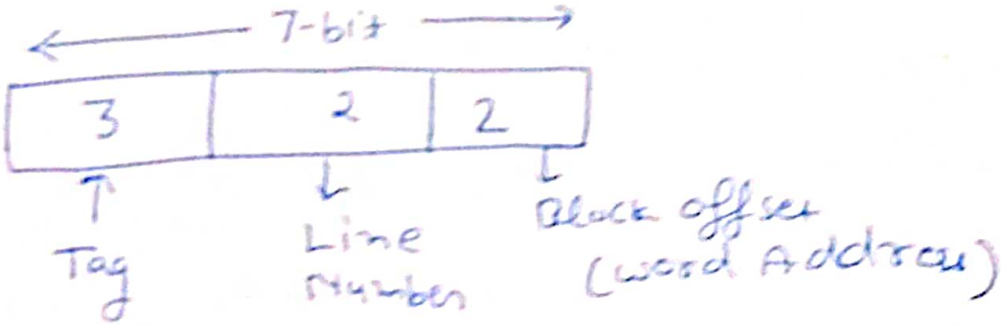
M.S

Block offset ⇒ Number of bytes required to represent a words

⇒ $2^{①} = 2$-bit

Cache Address :- Same Number of bits in add of cache also.

Logical Address :-



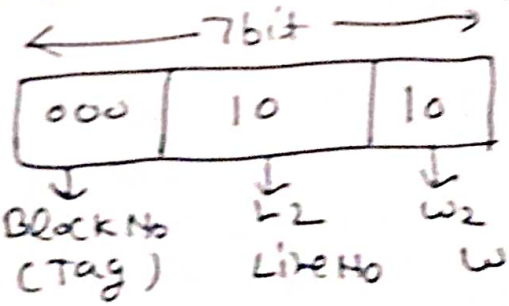
Ex

001	01	00 ⇒ 20
001	01	01 ⇒ 21
001	01	10 ⇒ 22
001	01	11 ⇒ 23

} block Line No - 5

000 10 10

Cache



m·5

Number of main memory blocks = $\frac{\text{total main memory words}}{\text{Block Size}}$

Number of cache memory blocks = $\frac{\text{total cache memory words}}{\text{Block Size}}$

At a time only one block can reside in a line.

Now How we can access a word

Range ⇒ 0 — 127

Physical Address :

Number of bits req $= 2^{7} = 128$

PA (7-bit)

5-bit	2-bit
Block No	Block offset

Block No $= 32$

$= 2^{5}$

M·S

Block of set = no of word in on block = 4

No of bits for block off-set $= 2^{(2)}$

$\Rightarrow$ 2 bit.

Numerical:-

(1) consider a direct mapped cache of size 16 KB with block size 256 bytes. The size of main memory is 128 KB find out the Number of bit in Tag and Tag directory size.

Sol

given

Cache memory size = 16 KB

Block size = Line size = frame size

= 256 bytes.

main memory size = 12 KB

(we consider that the memory is byte)

Number of bits in Physical Address

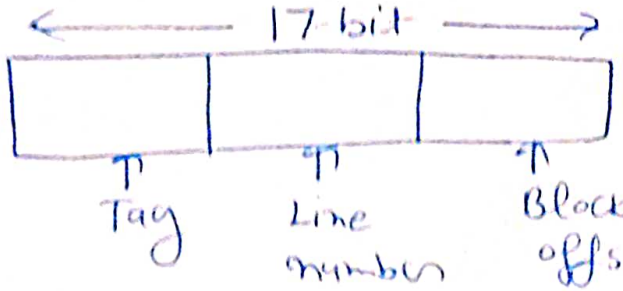
Size of main memory = 128 KB

$= 2^7 \times 2^{10}$ bytes

$= 2^{(17)}$ bytes

M·S

The number of Physical address = 17 bits



Number of bits in block offset

we have

Block size = 256 bytes

$= 2^8$ bytes

No of Bits in block offset = 8-bit

Number of bits in Line No

total No of Line in cache = $\frac{\text{Cache size}}{\text{Line size}}$

$= \frac{16\ KB}{256\ bytes}$

$= \frac{16 \times 10^{10}}{2^8}$

$= 16 \times 2^2$ bit

$= 2^4 \times 2^2$ bit

$= 2^6$ bit

$= 6$-bit

M·S

Number of bit in Line No = 6-bit

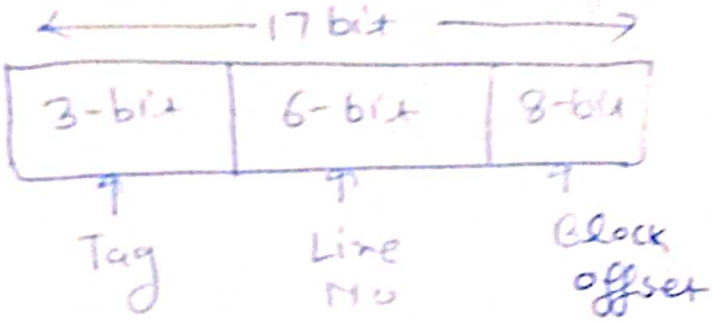
Number of Tag Bit = No of bit in Physical address —

(No of bits in Line No) + No of Bits in Block offset)

= 17 − (6-bit + 8-bit)

= 3-bit

No of Tag Bit = 3-bit



Tag directory Size = No of Tags x Tag size

= No of Line in cache x No of Bits in Tag

$= 2^6 \times 3\text{-bit}$

= 64 x 3-bit

= 192-bit

= 24 byte Ans

M·S

8-bit	= 1 byte
1024 bytes	= 1 kB
1024 kB	= 1 mB
1024 mB	= 1 GB
1024 GB	= 1 TB

Ex2 Consider a direct mapped cache of size 512 kB with Block size 1-kB There are 7 bits in the Tag. find size of main memory and Tag directory size.

Sol given

Cache memory size = 512 kB

Block size = frame size = Line size = 1 kB

No of Bits in Tag = 7 bits

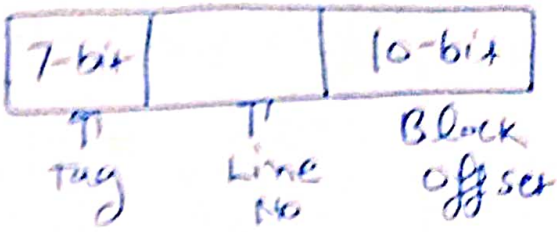
No of Bits in Block offset:

Block size = 1 kB

$= 2^{10}$ bytes

Number of bits in block offset = 10-bits

M.S



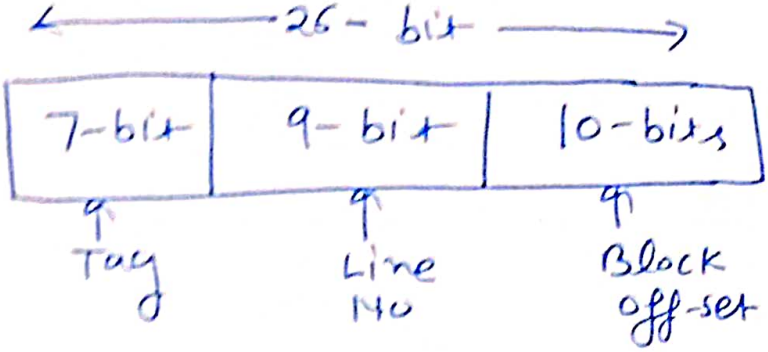
Number of bits in Line Number

Total Number of Line in Cache = $\frac{\text{Cache size}}{\text{Line size}}$

$= \frac{512\ KB}{1\ KB}$

$= 2^9$ lines

Number of bits in line Number = 9-bit



Number of Bits in Physical Address =

No of Bits in Tag + No of Bits in Line No + No of Bits in Block offset

$= 7 + 9 + 10$

$= 26$ bits

M·S

Size of main memory

Number of bits in Physical Address = 26-bit

Size of main memory = 2^{26} bytes

Size of main memory = 64 MB

Tag directory size = No of Tags ~~bit~~ x Tag size

= Number of Line in cache x No of bits in Tag

= 2^{9} x 7-bit

= 3584-bits

= 448 bytes

Tag directory size = 448 bytes

AP

M·S

Ex3 consider a direct mapped cache with block size 4KB. The size of main memory is 16 GB and there are 10-bits in the Tag find

(1) Size of cache memory.

&1 Tag directory size

Size of cache memory = 16 MB
Tag Directory size = 5120 bytes

Practice Problem

Numerical FULLY ASSOCIATIVE MAPPING:-

Ex1
Consider a fully associative mapped cache of size 16 KB with block size 256 bytes. The size of main memory is 128 KB find out the Number of bits in Tag and Tag directory size.

Sol given
cache memory size = 16 KB

Block size = Line size = 256 bytes

main memory size = 128 KB

Number of Bits in Physical Address

we have

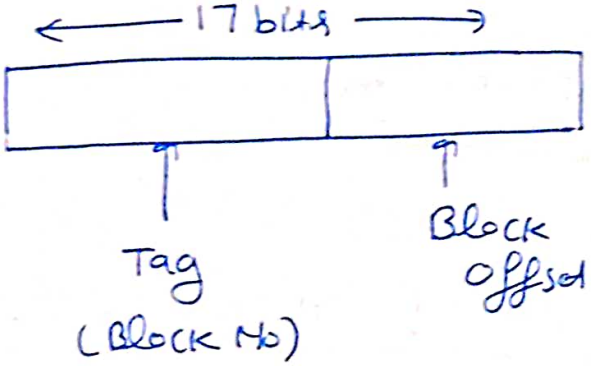
size of main memory = 128 KB

$= 128 \times 2^{10}$ bytes

$= 2^7 \times 2^{10}$ bytes

$= 2^{17}$ bytes

Number of bits in Physical Address = 17 bits

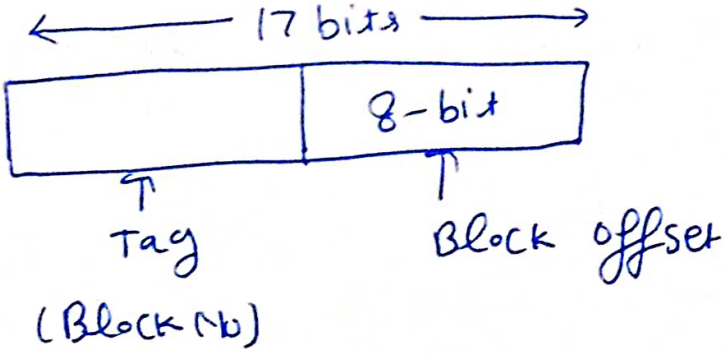


Number of Bits in Block offset

we have Block size = 256 bytes

$= 2^8$ bytes

Number of bits in block offset = 8-bits



Number of bits in Tag = No of Bits in Physical Address − No of Bits in Block offset

= 17 − 8 bits

= 9 bits

Number of bits in Tag = 9 bits

No of Line in cache = $\frac{\text{Cache Size}}{\text{Line size}}$

$= \frac{16\text{KB}}{256 \text{ bytes}}$

$= \frac{16 \times 2^{10} \text{ bytes}}{2^{8} \text{ bytes}}$

$= \frac{2^{14}}{2^{8}}$

$= 2^{6}$ Lines

Tag Directory size = No of Tags × Tag size

= Number of Lines in cache × No of bits in Tag

$= 2^{6} \times 9$ bits

$= 64 \times 9$ bits

M.S

= 576 bits

= 72 bytes

1 byte = 8-bit

576 bit = $\frac{576}{8}$

= 72 bytes

Tag directory size = 72 bytes

Ex3 Consider a fully associative mapped cache of size 512 kB with Block size 1 kB. There are 17- bits in the Tag. find out the size of main memory and Tag directory size.

Sol give

Cache memory size = 512 kB

Block size = frame size = Line size = 1 kB

Number of Bits in Tag = 17 bits.

Number of Bits in Block offset!

Block size = 1 kB

$= 2^{10}$ bytes

Number of bits in block offset = 10 bits

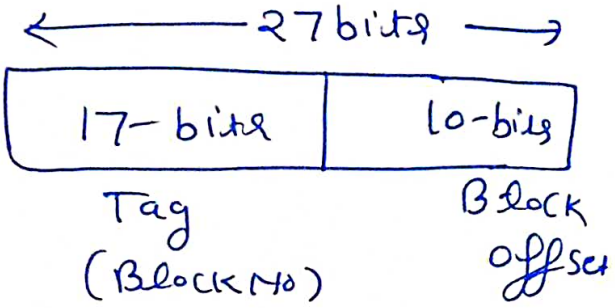
M.S

Number of Bits in Physical Address =

No of bit in Tag + No of bits in Block offset

= 17 bits + 10 bits

= 27-bits



Size of main memory :-

Number of Bits in Physical Address = 27 bits

Size of main memory = 2^{27} bytes

= 128 MB

Number of Lines in cache = $\frac{\text{Cache Size}}{\text{Line size}}$

= $\frac{512\ KB}{1\ KB}$

= 512 lines

= 2^9 lines

M·S

Tag directory size!-

Tag directory size = No of Tags x Tag size

= No of line in cache x No of Bits in Tag

$= 2^9 \times 17$ bits

= 8704 bits

= 1088 bytes

Size of Tag directory = 1088 bytes

Ex3 Consider a fully associative mapped cache with block size 4KB. The size of main memory is 16 GB find the number of bits in Tag.

Number of bits in Tag = 22-bits

Practice Problems

SET ASSOCIATIVE MAPPING:-

Ex-1 Consider a 2-way set Associative mapped cache of size 16 KB with block size 256 bytes. The size of main memory is 128 KB find the Number of bits in Tag and Tag directory size.

Sol given

Set size = 2

Cache memory size = 16 KB

Block size = frame size = Line size = 256 bytes

main memory size = 128 KB

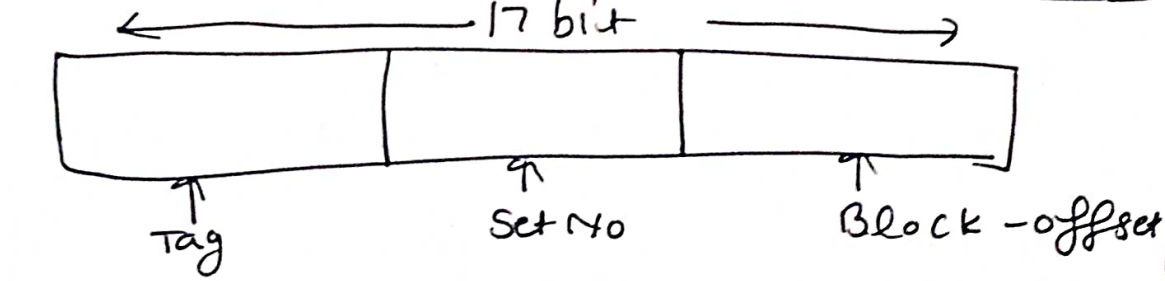
Number of Bits in Physical Address.

size of main memory = 128 KB

$= 2^7 \times 2^{10}$ bytes

$= 2^{17}$ bytes

No of Physical Address = 17-bits



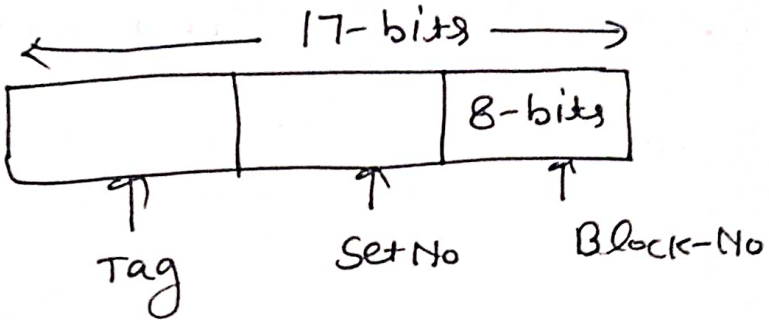
M.S

Number of Bits in block offset

Block size = 256 bytes

$= 2^8$ bytes

No of Bits in Block offset = 8-bits



No of Line in Cache

total Number of Lines in Cache = $\frac{\text{Cache size}}{\text{Line size}}$

$= \frac{16\ KB}{256\ \text{bytes}}$

$= \frac{2^4 \times 2^{10}\ \text{bytes}}{2^8\ \text{bytes}}$

$= 2^6$ Lines

$= 64$ Lines

No of line in cache = 64-Lines

M.S

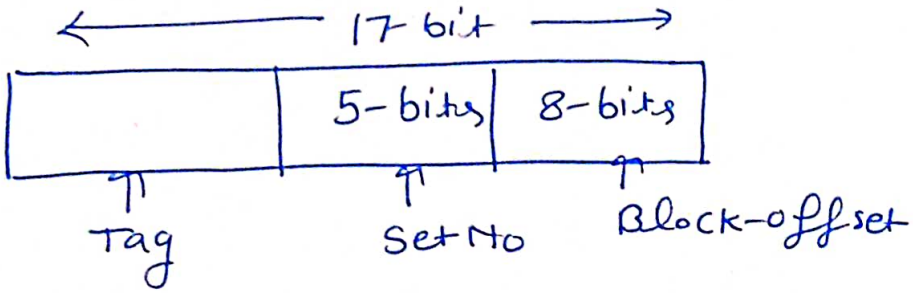
Number of sets in Cache = $\frac{\text{total No of Lines in cache}}{\text{set size}}$

$= \frac{64}{2}$

$= 32$ set

$= 2^5$ sets

Number of bits in Set No = 5- bits



Number of Bits in Tag :-

No of Bits in Tag = No of Bits in Physical Address - (No of bits in Set No + No of bits in block offset)

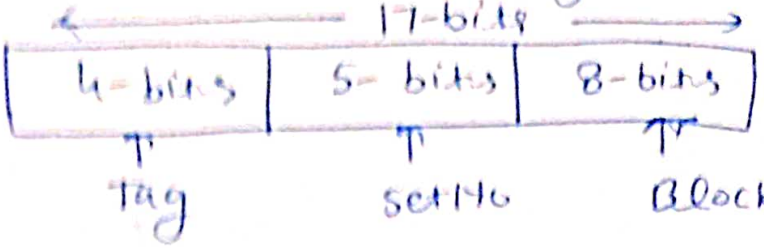
= 17- bits - (5 bit + 8-bits)

= 17- bits - 13- bits

= 4 bits

M.S

Number of bits in Tag = 4-bits



Tag Directory size:-

Tag directory size = Number of Tage x Tag size

= No of lines in cache x No of Bits in Tag

= 64 x 4-bits

= 256 bits

= 32 byte

Tag directory size = 32 bytes

Ans

M·S

Ex-3 Consider a 8-way set Associative mapped cache of size 512 KB with block size 1KB. There are 7-bits in the Tag find out the size of main memory and Tag directory size.

Sol given

Set size = 8

Cache memory size = 512 KB

Block size = frame size = Line size = 1KB

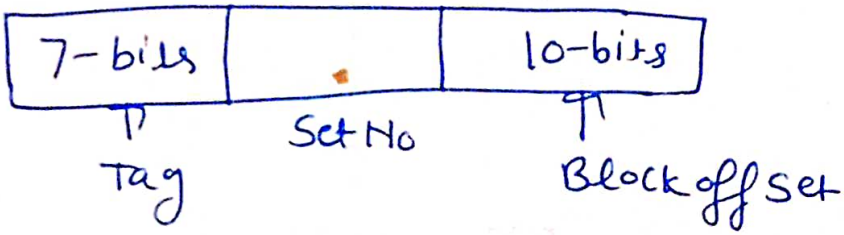
Number of bits in Tag = 7-bits

Number of Bits in Block offset -

Block size = 1KB

$= 2^{10}$ bytes

Number of bits in block offset = 10-bits



M·S

Number of Lines in Cache = $\frac{\text{Cache size}}{\text{Line size}}$

$= \frac{512\ KB}{1\ KB}$

$= 512$ lines

Number of lines in cache = 512-lines

Number of sets in Cache:

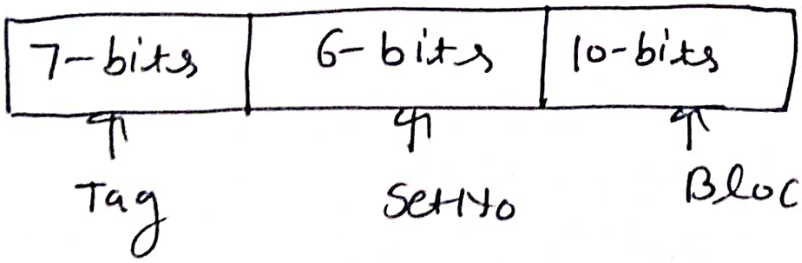
total No of sets in Cache = $\frac{\text{total No of Lines in cache}}{\text{set Size}}$

$= \frac{512}{8}$

$= 64$-Sets

$= 2^6$ sets

Number of Bits in Set No = 6-bits



No of Bits in Physical Address:-

M·S

No of Bits in Physical address = No of Bits in Tag + No of bits in set + No of Bits in offset

= 7-bits + 6-bits + 10-bits

= 23-bit

Number of Bits in Physical Address = 23-bits.

Size of main memory :-

Number of bits in Physical address = 23-bit
the size of main memory $= 2^{23}$ bytes
$= 8$ MB

Tag directory size :-

Tag directory size = No of Tags x Tag size

= No of lines in cache x No of bits in Tag

= 512 x 7-bits

= 3584 bits

Tag directory Size = 448 bytes

M.S

Ex-3 Consider a 4-way set associative mapped cache with block size 4KB. The size of main memory is 16 GB and there are 10-bits in the Tag find the size of cache memory and Tag directory size.

Sol given

Set size = 4

Block size = frame size = Line size = 4KB

main memory size = 16GB

Number of bits in Tag = 10-bits

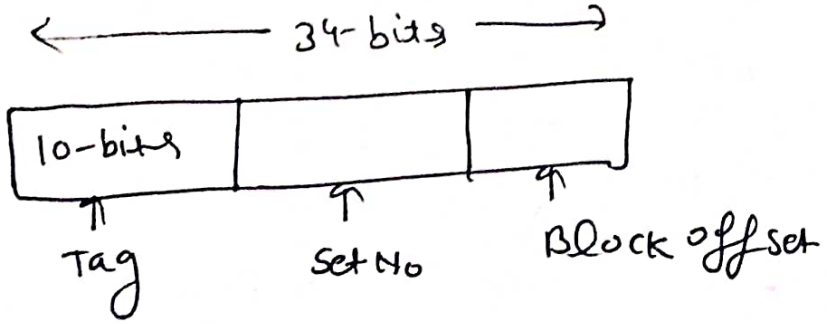
Number of bits in Physical Address:-

Size of main memory = 16 GB

$= 2^4 \times 2^{30}$ bytes

$= 2^{34}$ byte.

Physical address = 34-bits



1 GB = 1024 MB

1 MB = 1024 KB

1 MB = 10^{10} KB

1 KB = 10^{10} bytes

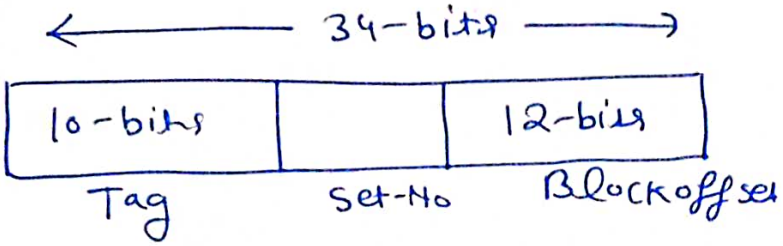
No of Bits in block off set :-

Block size = 4 KB

$= 2^2 \times 2^{10}$ bytes

$= 2^{12}$ bytes

Number of bits in block offset = 12-bits



Number of bits in set No :-

Number of Bits in set No = No of Bits in Physical Address - (No of bits in Tag + No of Bits in Block off-set)

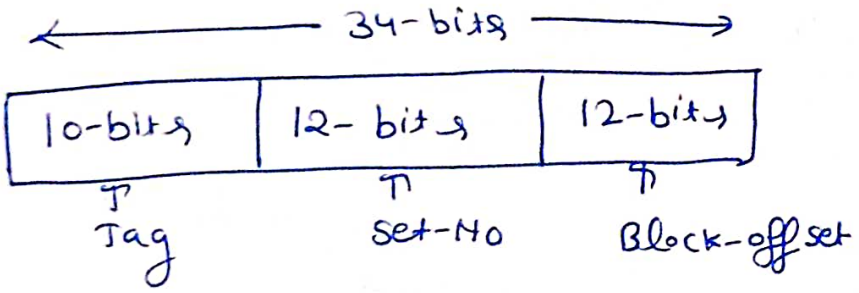
= 34 bits - (10 bits + 12 bits)

= 34 bits - 22-bits

= 12-bits

Number of Bits in set No = 12 bits.

M·S



Number of sets in Cache:-

Number of Bits in Set Number = 12-bits

total No of sets in Cache = 2^{12} sets

Number of Lines in Cache:-

total No of Sets in Cache = 2^{12} sets

Each set contains 4 Lines.

total No of Lines in Cache = Total No of Sets in Cache x No of Lines in each set

$= 2^{12} \times 4$ Lines

$= 2^{14}$ Lines

Size of Cache memory:-

M·S

Size of cache memory = total No of lines in Cache x Line size

$= 2^{14} \times 4$ KB

$= 2^{16}$ KB

$= 64$ MB

Size of cache memory = 64 MB

Tag directory size:

Tag directory size = No of Tags x Tag size

= No of Lines in cache x No of bits in Tag

$= 2^{14} \times 10$ bits

$= 16384$ bits

$= 20480$ bytes

$= 20$ KB

1 byte = 8 bits

Size of Tag directory = 20 KB

M·S

Ex4 consider a 8-way set associative mapped cache. The size of cache memory is 512 KB and there are 10-bits in the Tag. find the size of main memory.

Size of main memory = 64 MB

Ans

Practice Problem

Ex5 consider a 4-way set associative mapped cache. The size of main memory is 64 MB and there are 10-bits in the Tag. find the size of ~~main~~ cache memory.

Size of cache memory = 256 KB

Ans

Practice Problem

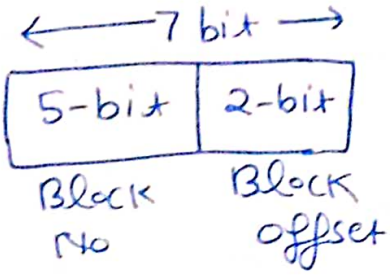
M·S

Line size = block size

Range 0 – 127

$128 = 2^{7}$

total Number of Bit in Physical Address



Set Associative mapping (K-way mapping):-

It is a combination of direct mapping and associative mapping.

Set Associative mapping	=	Direct mapping	+	Associative mapping

In this cache memory is divided into set.

Set = groups of blocks

Blocks = group of words

$$\text{cache memory Set No} = \left(\frac{\text{main memory}}{\text{block No}}\right) \bmod \left(\text{No of sets in cache memory}\right)$$

M.S