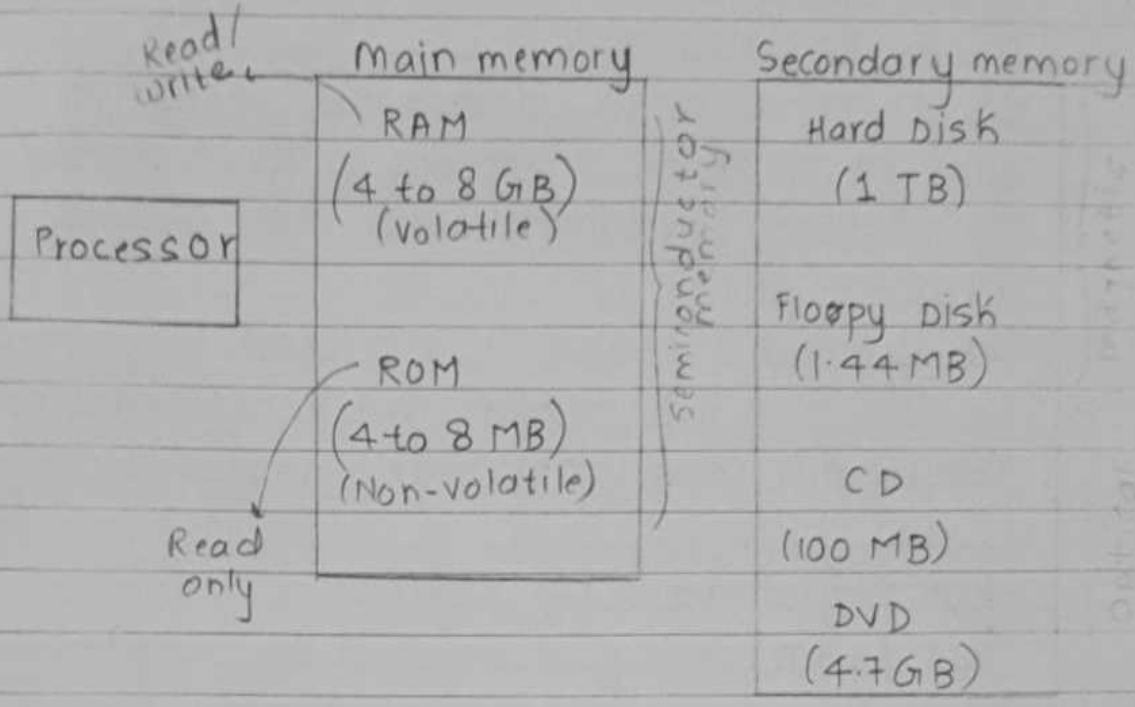
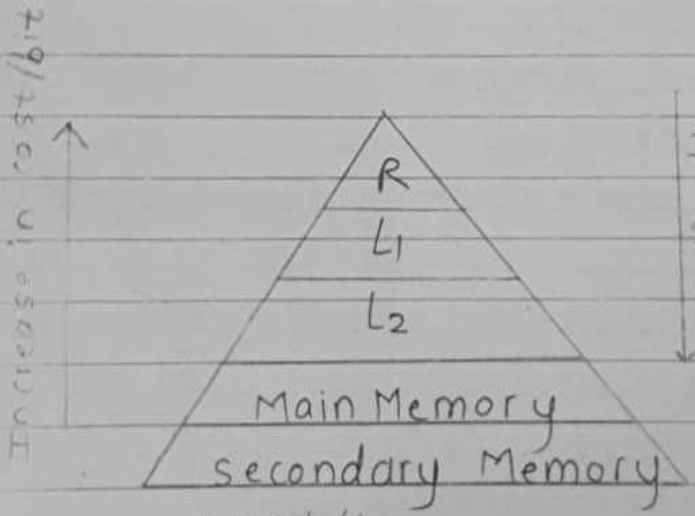
4. Memory Organisation

Memory Hierarchy –





* cost/bit
* Access time
* capacity / Storage space

- To match the processor speed the memory holding programs and data must be accesible in 1 nano second or less.
- The challenge is to design the overall memory system so that it appears to have speed of the fastest component and cost of the cheapest one.
- The pyramidal shape is intended to convey the smaller size of memory unit near the top, and higher capacity memory near the bottom.

Necessity of memory hierarchy -
- Memory hierarchy is required to maintain in order to reduce average cost per bit of entire memory system.
- To maintain average data transfer rate of memory system.
- To provide data recovery incase of data corruptio

Description of main, secondary memory in module 1

Memory characteristics:-

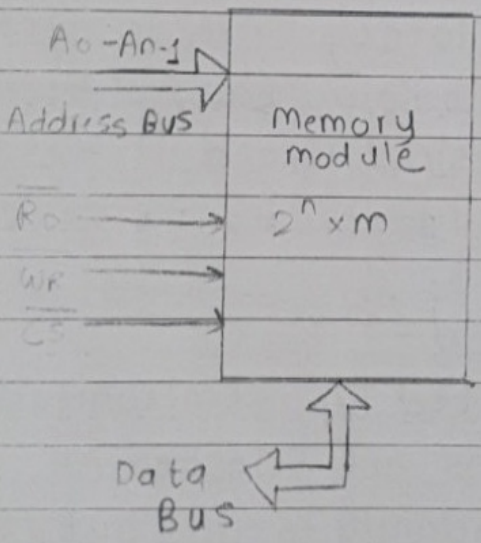
1) Location -
on-chip memory -
- The memory unit located inside the CPU is called on-chip memory.
eg. L1 cache, internal registers.

Internal memory -
- Present on mother board
eg. L2 cache, main memory

This memory is directly accessible to the CPU.

External memory -
- Only connectors are present on the mother board. It provides back storage.
  eg. CD, pendrive.

2] Capacity -
- The capacity of any memory system is represented by $2^n \times m$ bits where,
  2^n → memory requires a_0 to a_{n-1} address lines.
  m (word size) → memory requires d_0 to d_{n-1} data lines



3) Access Method-

Sequential method-
- It has been supported by magnetic tape systems in which, to reach desired record the computer may have to read all previous records.
- As a result it provides very low data transfer speed.

Direct access method-
- It has been supported by floppy discs, HD, CDs, DVDs, in which the read/write head can reached to desired record directly by involving seek motion and rotational latency time where seek motion is necessary for bringing read/write head on desired track number, where as rotational latecy time is necessary to bring desired sector below read/write head.
- After reaching to desired record it has to read information sequentially.

Random access method-
- It has been supported by semiconductor memory in which the locations are selected or identified by their address randomly to carry out read/write operation.
- In this random access memory between 2 successive read/write operation significant amount of processor and memory time will be wasted to calculate the successive address and to identify successive locations. As a result data transfer rate of RAM is comparitively low.

Page No.:
Date:

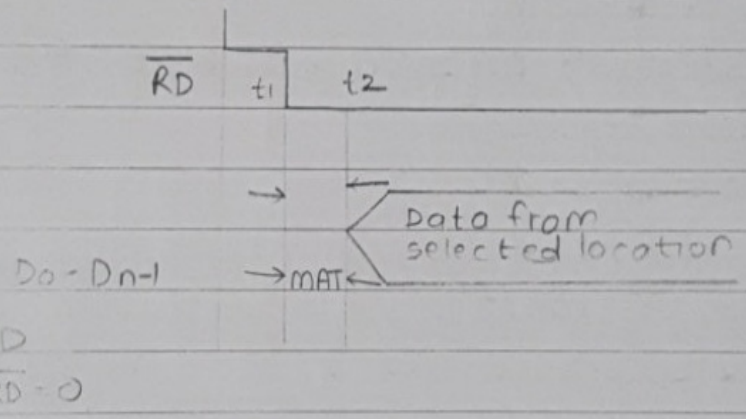
Associative access method-

- The memory system in which search is initiated by the data associated with CPU's ~~memory~~ requirement is called associative memory.
- In this memory, search is initiated by content rather than address due to which it is also known as content addressible memory.

4] Performance -

MAT (Memory access time)-

- It defines the time gap between ~~read~~ READ signal initiated by CPU by making $\overline{RD}$ - 0 and actual data is placed on data bus $D_0 \cdots D_{n-1}$ of memory system as shown below.



MCT (Memory Cycle Time)-

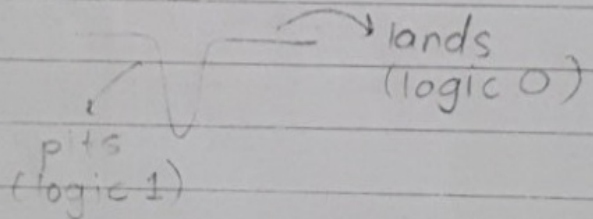
- The time that must be elapsed between 2 successive read/write operation ~~is called~~ so that memory can respond properly is called MCT.
- Data transfer rate is inversely proportional to MAT and MCT.

5) Physical Type -

Magnetic -
- It stores binary bits in terms of direction of magnetization of tiny particles of iron oxi on magnetic tape.

Semiconductor -
- It stores binary bits in terms of voltage / charge on capacitor.
- In flip-flops stable ~~or~~ and unstable state or reset and set state represent logic 1 and 0

Optical -
- It stores binary bits in terms of pits and lands which are created during burning of the CD by laser beam.
- The physical appearance of pits and lands is as shown below -



6) Physical Characteristics -

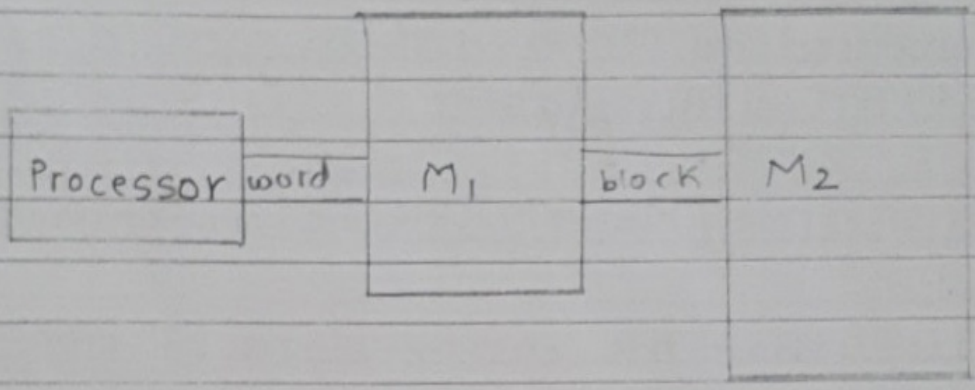
Volatile (erasable) -
- As the name suggests ~~it~~ the contents of volatile memory is lost when power is turned OFF. It requires continuous power supply eg. RAM

Non-volatile (Non-erasable) -

- (content) is retained even when power is turned off.

eg. ROM

Performace characteristics of two level memory-



$c_1 \rightarrow$ cost/bit $\quad c_2 \rightarrow$ cost/bit

$s_1 \rightarrow$ capacity of M_1 $\quad s_2 \rightarrow$ capacity of M_2

$t_{A_1} \rightarrow$ Access time for M_1 $\quad t_{A_2} \rightarrow$ Access time M_2

- M_1 is present on motherboard (main memory) and is physically connected to system bus of processor.
- M_2 is not present on motherboard only its connectors are present (secondary memory).
- whenever processor requires any data it first check in M_1 (primary memory) and if the data is found in M_1 its called HIT. Then operation is performed at M_1 itself as it is faster.

1) t_{A1} — time required to access M_1 (primary) memory.
- time required to transfer data from primary memory to processor.
- time[M_1 → ~~memory~~ processor]

- If the data is not found in M_1, it is called as MISS.
- The block containing that data is first transfered to M_1 and then data is operated from M_1 as it is faster.

2) t_{A2} — time [$M_2 \rightarrow M_1$] + time [$m_1 \rightarrow$ processor]

- Notice that the entire block is transferred as remaining data from the same block may be required in future.
- Such access of further data will be hit.

3) Hit ratio -

$$H = \frac{N_1}{N_1 + N_2}$$

N_1 – no. of hits
N_2 – no. of miss

4) Average access time -

$$t_A = (H \times t_{A1}) + (1-H)\, t_{A2}$$

HIT MISS

$t_A =$ ~~$A_1 \times H_1 + N_2 \times H_2$~~

~~$A_1 H_2$~~

5.) Average Cost (C) -

$$C = \frac{C_1S_1 + C_2S_2}{S_1 + S_2}$$

6.) Average efficiency (E) -

$$E = \frac{tA_1}{tA}$$

7.) Speed Ratio (R) -

- How slow M_2 is as compared to M_1

$$R = \frac{tA_2}{tA_1}$$

8.) Reliability - (MTTF)

- It measures the mean time to failure.

Q1.] For given two level memory:-

$tA1 \rightarrow 10^{-7}$ sec

$tA2 \rightarrow 10^{-2}$ sec

$E \rightarrow 0.9$

$H \rightarrow ?$

Soln:-

$$E = \frac{tA_1}{tA}$$

$$tA = \frac{tA1}{E} = \frac{10^{-7}}{0.9} = 1.11 \times 10^{-7}$$

$$tA = H + tA1 + (1-H)\,tA2$$

$$1.11 \times 10^{-7} = H + 10^{-7} + (1-H) \times 10^{-2}$$

$$H = 0.9999$$

2.] For M_1

$S_1 \rightarrow 2^{10}$ (1KB)

$tA_1 \rightarrow 10^{-8}$ sec

$C_1 \rightarrow 0.1$

$H \rightarrow 0.9$

For M_2

$S_2 \rightarrow 2^{16}$ (64KB)

$tA_2 \rightarrow 10^{-6}$ sec

$C_2 \rightarrow 0.01$

Calculate avg. cost and avg. tA

Soln:-

$$C = \frac{C_1S_1 + C_2S_2}{S_1 + S_2}$$

$$= \frac{0.1 \times 2^{10} + 0.01 \times 2^{16}}{2^{10} + 2^{16}}$$

$$= 0.0113$$

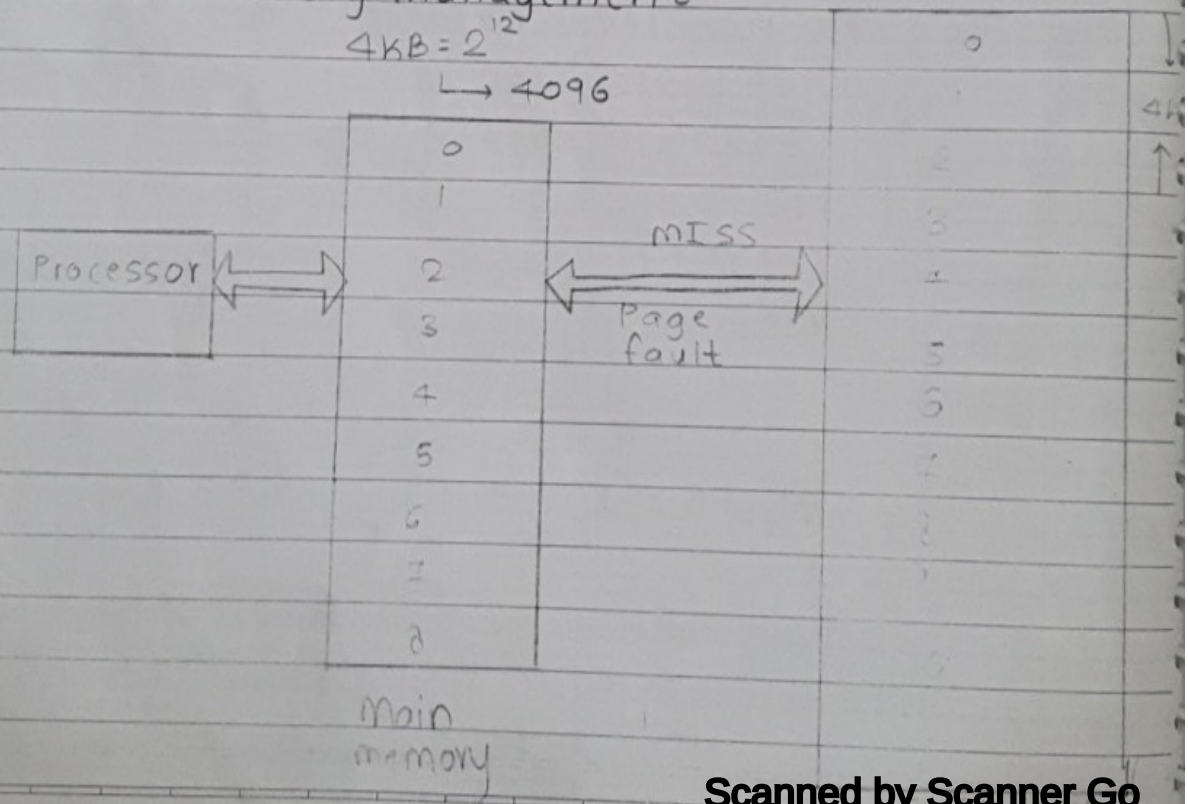
$$tA = H \times tA_1 + (1-H)\,tA_2$$

$$= 0.9 \times 10^{-8} + 0.1 \times 10^{-6}$$

$$tA = 1.09 \times 10^{-7} \text{ sec}$$

$$tA = 0.109\ \mu\text{sec}$$

Virtual memory management:-



- The need of introducing virtual memory in the system was to increase memory size.
- main memory size is restricted by the size of address bus (n bits of address bus can generate 2^n locations). eg. 80386 processor –

32 bit address line

$\therefore 2^{32}$ = 4GB main memory size

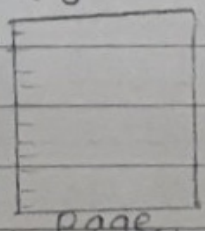
(2^{46})

But it can also access 64TB of virtual memory.

- Virtual memory is implemented by use of cheaper storage technology i.e. magnetic hard disk and optical CD ROM / DVD. Hence we can get a large amount of memory at low cost
- Whenever CPU wants data from virtual memory the data is first copied from virtual memory to main memory
- Instead of copying only required data the entire block containing the data is copied so that further access will be from main memory directly.
- To implement this there are 2 techniques –
  1. Paging
  2. Segmentation.

Paging –

- Here virtual memory space is divided into equal size pages (generally 4KB) (0 to 4095 locatio



- The main memory space is divided into equal size page frames.
- Each frame of main memory can hold any page from virtual memory.
- When the CPU wants to access a page, it first looks at the main memory.
- HIT - If the required page is found in main memory it is called as HIT. Further operation will take place from main memory to processor.
- Page fault - If the CPU needs a page that is not present in main memory then it is called as page fault. Now the new page has to be loaded from virtual memory to main memory.

Page Replacement -
- After a page fault, if there are no empty frames in the main memory then an old page from main memory is removed and the new page is inserted. This is called as page replacement.

① FIFO (First In First Out) -
- In this in case of page fault, the oldest page will be replaced with the new page in the main memory.
- The page which comes first is replaced first.

② LRU (Least Recently Used) -
- For a longer time if CPU has not used a particular page, that is the page which will go out first and get replaced by new one.

(3) LFU (least Frequently Used)-

- CPU has to keep a count of the no. of times a page is used.
- The least frequently used page is replaced with the new page.

Dirty page / Dirty bit -

- During replacement if the old page has been modified in the main memory then it needs to be first saved into the virtual memory and then replaced.
- The CPU keeps track of all such updated pages by maintaining a dirty bit for each page.
- When the page is updated in the main memory, dirty bit is set to 1 and then page is said to be a dirty page.
- These dirty pages are first copied into virtual memory and then replaced.

Thrashing -

- When the program causes a lot of continuous page fault it said to be thrashing and it should be avoided. ~~Ways to avoid it-~~

Demand paging-

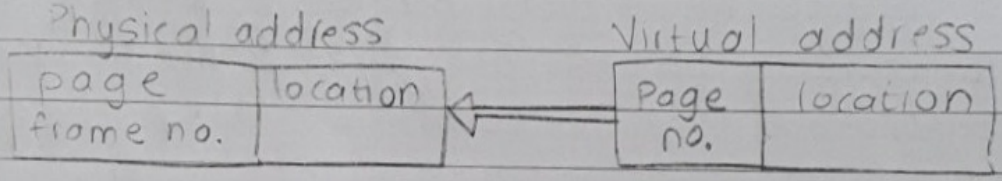
- If the pages are loaded into main memory only when required by the CPU then it is called demand paging.
- Thus pages are loaded only after page fault

Working set model –

- This method is opposite to demand paging.
- Here the pages required by application are grouped together as its working set model and whenever the application loads, its entire working set is loaded into the main memory.
- Hence application will not cause any page fault while running and hence runs faster.
- But it may tend to monopolise the main memory space as all its pages are loaded and it leaves no space for other programs.

Address Translation –

- It is the mechanism used for finding main memory location from virtual memory location.



Virtual address → page table → physical address.

Page table –

- Page table contains the info. about which page frame number holds which page no. from secondary memory.

Translation look aside buffer (TLB) –

- This is an onchip buffer within the CPU used to speed up the paging process. ~~Once the~~ since a page from virtual memory can get stored into any frame of main memory the OS must store

a page table which indicates which page of virtual memory is stored in which frame of main memory.
- Hence for accessing the page the CPU has to do 2 memory operations.
- First access the page table to get information about where the page is stored in the main memory and then access the main memory for that page.
- To solve this problem the CPU copies the page table information of the most recently used pages in the on chip TLB.
- Thereafter any subsequent access to the page will be faster as the information will be provided by TLB and CPU need not access page table.
- CPU starts 1st searching into TLB for required page no. but simultaneously it starts searching in page table as well.
- If the required page frame no. is not available in TLB processor can continue the search in page table without wasting its time.

Segmentation:-
- Here the memory space is divided into logically different areas called segments.
- This makes the memory well organised and hence easy to use for programmer.
- eg. 8086 has 4 segments - code, data, stack, extra
- Programmer is aware of segments and can relocate the segments during the program.

- Segments don't have a fixed size and hence unnecessary wastage of space called fragmentat of main memory is prevented.
- Here the best fit algorithm is widely used due to the flexible size of segments.
- And address is given as combination of segment address and offset address.
- Segment address is the starting address of segment.
- offset address is the address of current location within the segment.

Paging	Segmentation
- Pages are of fixed size.	- There is flexibility in size of segment depending on user.
- Paging leads to fragmentation of main memory.	- Fragmentation of main memory is prevented.
- Programmer is unaware of paging.	- Programmer is aware of different segments.
- This uses LRU, FIFO, LFU, Optimal.	- This uses best fit method, first fit method.
- Paging has physical boundaries.	- Segmentation will have logical boundaries.
- Paging is difficult to implement.	- Segmentation is easy to implement.
- Programmer cannot differentiate between code and data.	- Programmer can easily differentiate between code and data.

Page replacement algorithms -

Q] Main memory has 3 pages and the processor requires pages from virtual memory in the following order.

2 3 21 5 2 4 5 3 2 5 2

Show the implementation of FIFO, LRU, LFU and optimal and comment on which is the best suitable algorithm for the given string.

1] FIFO

FIFO	2	3	2	1	5	2	4	5	3	2	5	2
1	2*	2*	2*	2*	5	5	5*	5*	3	3	3	3
2		3	3	3	3*	2	2	2	2*	2*	5	5
3				1	1	1*	4	4	4	4	4*	2*
			H					H		H		

$$H = \frac{\text{No. of hits}}{\text{No. of attempts}} = \frac{3}{12}$$

2] LRU

LRU	2	3	2	1	5	2	4	5	3	2	5	2
1	2	2	2	2	2	2	2	2	3	3	3	3
2		3	3	3	5	5	5	5	5	5	5	5
3				1	1	1	4	4	4	2	2	2
			H			H		H			H	H

$$H = \frac{5}{12}$$

3]

LFU	2	3	2	1	5	2	4	5	3	2	5	2
1	2(1)	2(1)	2(2)	2(2)	2(3)	2(3)	2(3)	2(3)	2(3)	2(4)	2(4)	2(4)
2		3(1)	3(1)	3(1)	5(1)	5(1)	5(1)	5(2)	5(2)	5(2)	5(3)	5(3)
3				1(1)	1(1)	1(1)	4(1)	4(1)	3(1)	3(1)	3(1)	3(1)
			H			H		H		H	H	H

$$H = \frac{6}{12} = 50\%.$$

4] Optimal

	2	3	2	1	5	2	4	5	3	2	5	2
1	2	2	2	2	5	5	5	5	5	5	5	5
2		3	3	3	3	3	3	3	3	3	3	3
3				1	1	2	4	4	4	2	2	2
			H					H	H		H	H

Q] FIFO

	6	0	12	0	30	4	2	30	321	20	15
1	6*	6*	6*	6*	30	30	30*	30*	321	321	3
2		0	0	0	0*	4	4	4	4*	20	2
3			12	12	12	12*	2	2	2	2*	15
				H				H			

$$H = \frac{2}{11}$$

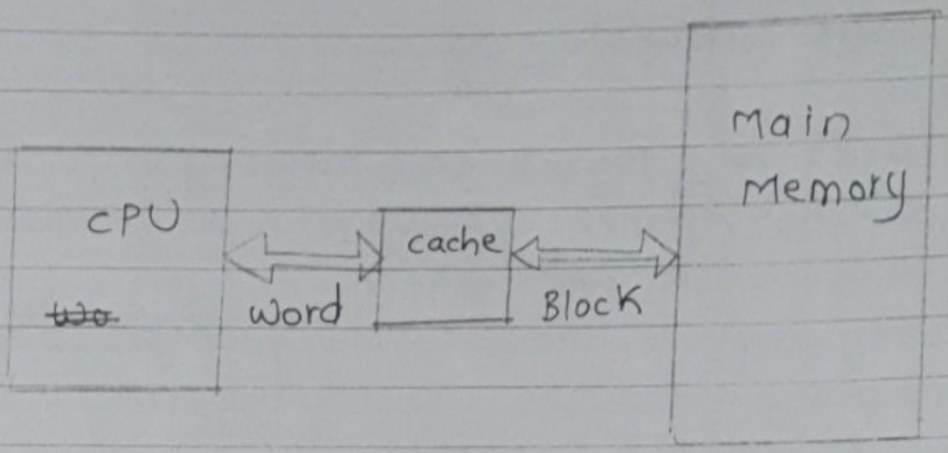
LRU

	6	0	12	0	30	4	2	30	321	20	15
1	6	6	6	6	30	30	30	30	30	30	15
2		0	0	0	0	0	2	2	2	20	20
3			12	12	12	4	4	4	321	321	321
				H				H			

$$H = \frac{2}{11}$$

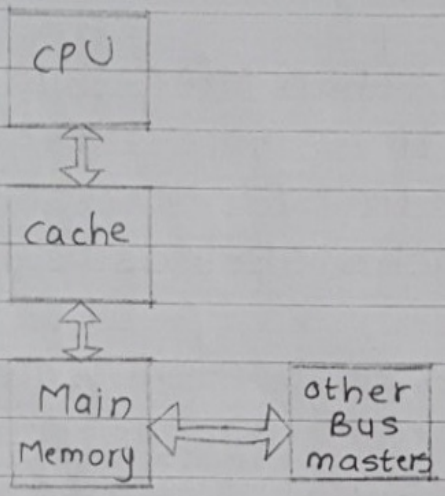
Cache Memory –

- Cache is a high speed memory introduced in the system to improve its performance as processor becomes faster.
- They require faster memory chips otherwise memory operation would require the CPU to wait. (wait states)
- Cache memories were introduced to avoid these wait states.
- Cache is implement using SRAM (Static RAM) while the main memory uses DRAM (Dynamic RAM).
- SRAM is much faster than DRAM but takes a lot more space and it is more expensive and also more power consuming. Hence only a little amount of cache should be implemented in the system.
- When the CPU wants to perform a memory operation it is performed on cache itself.
- If the data is not found in cache we call it as MISS.
- Now the block containing the data is copied from the main memory to the cache memory.
- If required data is found in cache it is a HIT.
- HIT Ratio is the ratio of the no. of hits to the total no. of attempts.
- During the miss if there is no empty block in cache memory then the same replacement policies such as FIFO, LRU, LFU will be used.

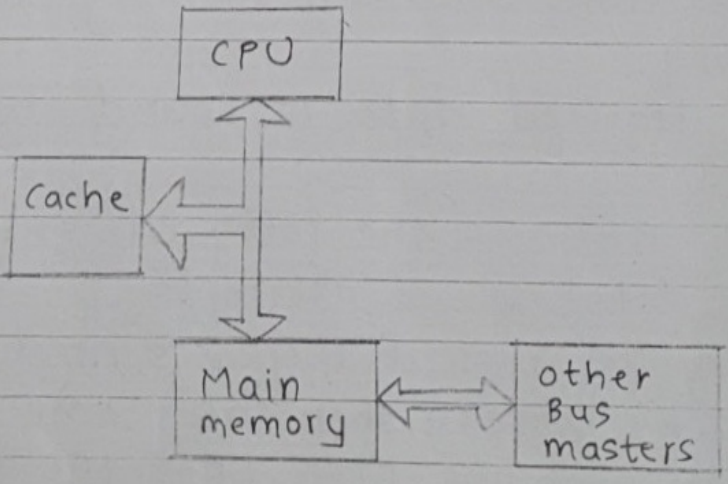


Cache Architecture / layout -

1.] Look Through Cache



2.] Look aside cache



Page No.:
Date:

look through cache-

- Here the CPU will first search the cache to find the required data.
- If it is found (its a HIT) the transfer is performed between CPU and cache memory itself.
- But if there is a miss then the CPU will repeat the search now in main memory.

Advantage-

- During the hit, ~~ma~~ as the transfer is between CPU and cache memory the main memory is free to be used by other bus masters.
- This is called concurrency. It improves the system performance.

Disadvantage-

- During the miss, time is wasted in searching the cache memory first and then searching the main memory. This is called look up penalty.
- It lowers the system performance.

Use-

- It is preferred in multi processor systems due to advantage of concurrency.

Look aside Cache-

- Here CPU will simultaneously start search in main memory and cache.

Advantage-

During a miss there is no look up penalty as search was performed simultaneously in cache and main memory.

Disadvantage-
As the CPU is always searching in the main memory, other bus masters can't access CPU at the same time hence concurrency is not possible.

Use-
Preferred in uni-processor system as it is simple and no look up penalty.

Cache Consistency / Coherance-

To avoid inconsistency following policies are used -
1] Write through policy
2] Write back policy / Buffered write / Delayed.
3] Snoopy writers.

- When the data of location in the cache is different from the data of the same location in the main memory then cache is said to be inconsistent.
- Inconsistency may arise when the CPU writes into a location in the cache memory and now if any other bus masters access the same data from the main memory it will get old or stale data.
- To avoild inconsistency the write policies are used.

1) Write through policy -
- Here whenever the CPU writes into cache memory it will also perform the write operation on the main memory.
- Hence the main memory will always contain the updated data.
- As an advantage there is a high level of consistency.
- It is very simple to implement.
- But write operation will be slow as CPU writes into both cache and main memory.

2) Write back / Buffered write / pelayed write -
- Here CPU only writes into cache memory.
- Additionally there is a cache controller provided in the system.
- The cache controller keeps a track of all locations that are updated in cache memory.
- This is done by maintaining changed bit for each cache block (similar to dirty bit in paging).
- Whenever it finds that CPU is free or that the block will be removed from the cache, it copies the updated contents from cache to main memory.
- This has an advantage that since the CPU does only one write operation, write operations are faster.
- Disadvantage is that the main memory does contain stale data till it has been updated which may happen after a long time. Hence low level of consistency.

3) Snoopy writes -

- This is a very efficient method of updating main memory.
- Here the cache controller snoops (monitors) the activities of other bus masters on the system bus.
- If it finds bus master is trying to read location from the main memory that has been modified in cache memory, it will first copy the updated data from cache memory to main memory and then only allow other bus master to access the data from main memory.
- The performance of such system is very high as no unnecessary updates are performed.
- other bus masters will always get updated data.
- Disadvantage is that advance and expensive cache controller would be required which will increase cost and complexity of the system.

Cache mapping Techniques:-

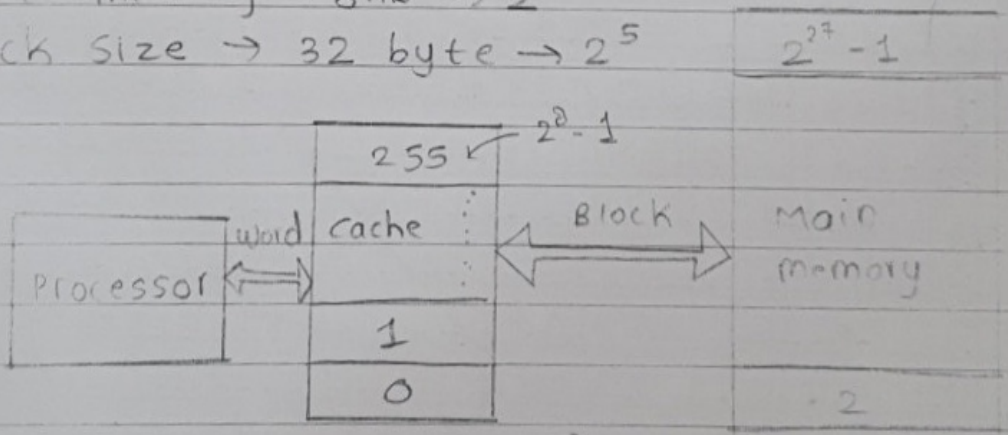
1) Associative Mapping (Fully-Associative mapping)
2) Direct Mapping (one-way set associative)
3) Set associative mapping (Two-way set associative)

- Blocks are loaded from main memory to cache memory.
- Cache mapping decides which block of main memory comes into which block of cache memory.
- There are several mapping techniques trying to balance between hit ratio, search time and tag size.

1) ~~Associative Mapping~~

eg. Pentium Processor :-

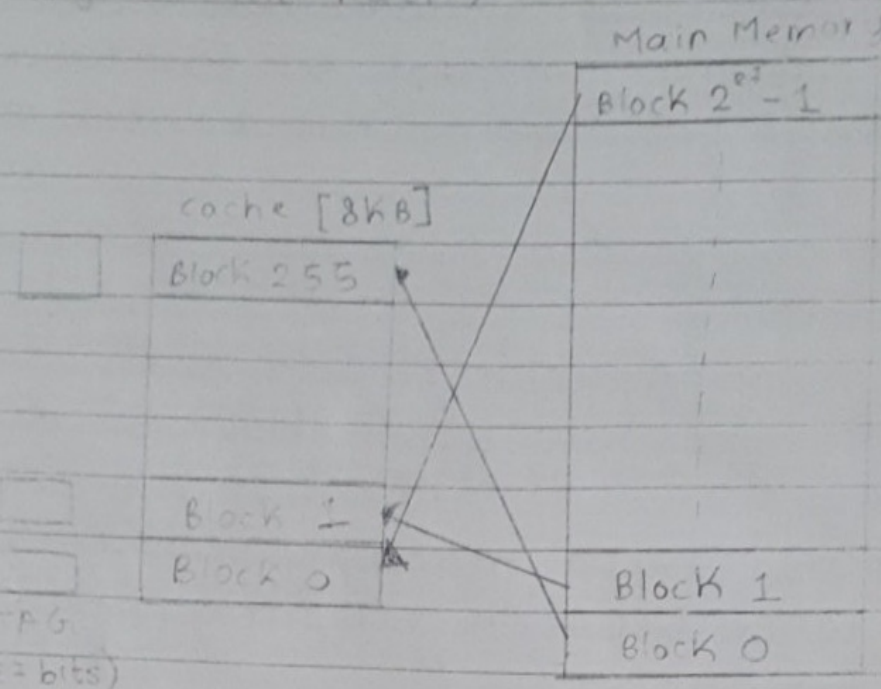
Main memory → 4GB → 2^{32}
Cache memory → 8KB → 2^{13}
Block size → 32 byte → 2^5



No. of blocks in cache = $\frac{2^{13}}{2^5}$
$= 2^8$
$= 256$

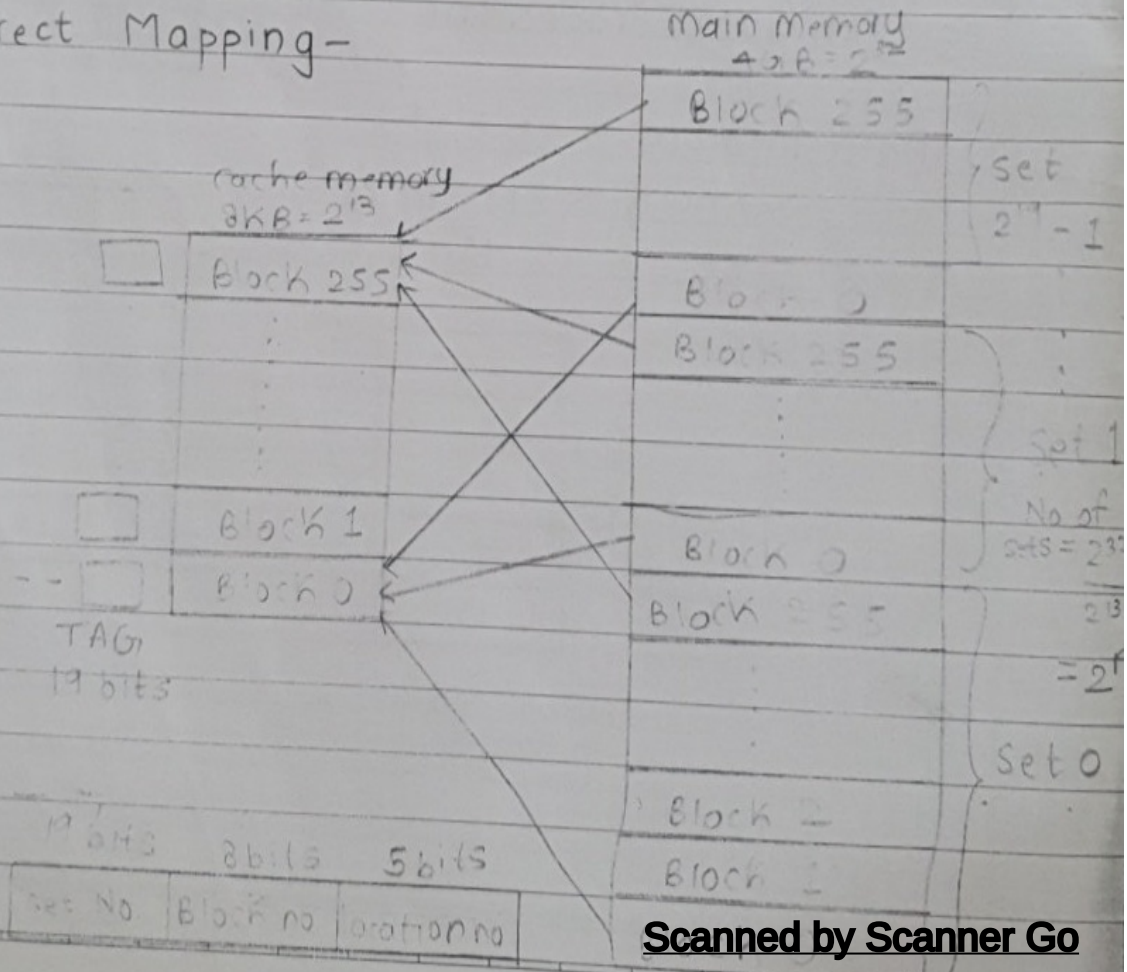
No. of blocks in main memory = 2^{32}

1) Associative Memory -
(Highest Hit Ratio)



27 bit	5 bit
Block No	location

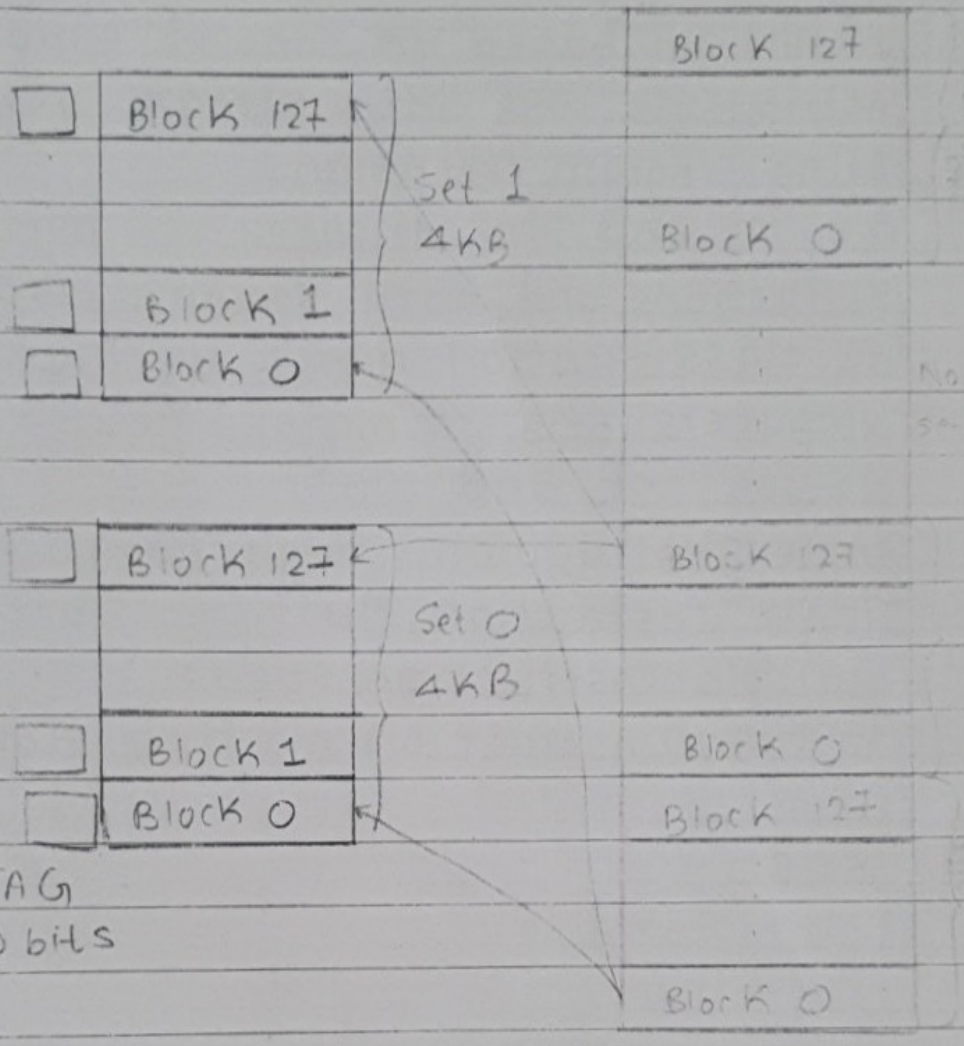
2) Direct Mapping -



3) Set Associative mapping-

No. of cache set → $2^{13}/2 = 2^{12}$ (4KB)

No. of block in each set = $2^{2}/2^{5} = 2^{7}$ (0-127)



20 bit	7 bit	5 bit
Set no.	Block no.	within block location