"Non-Linear Pipeline" :-

★ Non-Linear Pipeline contain forward and Backward connections so Reservation table is used to process the inputs in the Pipeline.

★ Let us consider the following Reservation table to process the inputs in a Non-Linear pipeline.

	1	2	3	4	5
S_1	X				X
S_2		X		X	
S_3			X		

★ multiple check-marks in the Row indicates that, Repeated use of same stage in different cycles

★ Contiguous check-marks in the Row indicate, extended use of same stage upto some number of cycles.

★ multiple check-marks in the colour indicate, parallel use of a different stages in the same cycle.

Latency Analysis :

★ Latency means Clock cycle difference between the two successive initiations in the Pipeline it is a positive Constant.

* Linear pipeline Latency is always '1' but Non-linear pipeline Latency will be depends on the reservation table.
* In the Non-~~Pipeli~~ linear pipeline some of the latencies causes the collision called 'forbidden latency' (Non-premissible latency).
* Some of the Lantencies does not causes collision named as Non-forbidden Latency (Premissible latency).
* To Detect the forbidden Latency we need to take the difference between any two check-marks in the Row.

eg:- Row 1. : (5-1) = 4 ← Latency {also forbidden latencies}
Row 2 : (4-2) = 2
Row 3 : 1

* Based on the forbidden Latency, collision vector is created that is:

$C_i : [c_n \ldots c_3 \ c_2 \ c_1]$

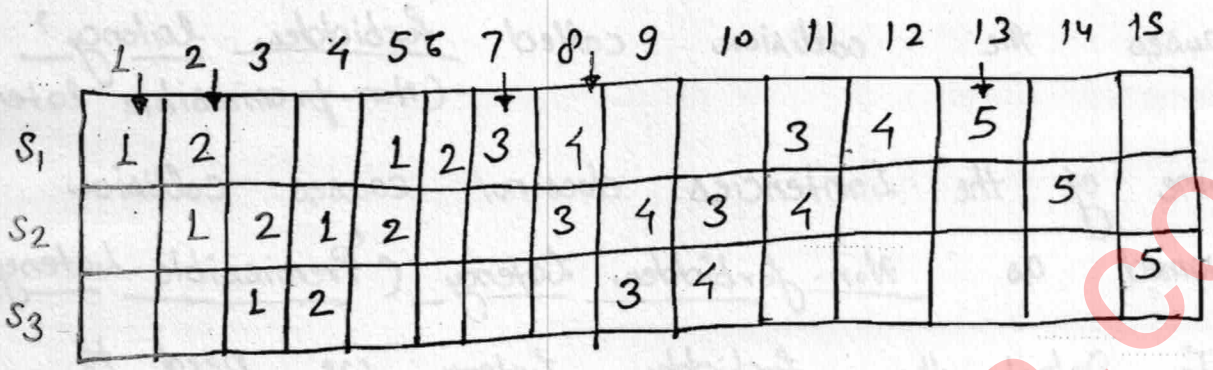
C → 0 : Premissable
C → 1 : Non Premissable

collision vector : $[c_5 \ c_4 \ c_3 \ c_2 \ c_1]$

$= [0 \mid 0 \mid 1 \mid 0]$
1

* Latency sequence is generated based on the collision vector to find out the Latency cycle. i.e.

1.) Latency sequence with respect to "C_1"

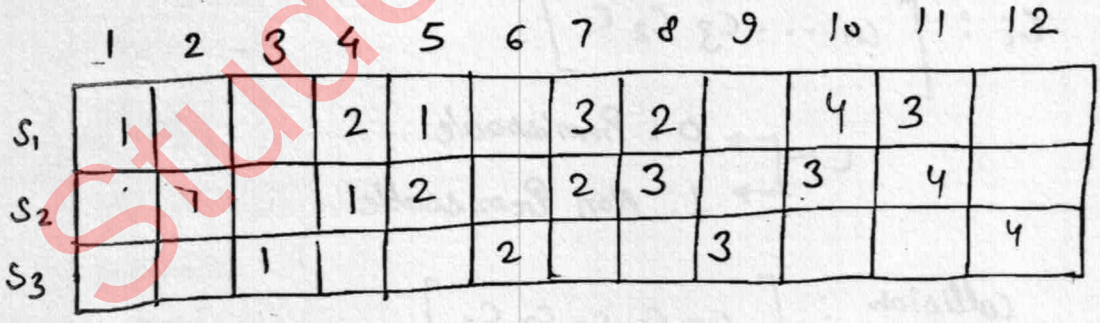


Latency cycle : (2-1) = 1
(7-5) = 5
(8-7) = 1
(13-8) = 5

(1,5), (1,5), (1,5) ...

Avg latency $\frac{(1+5)}{2} = 3$

2.) Latency sequence with respect to "C_3"



Latency cycle :-
(4-1) = 3
(7-4) = 3
(10-7) = 3

3, 3, 3, 3 ....

Avg latency = 3

③ Latency sequence w.r.to "C_5"

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S_1	1				1	Q	3			2	3	4	5		
S_2		1		1			2	3	2	3			4	5	4
S_3			1					2	3					4	5

Latency cycle :-

$(6-1) = 5$

$(7-6) = 1$

$(12-7) = 5$

$(13-12) = 1$

$(5,1), (5,1), (5,1) \ldots$

Avg latency $= \frac{5+1}{2} = \underline{(3)}$

MAL [Min Average Latency] $= \underline{3}$

"Memory organization" :-

Types :-

Based on the style of accessing the memory system, memory organization is two kinds :

1.) Simultaneous access m/y organization

2.) Heirarchical access m/y organization

- Simultaneous access m/y organization :-

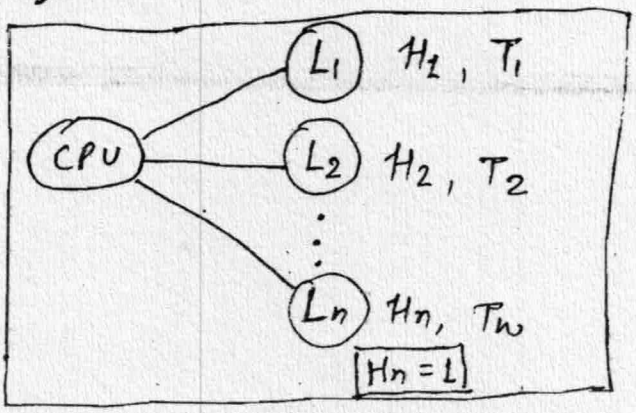
* In this organization, CPU directly connected to all the level of memories but accessing the memories in a sequence that is :

When there is a miss in level-'i' memory then

* CPU directly access the data from the level-'2' memory without copying into level-'i'

* When there is a miss in level-'2' m/y then CPU directly access the data from the level-'3' memory without copying into level '2' and level 'i'. and so on.

* In this memory organization data block is need not be required in the level-L memory



Here,

* $H_1, H_2 \ldots\ldots H_n \rightarrow$ Hit ratios

$H_1, H_2 \ldots H_n$ are the Hit ratios of respective memories,

$$\text{Hit Ratio, } H = \frac{\#\text{ Hits}}{\text{Total }\#\text{ Access}}$$

$$H = \{0 \text{ to } 1\}$$

Here

$T_1, T_2 \ldots\ldots T_n$ are the access times of a respective memories.

* Amount of the Time Required to access 1-word data from the memory is called as Average Access time that is,

$$T_{avg} = H_1 T_1 + (1-H_1) H_2 T_2 + (1-H_1)(1-H_2) H_3 T_3 + \ldots\ldots + (1-H_1)(1-H_2)\cdots(1-H_{n-1}) H_n T_n$$

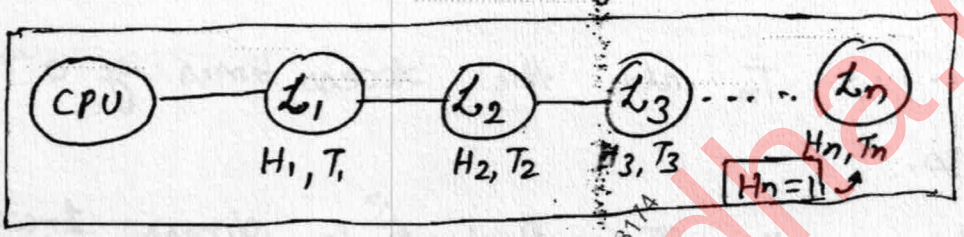
1-word takes — T_{avg}

? # words — 1 sec.

$$\eta_{memory} = \frac{1}{T_{avg}} \text{ words/sec.}$$

- Heirarchical Access Memory organization :-

* In this organization, CPU always accessing the data only from the level-1 memory. when there is a miss in level-'i' m/y then the Respective data-block is transferred from Higher-levels to lower level $\{L_i\}$. Later CPU accessing the data only from the level-'i' memory.



$$T_{avg} = H_1T_1 + (1-H_1)H_2(T_2+T_1) + (1-H_1)(1-H_2)H_3(T_3+T_2+T_1) + \cdots + (1-H_1)(1-H_2)\cdots(1-H_{n-1})H_n(T_n+T_{n-1}+\cdots+T_1)$$

$$\eta_{m/y} = \frac{1}{T_{avg}} \text{ words/sec.}$$

Code:

I_1 : Data (L_2 m/y)
I_2 :
I_3 : Data
I_4 :
I_5 : Data
I_6 :
I_7 : Data

} Reuse the data

Code:

Simultaneous	Hierarchical
$I_1 : ML_1, T_2$	$I_1 : ML_1, T_2 + T_1$
$I_3 : ML_1, T_2$	$I_3 : T_1$
$I_5 : ML_1, T_2$	$T_5 : T_1$
$I_7 : ML_1, T_2$	$T_7 : T_1$
I_9 ML_1, T_2	$T_9 : T_1$
5 times L_2 mfy is referenced	$(1 * T_2) + (5 * T_1)$

Reuse the Data Space } Locality of Reference

* In the Heirarchical memory design, Locality of reference is present so Avreage accepping time is minimized.

Locality of Reference means CPU performing the operations on a Re-usable data-space.

It is two kinds :-

1) Temporal Locality : means some word in the Data-Block will be referenced by the CPU again and again in near future.

2.) Spatial Locality : means Adjacent words in the Data-block is referenced by the CPU in a sequence.

* To satisfy the above locality of references we need to design the Block-size with multiple words.

Blocksize > wordsize.

Note:- In the computer system design m/y system designed using the Heirarchical principal to minimize the 'Speed gap' b/w. CPU & m/y.

Note:- when the Question contain Heirarchy word, Heirarchy meaning or cache memory word then use Heirarchical memory organization otherwise use Simultaneous m/y organization.

Que:- Consider 2-Level m/y organization where Level-1 memory is 12 times faster then level-2 m/y and its access time is 30 ns less than the avg access time. Let Level 1 m/y access time is 40ns what is the Hit ratio.

Level 1 = 12 Level 2 m/y

$T_{avg} = H_1 T_1 + (1-H_1) H_2 T_2$

$70 = (H_1 * 40) + (1-H_1) 1 * 480$

$H_1 = 0.93$

$L_1 \rightarrow H_1, T_1$

$L_2 \rightarrow H_2 T_2$

$H_2 = 1$

2 level m/y org.

$r = \frac{T_2}{T_1}$

$12 = \frac{T_2}{T_1}$

$T_2 = 12 T_1$

$T_1 = T_{avg} - 30ns$

$T_{avg} = T_1 + 30ns$

Let

$T_1 = 40ns$ then

$T_2 = 12 \times 40 ns$

$= 480$

$T_{avg} = 40 + 30$

$= 70ns$

Que:- 3-level m/y organization has the following specification:

Level	Access time/ word	Block size in words	Hit ratio	Block Access time
1	20 ns	—	0.7	$T_1 = 20ns$
2	100 ns	2	0.9	$T_2 = 200ns$
3	200 ns	4	1	$T_3 = 800$ ns

If the referred block is not in L_1, then Transfer it from L_2 to L_1. If Not in L_2 then L_3 to L_2 and L_2 to L_1, How long will it take to access the Data Block.

:- Heirarchical: 3 level

$$T_{avg} = H_1 T_1 + (1-H_1) H_2 (T_2 + T_1) + (1-H_1)(1-H_2) H_3 (T_3 + T_2 + T_1)$$

$$= 0.7 * 20 + (0.3) * 0.9 (220) + (0.3)(0.1) 1 (1020)$$

$$T_{avg} = 104 \text{ ns}$$

★ Que:- In a 2-level m/y design Level-1 m/y ~~access time~~ design Level-2 m/y access time is 100ns and its Bandwidth is 1 Gbps. Blocksize in the m/y design is 128 Bytes. How much time is required to access the block from Level-2 m/y.

$\frac{2^{30}}{2^{10}} = 2^{20} \times 100 ns = 2^{13} ns$

Total time = Access time + Transfer time (If B.w. given)

= 100 ns + 1GB → 1 sec.
128 B → ?

⇒ $\frac{128\ B}{1\ GB}$ sec.

= $128 * 10^{-9}$ sec.

= 100 ns + 128 ns

= 228 ns

"Memory Heirarchy design":-

★ In the computer system design memory system is organized by using the memory heirarchy policy.

According to a memory heirarchy design, system supported memory standards is as follows:

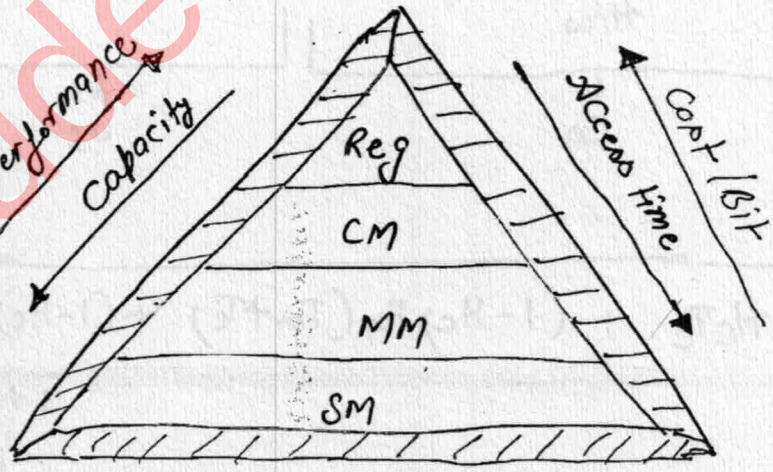
memory standards:

According to a M/y Heirarchy design accessing - sequence of a system supported memories is as follows:-

Level :	1	2	3	4
Name	Register (	cache memory (CM)	main memory (MM)	Secondry memory (SM)
Typical size :	< 1 KB	< 16 MB	< 16 GB	> 100 GB
Implementation :	customized multiports	SRAM (flip-flop)	DRAM (capacitors)	magnetic
Access time :	(0.25 - 0.5)	~~(500 - 1000)~~ (0.5 - 25)	(80 - 250)	50,00,000
Bandwidth (MB/sec) :	(20000 - 100000)	(5000 - 10000)	(1000 - 5000)	(20 - 150)
Managed by :	Compiler	H/w	OS	OS
Backed By :	CM	MM	SM	Compact Disk (CD)

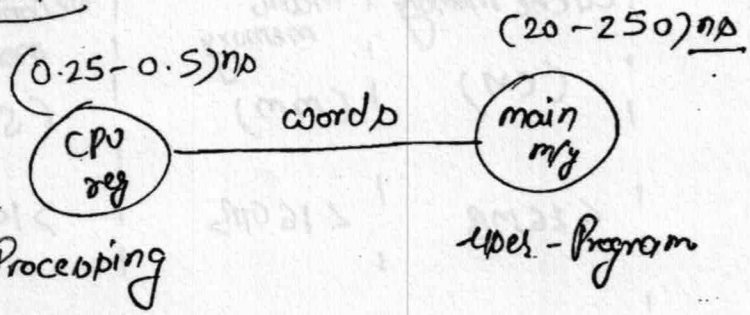
Bottom up approach :-

1) Capacity ↓
2) Access time ↓
3) Performance ↑
4) Cost/bit ↑
5.) Expensive ↑

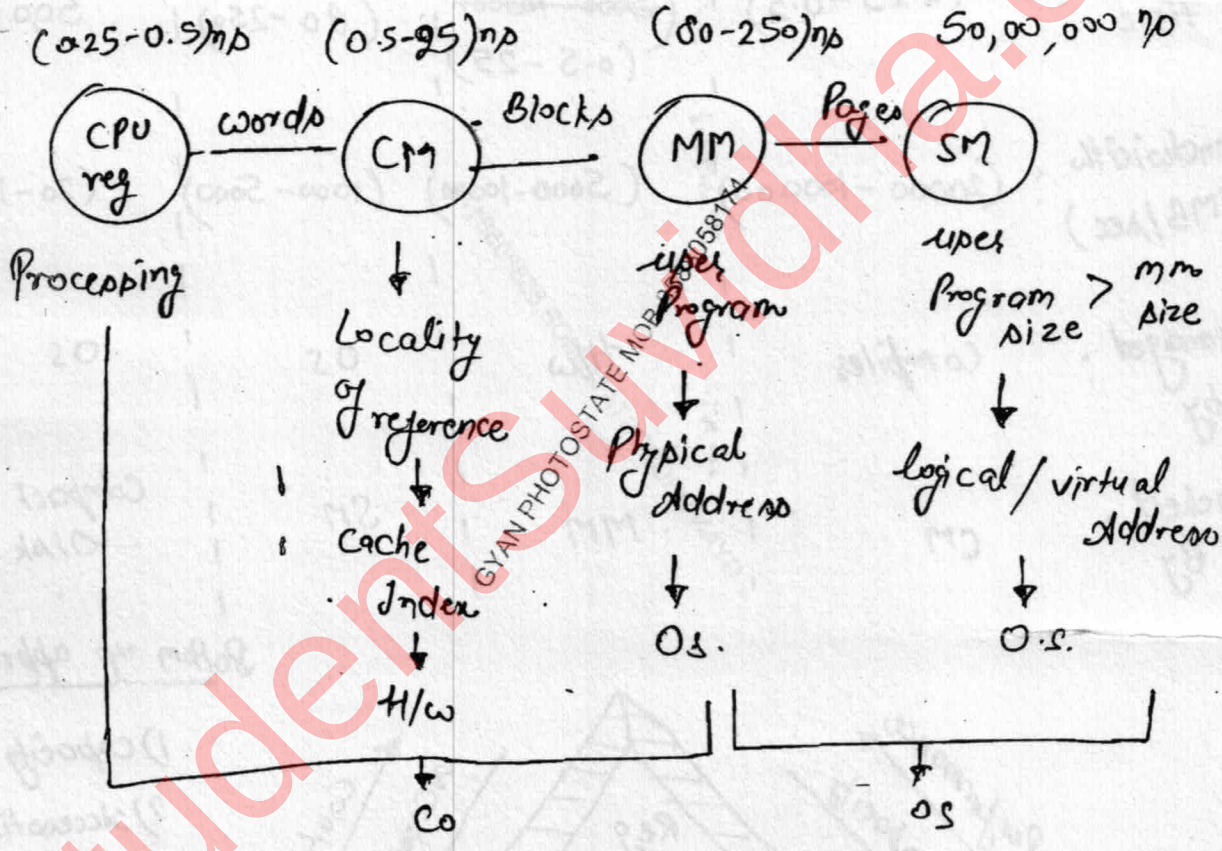


m/y Heirarchy Design

System old :



System New :



$$T_{avg} = H_c T_c + (1-H_c) H_m (T_m + T_c) + (1-H_c)(1-H_m) H_s (T_s + T_m + T_c)$$

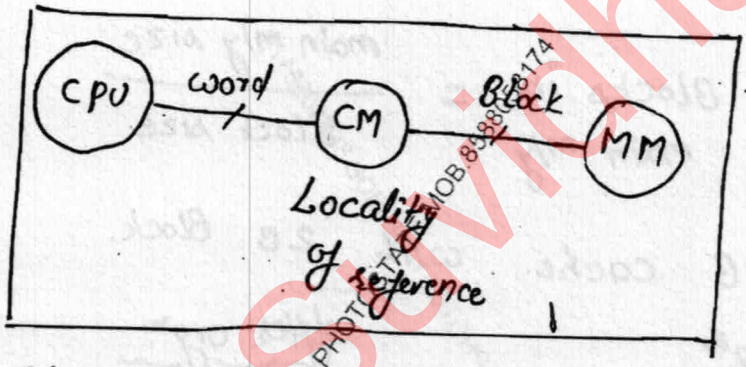
$H_s = 1$

3-level m/y structure

V. Imp

Cache Memory :-

:- Cache memory acts ap a intermediate memory between CPU and main-memory, used to hold the image of a main-memory data so CPU will be accessing the main-memory data from the cache-memory within less amount of time therefore speed-gap is synchronized between the CPU and main-memory.



Design-Elements :-

1. memory organization
2. mapping techniques
3. Replacement Algorithms
4. updating techniques
5. multilevel cache design.

memory organization :-

According to Heirarchy design, data will be transferred from main-memory to cache memory in a form of blocks so both of the memories are organized into a Blocks based on the block-size, so

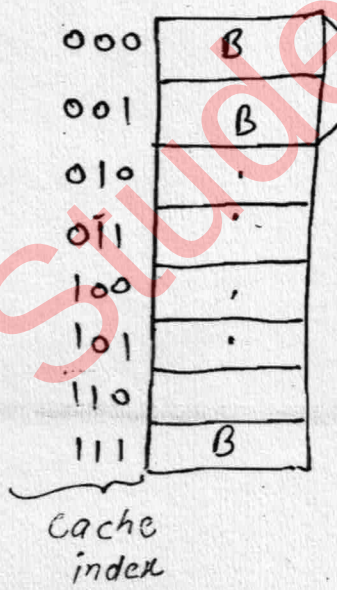
$$\#\text{ Blocks (Lines) in C.M.} = \frac{\text{C.M. Size}}{\text{Block size}}$$

$$\#\text{ Blocks in main m/y} = \frac{\text{main m/y size}}{\text{Block size}}$$

→ Conpider 8B cache with 2B Block.

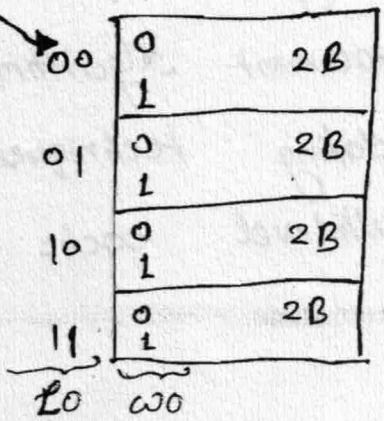
Before orgⁿ

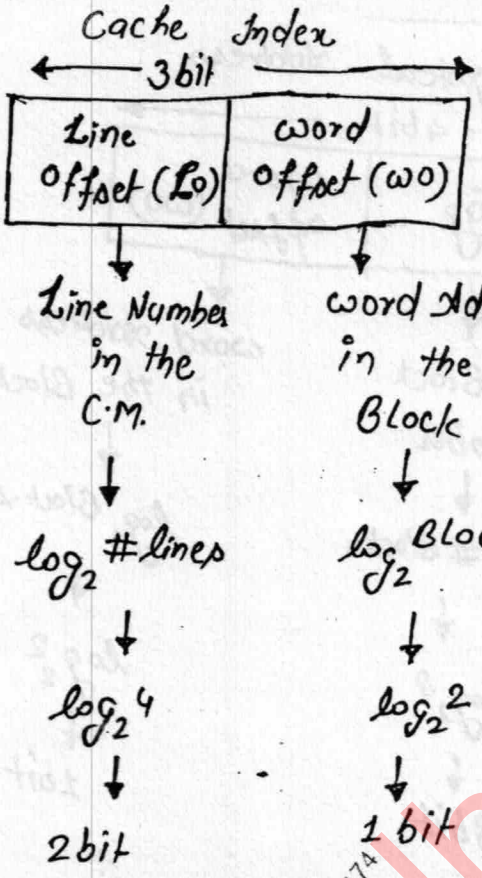
8B → C.M.



After orgⁿ

- # lines (N) = $\frac{8}{2} = 4$
- 4 * 2B = 8B

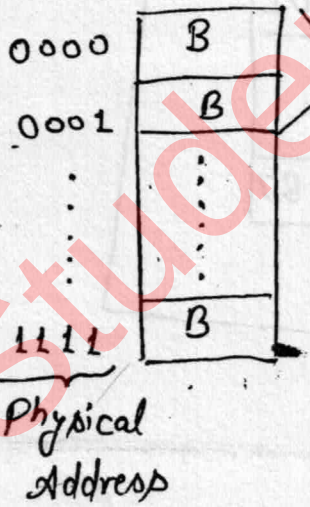




Consider 16B MM with 2B Block :-

Before org^{n} :

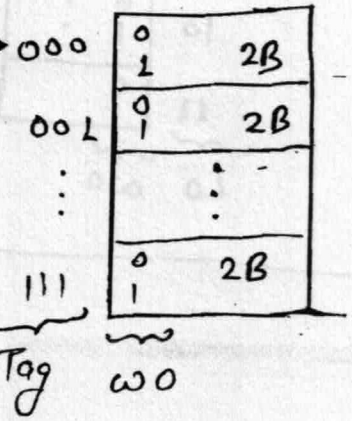
16B → MM

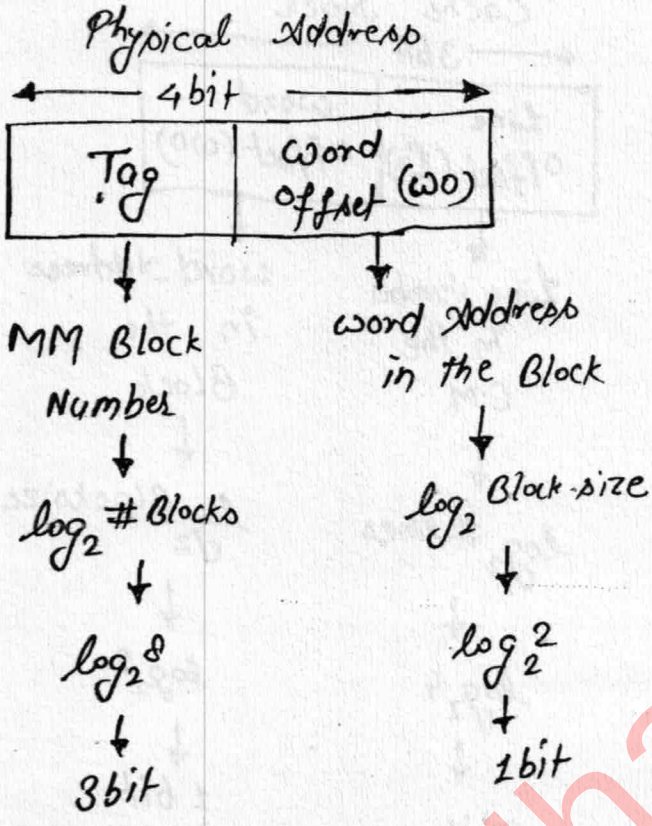


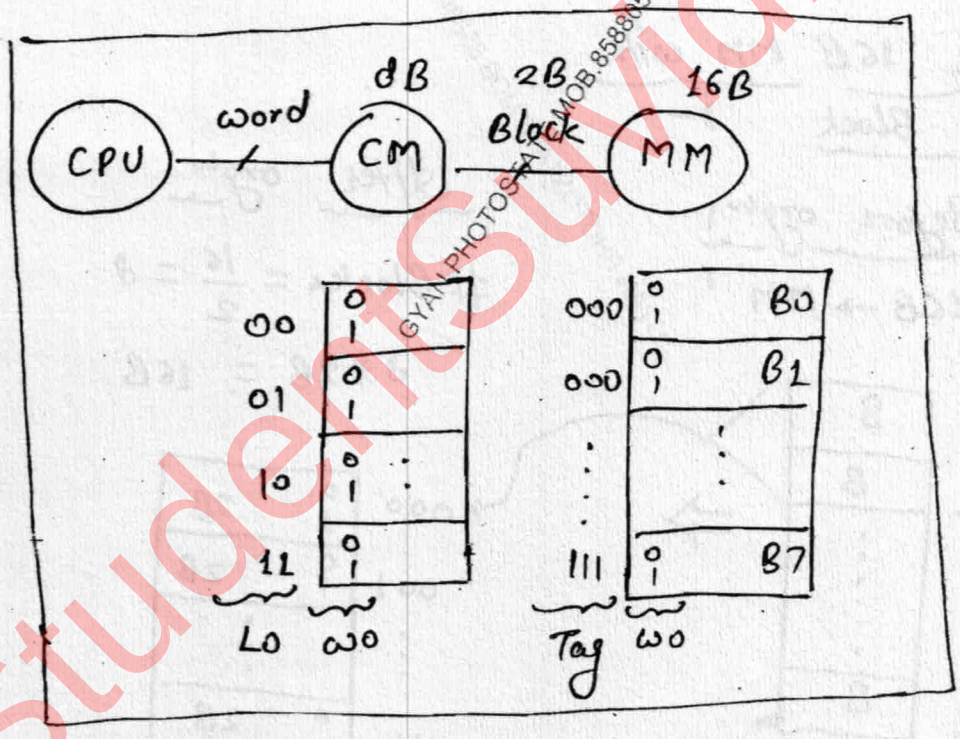
After org^{n}

Blocks = $\frac{16}{2} = 8$

8 * 2B = 16B





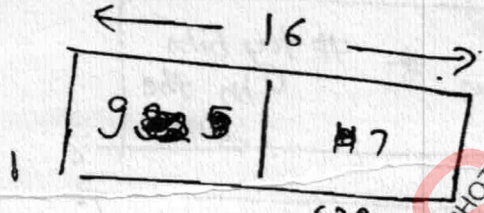


Que:

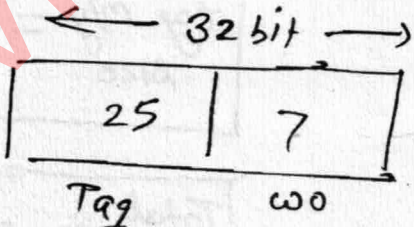
Que :- Consider a system configuration with 64kB cache organized into a 32 bi word Blocks. word size in the cpu is 32 bit. cpu generates 32 bit Physical address to access the data during the Program execution, How many lines and how many blocks are present in the respective memories. show the address format :

#. Blocks in cache = 32

Cache index



Physical address



lines = $\frac{\text{CM size}}{\text{Block size}}$

$= \frac{64kB}{32 \text{ word}}$

$= \frac{64kB}{32 \times 4B}$

$= \frac{2^{16}}{2^7}$

$= 2^9$

Blocks = $\frac{\text{MM size}}{\text{Block size}}$

$= \frac{2^{32}B}{32w}$

$= \frac{2^{32}B}{32 \times 4B}$

$= \frac{2^{32}}{2^7}$

$= 2^{25}$

Mapping Techniques :-

:-. The Process of transferring the data from main m/y to cache m/y is called as mapping.

** During the mapping Process Data Block is Copied into a cache memory along with a physical Address.

→ To Hold the Physical Address information, some extra d' Space is maintained in the cache line called as Tag-Space.

* Tag memory size in the cache controller is calculated as :

$$\text{tag m/y size} = \#\text{lines in C.M.} * \#\text{tag bits in the line}$$

$$\text{Total Cache size} = \text{Tag m/y size} + \text{Data m/y size}$$

$$\text{Data m/y size} = \{\#\text{lines in C.M.} * \text{Block size}\}$$

* In the cache Design, three kinds of mapping techniques are used to map the data name as :

1) Direct mapping
2) Associative mapping
3) Set Associative mapping