

Gatenotes.in

# **GATE**

## **Computer Science**

**Ravindrababu Ravula GATE CSE**  
**Hand Written Notes**

**-: SUBJECT :-**  
**Computer Organization**

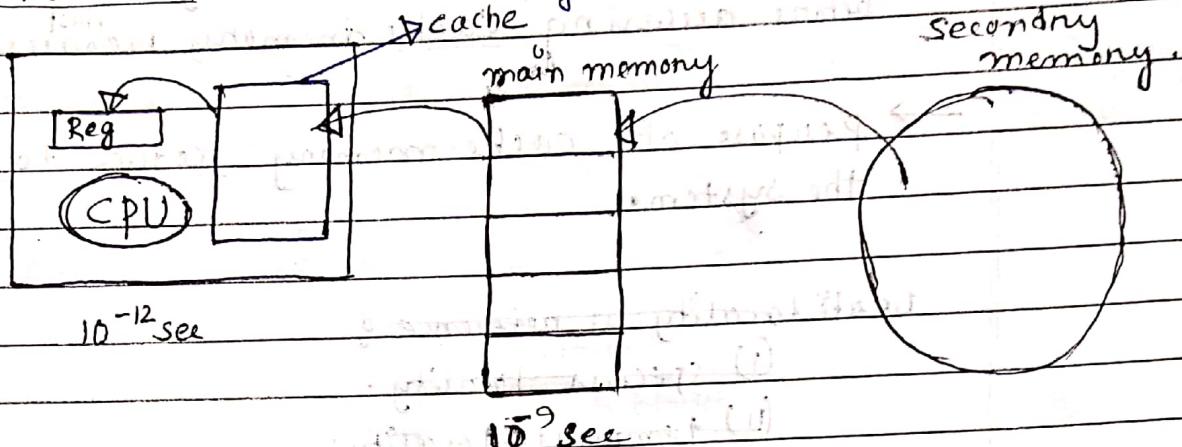
# INDEX

Name : Rakesh Nama.  
Class :

Subject : C.O.A

## Cache memory

- Introduction to cache memory:



→ Cache memory faster than main memory.

• Cache hit: If required element present in cache, then it is called 'cache hit'. That,

• hit latency: Time taken to find out whether element present on the cache or not, that is called "hit latency".

• cache miss: If required element not present in cache, that is called 'cache miss'.

• Miss latency: Time taken to get something from main memory and then place it into the cache and then read that's called "miss latency".

• page hit: If required element present in main memory,

• Page fault: If required element not present in main memory,

• Tag directory: Tag directory says that required element present in tag or not.

→ Before accessing main memory we access page table,  
 Before accessing Cache memory we access Tags.

→ purpose of cache memory reduce to cost of  
 the system.

to access locality of reference :-

(i) spatial locality :-

(ii) temporal locality :-

- Locality of reference :- is a term for the phenomenon in which the same values or related items in memory frequently access related storage locations are frequently accessed, depending on the memory access pattern.

### (Locality of reference)

#### ↓ (Temporal locality)

Recently referenced items are likely to be referenced in the near future.

#### ↓ (Spatial locality)

Items with nearby addresses tend to be referenced close together in time.

### • Introduction to Direct mapping

→ In direct mapping, the address is divided into three fields.

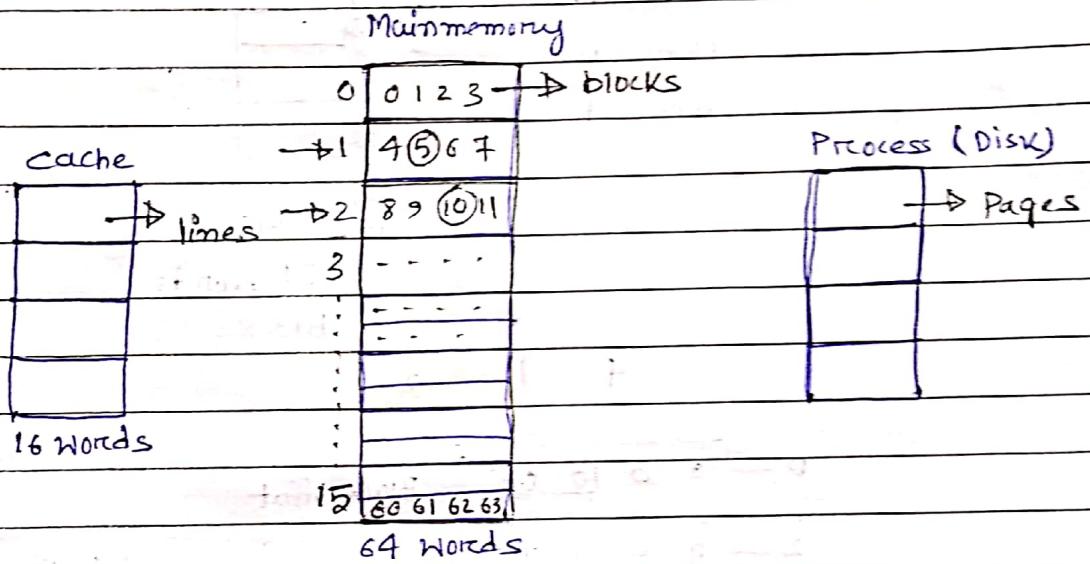
→ Address = Block Address + Offset Address  
 → Block Address + Offset Address

- Introduction to Direct mapping:

→ talking about Disk and main memory. No talking about paging.

→ talking about cache and " " " " " → BLOCKS.

→ Block size = linesize.



→ smallest addressable unit called word.

let's

$1W = 1B$  (means our system is byte addressable)

Block size = 4 words

$$\text{No. of Block in main memory} = \frac{64}{4} = 16 \text{ BLOCK.}$$

$$\text{No. of lines in cache} = \frac{16}{4} = 4 \text{ lines}$$

physical address contain = 6 bits



processor generate address = 

0	0	0	1	0	1
---	---	---	---	---	---

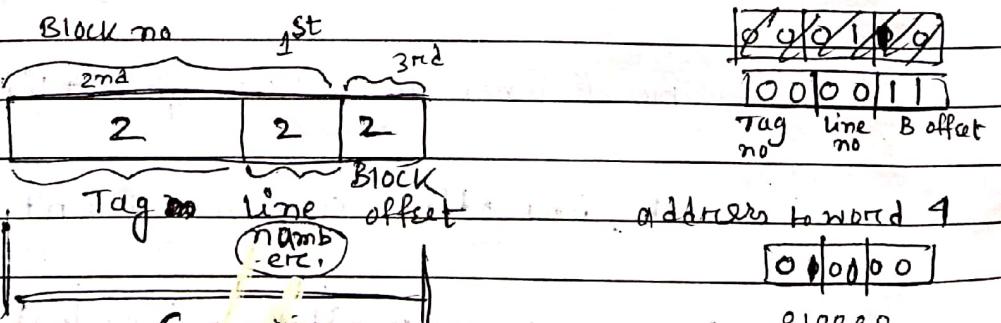
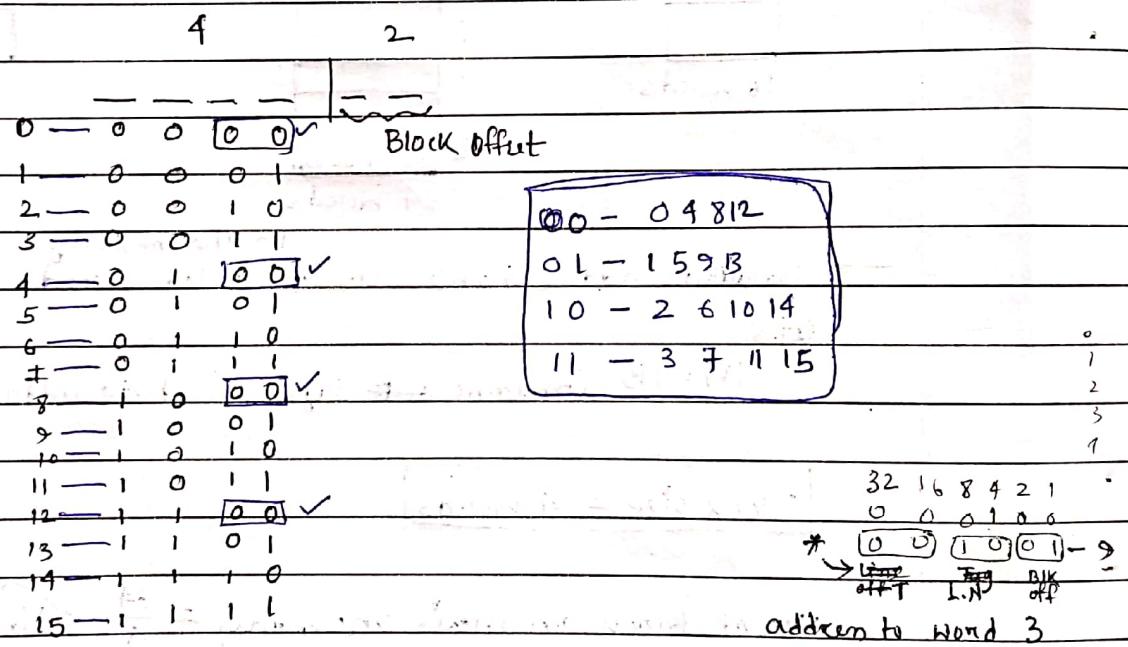
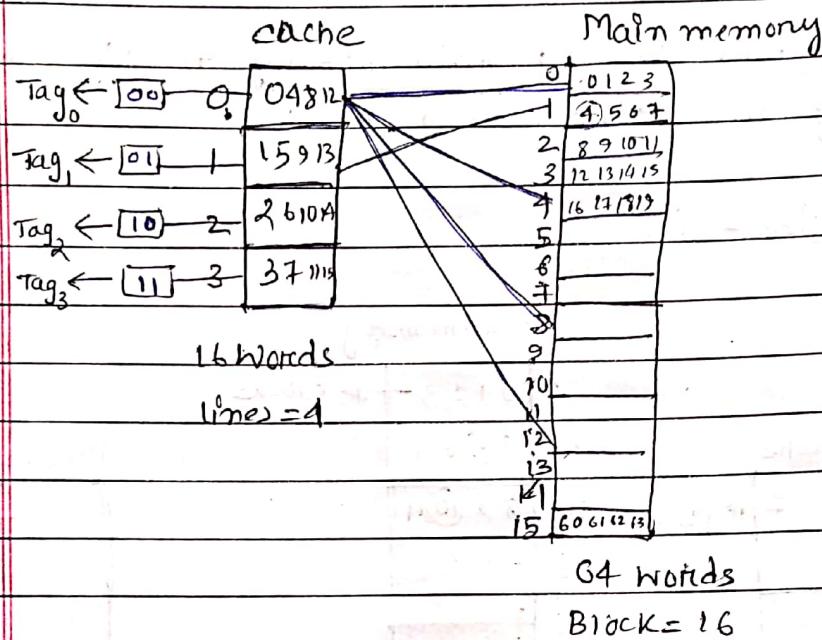
 (to word 5)  
B.I.K.      B.I.K.  
num            offset

= 

1	0	0	1	0	1
---	---	---	---	---	---

 (to word 10)  
B.no      B.off

• Direct Mapping:



1 Bit ~~and~~  
Byte = 8 bit  
 $KB = 2^{10} B$   
 $MB = 2^{10} KB$   
 $GB = 10^9 MB$

atlantis

Date \_\_\_\_\_  
Page 5

## Direct Mapping Problem - 1

MM size	Cache size	Block size	Tag bits	Tag directory size
$\checkmark \alpha_1$ $128 KB$	$16 KB$	$256 B$	$\checkmark 3$ bit	$(3 * 2^6)$ bit
$\checkmark \alpha_2$ $32 GB$	$32 KB$	$1 KB$	$\checkmark 10$ bit	$(10 * 2^5)$ "
$\checkmark \alpha_3$ $64 GB$	$512 KB$	$1 KB$	7	$(7 * 2^9)$ "
$\checkmark \alpha_4$ $16 GB$	$512 KB$	$1 KB$	10	$(10 * 2^9)$ "
$\checkmark \alpha_5$ $64 MB$	$2^{16} B$	can't guess	10	can't guess
$\checkmark \alpha_6$ $2^{26} B$	$512 KB$	can't guess	7	can't guess

Assuming that memory is Byte addressable.

~~Q1:~~

$$\text{Main memory size} = 128 KB$$

$$= 2^{10} * 128 B$$

$$= 2^{17} B$$

$$\text{Block size} = 256 B$$

$$= 2^{8} B$$

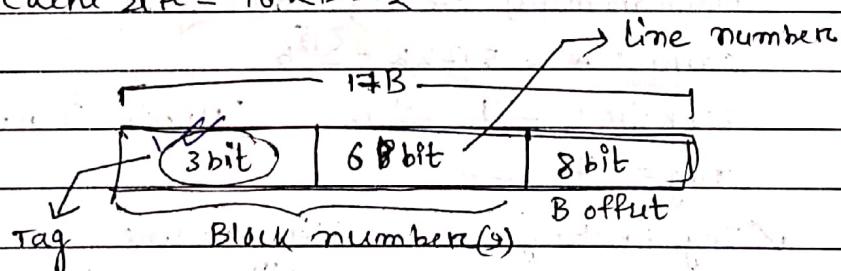
$$\text{no. of blocks} = \frac{2^{17}}{2^8} = 2^9 - \text{bit}$$

$$\text{Cache size} = 16 KB = 2^{14} B$$

$$2^{14}$$

$$2^8$$

6



$$\text{no. of lines} = \frac{2^{14}}{2^8} = 2^6 - \text{bit}$$

Line = Block size

$$\begin{aligned} \text{Tag directory size} &= (\text{Tag size} * \text{no. of lines}) \\ &= (3 * 2^6) \text{ bit} \end{aligned}$$

9

Q2:

$$\text{MM size} = 32 \text{ GB}$$

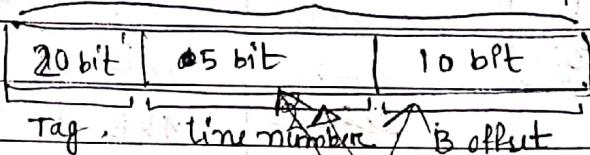
$$= 2^{30} \times 2^{10} \times 32 \text{ MB}$$

$$= 2^{10} \times 2^{10} \times 32 \text{ KB}$$

$$= 2^{10} \times 2^{10} \times 2^{10} \times 32 \text{ B}$$

$$= 2^{30+5} \text{ B} = 2^{35} \text{ B}$$

→ 35 bit



$$\text{Block size} = 1 \text{ KB}$$

$$= 2^{10} \text{ B}$$

(10 bit for Block B offset)

$$\text{Cache size} = 32 \text{ KB}$$

$$= 2^{10} \times 2^5 \text{ B}$$

$$= 2^{15} \text{ B}$$

$$\text{no. of line} = \frac{\text{C size}}{\text{line size}}$$

$$= \frac{2^{15}}{2^{10}}$$

$$= 2^5$$

$$\text{Tag directory size} = 10 \times 2^{10} \text{ Tag} \times \text{no. of lines.}$$

$$= (10 \times 2^5)$$

$$2^6 \text{ B}$$

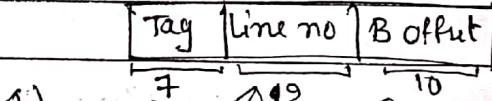
Q3:

$$\text{Main memory size} = ? \quad (64)$$

$$\text{Cache size} = 512 \text{ KB} = 2^{19} \text{ B}$$

$$\text{Block size} = 1 \text{ KB} = 2^{10} \text{ B}$$

$$\text{Tag} = 7$$



$$\checkmark \text{ Tag directory size} = 7 \times 2^{19} \text{ no. of line}$$

$$= 7 \times \frac{2^{19}}{2^{10}} = (7 \times 2^9) \text{ B bit}$$

$$\text{no. of line, } \frac{2^{19}}{2^{10}} = 2^9$$

~~no. of block & block size~~

$$\text{Address size} = 7 + 19 + 10 = 36 \text{ bits}$$

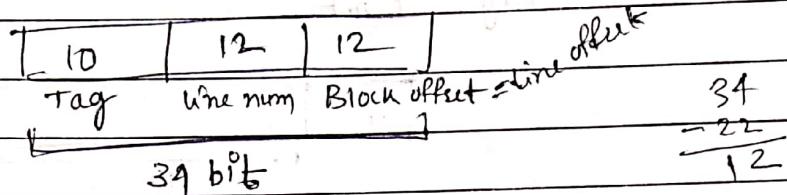
$$\text{Main memory size} = 2^{36} \text{ B}$$

$$= 2^{30} \times 2^6 \text{ B} = 2^6 \text{ MB} = 64 \text{ MB}$$

$$= 2^{30} \times 2^5 \text{ B} = 4 \text{ GB}$$

~~Q24~~

$$\text{MMU size} = 16 \text{ GB} = 2^{30} \times 2^4 \text{ B} = 2^{34} \text{ B}$$



Cache size = ?

$$\text{Block size} = 4 \text{ KB} = 2^{10} \times 2^4 \text{ B} = 2^{12} \text{ B}$$

tag = 10

tag = Directory.

$$2^{12} \times 2^{12} = 2^{24} \text{ B}$$

cache size =  ~~$2^{10} \times 2^{12}$~~  ~~2048 blocks~~ 4 KB

$$\text{no of lines} = \frac{\text{c size}}{\text{line size}} = \frac{2^{24}}{2^{12}} = 2^{12}$$

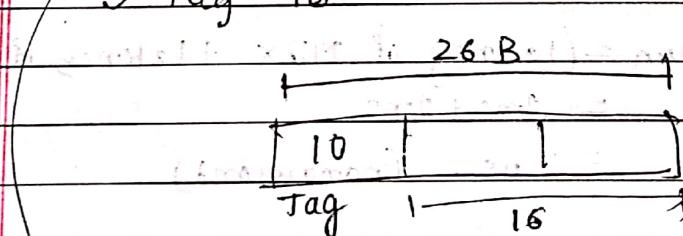
Tag directory = ~~10~~  $10 \times 2^{12}$ = 10 bits.  $(10 \times 2^{12})$  bit.~~Q25~~

$$\text{MMU size} = 64 \text{ MB} = 2^{20} \times 2^6 \text{ B} = 2^{26} \text{ B (PA)}$$

cache size = ?

BLOCK size = ?

tag = 10

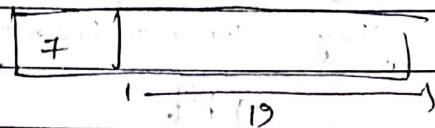
cache size =  $2^{16} \text{ B}$ 

Comments the width of the address is 26 bits  
(given block size) specified

~~B6~~

$$\text{Cache size} = 512 \text{ KB} = 2^{19} \text{ B}$$

Tag = 7 bit



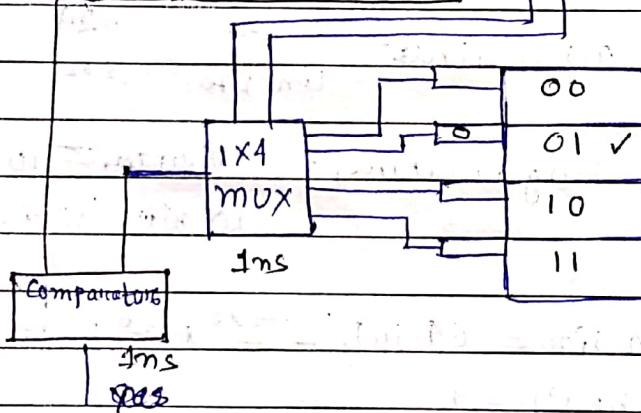
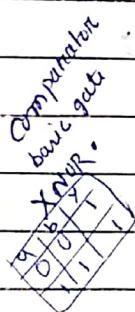
$$19 + 7 = 26$$

$$\checkmark \text{ MM size} = 2^{26} \text{ B}$$

### • Direct Mapping HW Implementation =

CPU generated address  $\rightarrow$ 

Tag	LN	BD
0	01	



1 - hit

0 - miss

$$\begin{aligned} \text{Total time taken} &= (\text{latency of MUX} + \text{latency of comparators}) \\ &= 1\text{ms} + 1\text{ms} \\ &= 2\text{ms. (nanoseconds)} \end{aligned}$$

$\rightarrow$  no of MUX and comparators depends on ~~no. of~~ no. of Tag bits.

Direct  
mapping  $\rightarrow$

$$\text{Tag size} = K \text{ bit}$$

$$\text{no. of MUX required} = K$$

$$\text{no. of Comparators} = 1 * (K \text{ bit compare})$$

always (in Direct map)

Q1:

MM size = 1 GB

(hence latency MUXA negligible)

Cache size = 1 MB

Comparator size = 10 Kms.

1-bit latency = ?



Hit latency



MM size 1 GB

\*\* Tag = Main M size  
Cache size

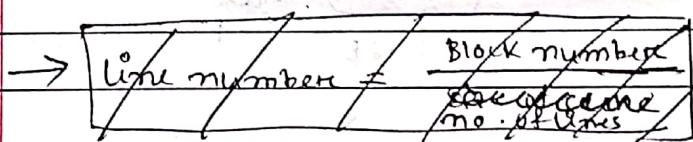
$$= \frac{1 \text{ MB}}{1 \text{ MB}} \Rightarrow 2^{10} \text{ KB } 1 \text{ KB} \Rightarrow 2^{10} \text{ B}$$

$$K = 10$$

Hit latency = ~~10 \* K~~ 10 Kns

$$= 10 * 10 \text{ ns}$$

$$= 100 \text{ ns}$$



line number = (B number) %  
(no. of lines)

$$k = m \% n$$

• Disadvantage of direct mapping =

→ Conflict miss problem.

ex:-

Block req by CPU -

0	4	8, 12, 5, 6, 9, 1, 15, 20
1	5	9
2	6	
3	15	→ page was empty for long time, but Line no. 0 used heavily, this no. of lines = 4 is conflict miss problem.

$$L-N = (B.N) \% (N.L)$$

Ex:

Block request by CPU

0	4	8	12	16	20
1		7			
2			3		
3				5	

cache.

$$K = m/n$$

~~n = num of lines in cache~~

$$4 \leq 20 \quad 4 \leq 4 = 0$$

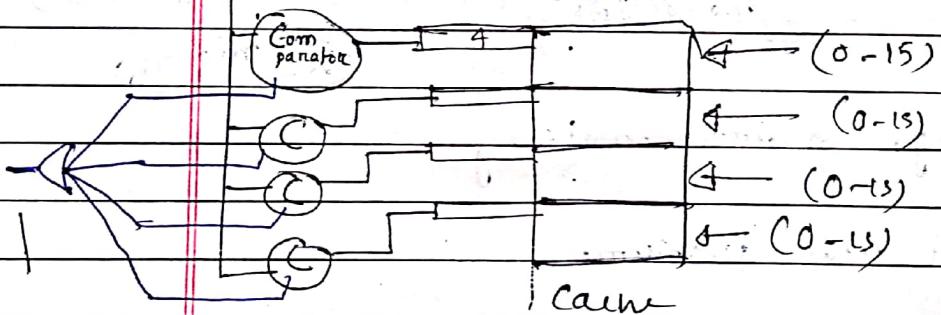
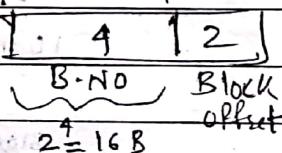
$$8 \leq 20 \quad 8 \leq 4 = 0$$

Line no. 1, 2, 3, empty but they line 0 are heavily used  
it is conflict miss problem.

### Associative mapping = (intro)

can solve

→ By Associative mapping A Conflict miss problem.



1 - Cache hit

0 - Cache miss

→ 4 lines - 4 comparators need, so hardware cost increase along with freedom.

→ No of Comparator required = No of cache lines,

→ We get freedom to place Block anywhere in cache, but requirement of Comparator increase.

(Q1)

Main memory size = 32 GB

Block size = 32 KB

✓ Tag(K) = ?

Propagation delay (PD) of comp = 10 ns

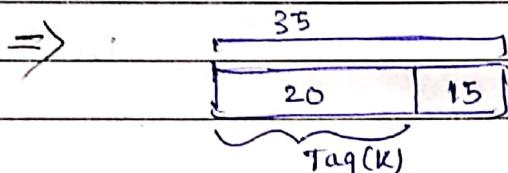
PD of OR Gate = 10 ns.

✓ Hit latency = ?

$$MM = 32 \text{ GB}$$

$$= 2^{30+5} \text{ B}$$

$$= 2^{35} \text{ B}$$



$$BS = 32 \text{ KB}$$

$$= 2^{15} \text{ KB}$$

$$\checkmark K = 20$$

$$\begin{aligned} \checkmark \text{hit latency} &= \text{PD of comparator} + \text{PD of OR gate} \\ &= 10 \text{ ns} + 10 \text{ ns} \\ &= (10 + 20) \text{ ns} + 10 \text{ ns} \\ &= 210 \text{ ns.} \end{aligned}$$

(Q2)

	MM size	cache size	Block size	Tag size	Tag directory size
① -	128 KB	16 KB	256 B	9 bits	( $2^9 \times 2^6$ ) bits
② -	32 GB	32 KB	1 KB	25 bits	( $2^5 \times 32$ ) bits
③ -	128 MB	512 KB	1 KB	17	( $17 \times 2^9$ ) bits
④ -	16 GB	?	4 KB	22 bits	(can't guess)
⑤ -	64 MB	?	64 KB	10	(can't guess)
⑥ -	not possi	512 KB	not possi	7	not possible.

$$\textcircled{1} \quad \text{MM size} = 128 \text{ KB}$$

$$\text{CS} = \text{Cache size} = 16 \text{ KB}$$

$$\text{no of lines} = \frac{\text{CS}}{\text{BS}}$$

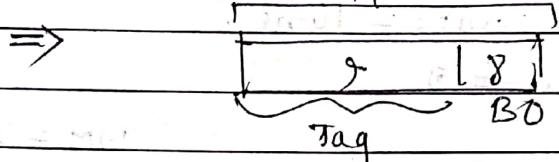
$$\text{BLOCK size (BS)} = 256 \text{ B}$$

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

$$\text{Tag size} = ?$$

$$= \textcircled{6}$$

$$\text{Tag directory size} = ?$$



$$\text{BB} = 128$$

$$= 2^7 \times 2^{10} \text{ B}$$

$$= 2^{17} \text{ B}$$

$$\text{BS} = 256 \text{ B} = 2^8 \text{ B}$$

$$\checkmark \text{ Tag} = 9$$

$$\checkmark \text{ Tag directory size} = \text{Tag size} * \text{no of lines.}$$

$$= (9 * 2^6) \text{ bits}$$

$$\text{no of com neg} = 2^6 = 64$$

\textcircled{2}

$$\text{MM} = 2^{35} \text{ B}$$

$$\text{CS} = 2^{15} \text{ B}$$

$$\text{BS} = 2^{10} \text{ B}$$

$$\text{no. of Line} = \frac{2^{35}}{2^{10}}$$

$$= 2^5$$

$$\checkmark \text{ tag directory} = \text{tag} * \text{no of lines}$$

$$= 25 * 2^5$$

$$\text{MM} = ?$$

$$\text{CS} = 2^{19} \text{ B}$$

$$\text{BS} = 2^{10} \text{ B}$$

$$\text{TS} = 17$$

$$\text{tag directory} = ?$$

$$\checkmark \text{ MM size} = 2^{10+4}$$

$$= 2^{27} \text{ B} = 8 \text{ } 2^{27} \text{ MB}$$

$$= 128 \text{ MB}$$

$$\text{no. of lines} = \frac{2^{35}}{2^{10}} = 2^9$$

$$\leftarrow \text{Tag D.S} = (17 * 2^9) \text{ bits.}$$

$$(4) MM = 2^{31} B$$

$$CS \Rightarrow CS = ? X$$

$$BS = 2^{12} B$$

$$\sqrt{TS} = ?$$

$$T \cdot DS = ? X$$

$$\begin{aligned} \text{Tag size} &= (34 - 12) \\ &= (22) \end{aligned}$$

$$(5) MM = 2^{26} B$$

$$CS = ?$$

$$BS = ? (2^{16})$$

$$\text{Tag} = 10$$

$$\text{Tag D-S} = ?$$

$$BS = 2^{26-10}$$

$$= 2^{16} = 64 KB$$

$$(6) MM = ? X$$

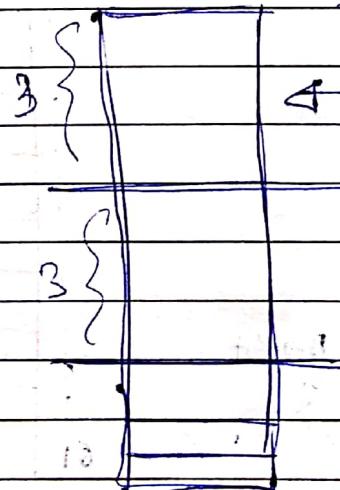
$$CS = 2^{19} B$$

$$TS = ?$$

$$BS = ? X$$

$T \cdot DS = ? X$  not possible.

3-way set



cache

memory

- set Associative mapping = (adv. no. of comparators reduced)

Ex: (how set associative work)

MM size = 64 B

Cache S = 32 B

Block S = 4 B

\* K-way set associative  
required K-comparators

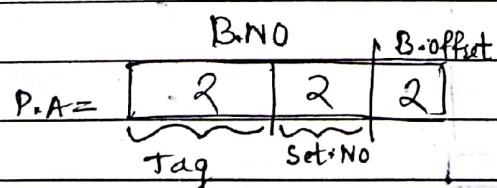
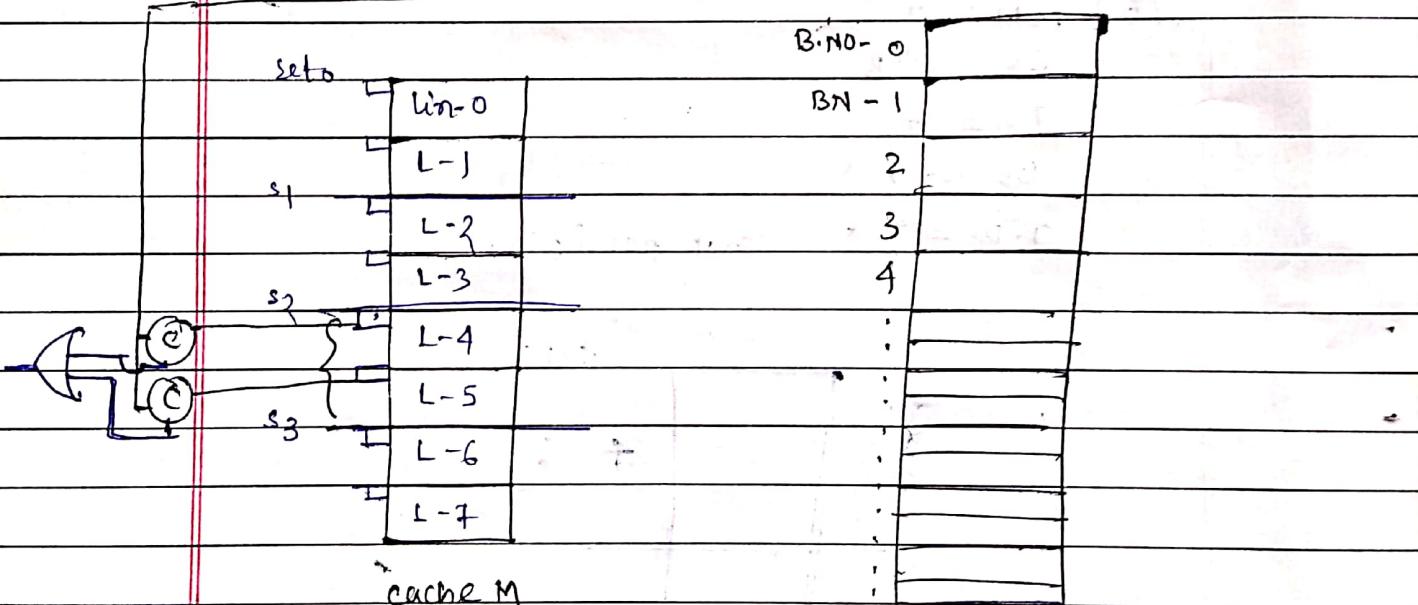
✓ set size = 2 blocks (lines)

✓ (2-way set associative)  
or

$$\text{lines} = \frac{CS}{BS} = \frac{32}{4} = 8 \text{ lines}$$

$$\text{sets} = \frac{\text{Lines}}{\text{sets}} = \frac{8}{2} = 4 \text{ sets}$$

give address (P-A) = [01 | 10 | 11]  
SNO



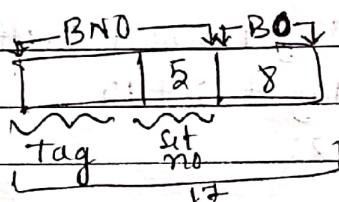
61  
62  
BN-63

Main M

(Q1)

	MM size	cache size	Block size	Tag bits	Tag directory size	set associative
①	128 KB	16 KB	256 KB	4	(4 * 2 <sup>6</sup> ) bits	2-way set
②	32 GB	32 KB	1 KB	22	(22 * 2 <sup>5</sup> ) bits	4
③	2 <sup>3</sup> MB	512 KB	1 KB	7	(7 * 2 <sup>9</sup> ) bits	8
④	16 GB	64 MB	4 KB	10	(10 * 2 <sup>14</sup> ) bit	4
⑤	64 MB	256 KB	X	10	X	4
⑥	88 MB	512 KB	X	7	X	8

(Q1)



$$MM\ size = 2^{17} B$$

$$CS = 2^{19} B$$

$$BS = 2^{10} B$$

$$\text{no. of lines} = \frac{CS}{BS} = \frac{2^{19}}{2^{10}} = 2^9 \text{ lines.}$$

$$\text{set no. of sets} = \frac{\text{no. of lines}}{\text{set size}}$$

$$= \frac{2^9}{2} = 2^8 = 256 = (32 \text{ sets})$$

$$PA = \text{Tag} + \text{set no} + \text{B.offset}$$

$$\Rightarrow 17 = \text{Tag} + 5 + 8$$

$$\Rightarrow \text{Tag} = 9$$

$$\begin{aligned} \text{Tag directory} &= \text{Tagsize} * \text{no. of lines} \\ &= (4 * 2^6) \text{ bits.} \end{aligned}$$

(2)

$$MM = 32GB = 2^{35} B$$

$$CS = 32KB = 2^{15} B$$

$$BS = 1KB = 2^{10} B$$

$$T = ? \quad (22)$$

$$T.D = ?$$

Set-arr = 4-way

$$\text{no. of set} = \frac{\text{no. of line}}{\text{set size}} = \frac{2^5}{2^2} = 2^3$$

$$\rightarrow P.A = Tag + S.no + B.off$$

$$\rightarrow 35 = Tag + 3 + 10$$

$$\checkmark \rightarrow Tag = 22$$

$$\checkmark T.D = (22 * 2^5)$$

(3)

$$MM = ?$$

$$CS = 512KB = 2^{19} B$$

$$BS = 1KB = 2^{10} B$$

$$T = 7$$

$$T.D = ?$$

Set-arr = 8-way

$\frac{2^3}{2^3}$

Tag	6	10
-----	---	----

= 20 bits & 23 bits

& (MRW) =

$$\text{no. of line} = \frac{2^{19}}{2^{10}} = 2^9$$

$$\checkmark MM \cdot \text{size} = 2^{20} = 2^3 MB$$

$$\text{no. of sets} = \frac{\text{no. of line}}{\text{size of sets}}$$

$$\checkmark T.D = T * \text{no. of lines}$$

$$= \frac{2^7}{2^3} = 2^6$$

$$= (7 * 2^9) \text{ bits.}$$

④  $MM = 16 \text{ GB} = 2^{34} \text{ B}$

$CS = ?$

$BS = 4 \text{ KB} = 2^{12} \text{ B}$

$T = 10$

$TD = ?$

set. am = 4-way set.

↓ 34 ↓

10	12	12
----	----	----

Tag set no B-offset

no. of sets =  $2^{12}$

no. of sets =  $\frac{\text{no. of lines}}{\text{set size}}$

no. of lines =  $\text{line size} \times \text{no. of sets}$

$TD = T * \text{no. of lines}$

$= 10 * 2^{12}$

$= 160 \text{ KB}$

$CS = \text{no. of lines} * \text{line size}$

$= 2^{12} * 2^{12}$

$= 2^{26} \text{ B}$

$= 2^6 \text{ MB}$

$= 2^{26} \text{ (64 MB)}$

$= 4 * 2^{12}$

$= 2^{12}$

⑤  $MM = 64 \text{ MB} = 2^{26} \text{ B}$

$CS = ?$  (2<sup>18</sup> B)

$BS = ?$

$Tag = 10$

$Tag.D = ?$

set. am = 4 way

↓ 26 ↓

10	?	?
----	---	---

Tag set no B-offset

↓ 26 ↓

10	X	Y
----	---	---

16

⑥ MM/E4

CS/E4 (512 KB)

BS/E4

TC/E4

TO/E4

set. am 8 way.

~~Cache~~

Tag

Cache size = no. sets \* cache associativity

\* line size (BS)

$= 2^x * 2^y * 2^z$

$= 4 * 2^{x+y}$

$= 4 * 2^{16}$

$= 2^{18} = 2^8 \text{ KB} = 256 \text{ KB}$

(Q2)

MM size	Cache size	Block size	Tag size	Tag directory	set associative
64MB	-	-	10	-	4-way
-	512KB	-	7	-	8-way

$$\textcircled{6} \quad MM = ? - (2^{23} \text{ B})$$

$$CS = 512\text{ KB} = 2^{19} \text{ B}$$

$$BS = -$$

$$TS = 7$$

$$TDS = -$$

$$\text{Assoc} = 8\text{-way}$$

Tag	Set	Block
-----	-----	-------

7	x	y
---	---	---

cache size = no. of sets \* lines per set

\* BLOCK size

$$2^{19} = 2^x * 2^3 * 2^y$$

$$\Rightarrow 2^{16} = 2^{x+y}$$

$$\Rightarrow x+y = 16$$

$$P.A = 7 + (x+y)$$

$$= 7 + 16 = \textcircled{23}$$

$$MM = 2^{23} = 8 \text{ MB}$$

- Comparing all the mappings

$$\text{MM size} = 4\text{GB} = 2^{32} \text{B}$$

$$\text{Cache} = 4\text{MB} = 2^{22} \text{B}$$

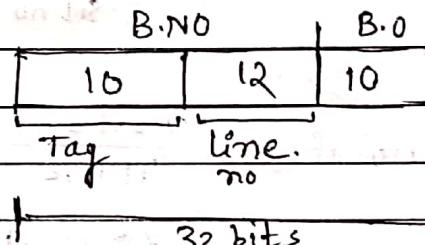
$$\text{BLOCK size} = 1\text{KB} = 2^{10} \text{B}$$

$$P.A = 32 \text{ Bits}$$

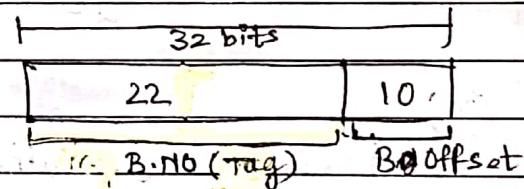
$$\text{no. of lines} = \frac{CS}{BS} = \frac{2^{22}}{2^{10}} = 2^{12} \text{ lines}$$

$$\text{no. of lines} = \frac{\text{no. of bits}}{\text{size of UF}} = \frac{2^{12}}{2^3} = 2^{10}$$

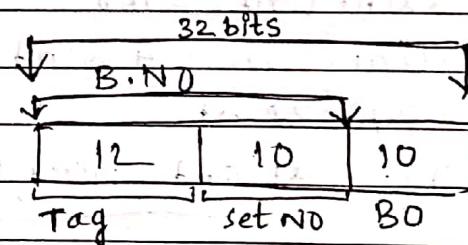
Direct mapping :-



Associative Mapping :-



4-Way Set Assoc :-



Relationship between

	Tag	Comparator registers	size of comparators (Tag directory)	Lines	size of comparator
Direct	10	1	$10 \times 2^{12}$	$2^{12}$	10
Associative	22	$2^{12}$	$22 \times 2^{12}$	$2^{12}$	22
4-Way	12	4	$12 \times 2^{12}$	$2^{12}$	12

✓ Questions - gate

Q-1995

(Q1)

Cache size = 1 K Words =  $2^{12}$  wordsBlock size = 64 words. =  $2^6$  words

Set size = 4 blocks.

The number of bits in "set" and "word" field of MM address are -

→

	1	4	6
--	---	---	---

Set no B.O or (word)

$$\text{no. of sets} = \frac{\text{no. of lines}}{\text{Set size}}$$

$$\text{no. of lines} = \frac{2^{12}}{2^6} = 2^6$$

$$= \frac{2^5}{2^2} = 2^4 \text{ (sets)}$$

(Q2)

4-Way set associative.

cache lines = 128

Lines size = 64 words

P.A. = 20 bits

Tag, set and word fields are ?

→

9	5	6
---	---	---

Tag || set no B.O or (word)

20

$$\text{no. of lines} = \frac{\text{Cache size}}{\text{Lines size}}$$

Cache size =  $2^6 * 2^7 = 2^{13}$  words.

$$\text{sets} = \frac{\text{lines}}{\text{Set size}} \Rightarrow \frac{2^7}{2^2} \Rightarrow 2^5$$

Tag = 9, set = 5, words (B.O) = 6.

(Q3) Blocks in cache = 128

4-way set associative

MM contains  $2^{14}$  blocks.

Block size is 256 eight bitwords.

(i) How many bits are required for addressing MM? (2)

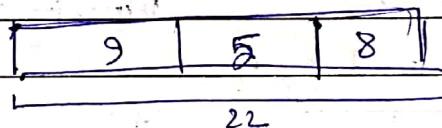
(ii) How many bits are needed to represent the tag, set and word fields. (9) (3) (8)

$$\rightarrow \text{cachelines} = 2^7$$

$$\text{set size} = 4$$

$$\text{MM Blocks} = 2^{14}$$

$$\text{Block size} = 2^8 \text{ bytes/word.}$$



$$\text{sets} = \frac{2^7}{2^2} = 2^5$$

no. of lines

$$\text{MM size} = 2^{14} \times 2^8 \text{ words}$$

$$= 2^{22} \text{ words}$$

(Q4)

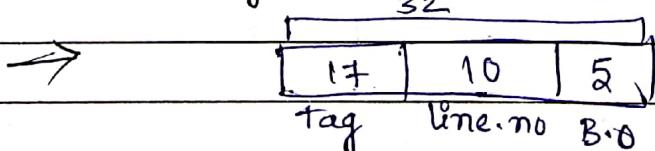
Direct mapped cache

$$\text{cache size} = 32 \text{ KB} = 2^{15} \text{ B}$$

$$\text{Block size} = 32 \text{ B} = 2^5 \text{ B}$$

$$\text{PA} = 32 \text{ bits}$$

The number of bits needed for cache indexing and tag bits are respectively.



$$\text{no. of lines} = 2^{15} = 2^{10}$$

$$2^5$$

Cache indexing = 10 bit

Tag bit = 17. Ans - (10, 17)

9-2006 05

1-20-1-21

Consider these & make connections

① first one

② 2<sup>nd</sup> one

## Sete associações

Opferkult

$$\text{Cache size} = 32 \text{ KB} = 2^{15} \text{ B}$$

### Career life stages

## 2-Way set association

## Direct mappe

$$\text{Block size} = 32B = 2^5B$$

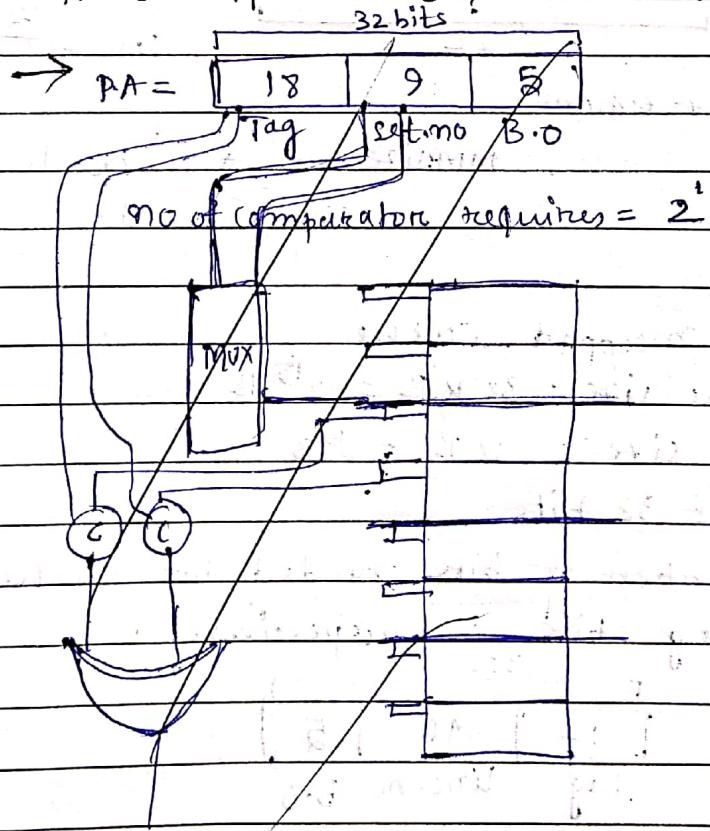
$$P.A = 32 \text{ bits}$$

2-to-1 MUX latency = 0.6 ns.

K bit comparators has latency =  $K/10$  ns.

The hit latency of set associative organization is  $\lceil \log_2 n \rceil$  and " " Direct mapped is  $\lceil \log_2 n \rceil$ .

find  $h_1$  and  $h_2$ ?



$$\text{line no} = \frac{2^{15}}{25} = 2^{10}$$

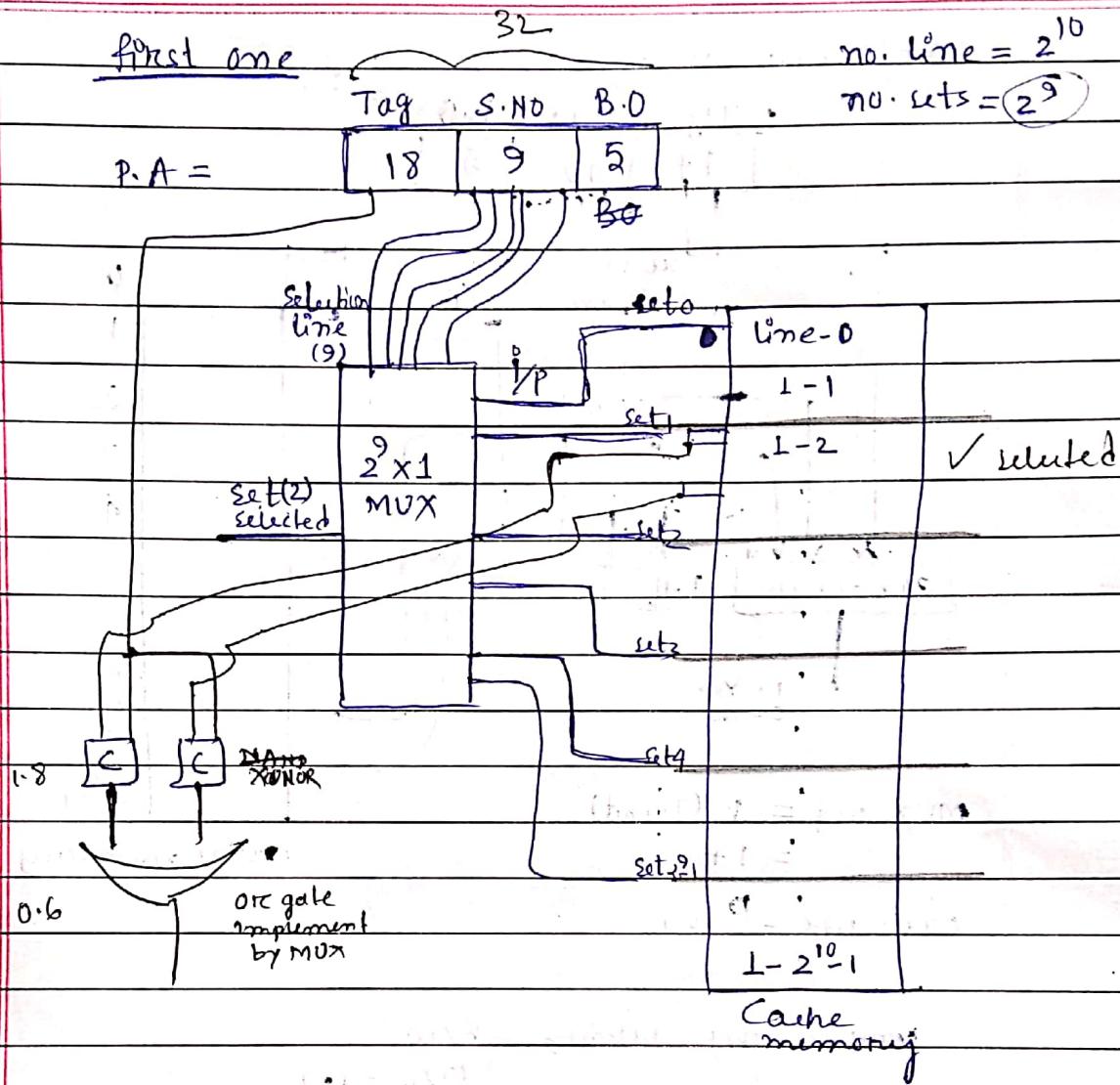
$$\text{no. of set} = \frac{2^{10}}{2^1} = 2^9$$

~~here, MUX required~~

$\approx 2 \times$  Tag bits

$$= K \times \text{Tag bits}$$

hit latency =



$$\text{MUX req} = \text{Tag bit} \times 2$$

K-way set associative  $\Rightarrow (\text{MUX req}) = T \times K$

size of each MUX =  $2^S \times 1$

Comparators need = K

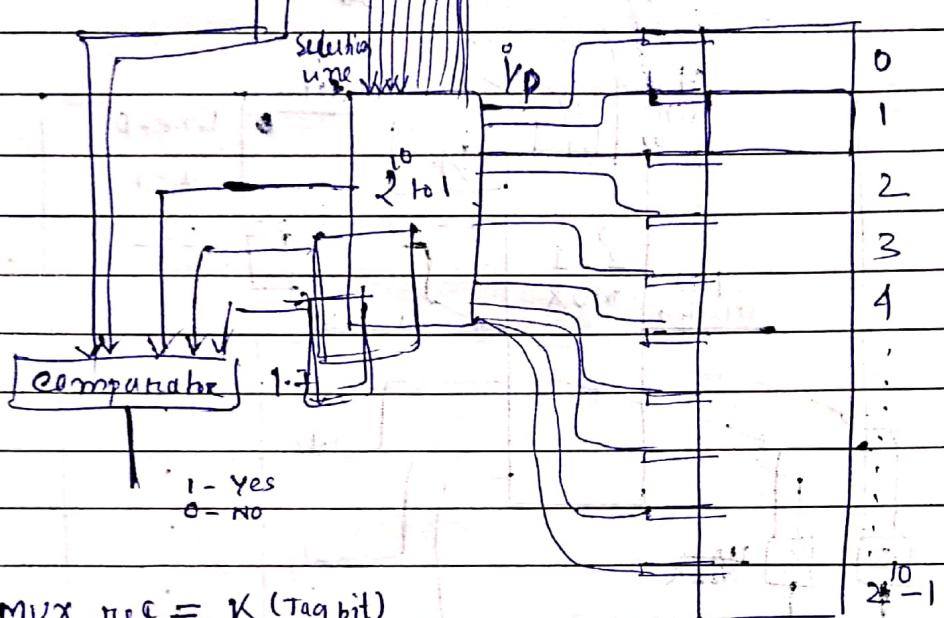
$$\text{comparator latency} = K/10 = 18/10 = 1.8 \text{ ns.}$$

$$t_1 = 1.8 + 0.6 \\ = 2.4 \text{ ns}$$

~~Second one -~~

Tag L.H.O B.O - 32 bit

17	10	5
----	----	---



$$\text{MUX ref} = K \text{ (Tag bit)}$$

$$= 17$$

cache memory

$$\text{each size} = 2^{10} \text{ to } 1$$

$$\text{computation latency} = K/10$$

$$= 17/10 = 1.7$$

$$\text{Total time} = 1.7 \text{ ms}$$

g-2011

(Q6) Direct mapping

cache size = 8KB

BS = 32 byte.

PA = 32 bits.

The cache controller maintains tag information for each cache block comprising of following.

1 valid bit, 1 modified bit and address many bits as the minimum needed to identify block mapped in the memory block mapped in the cache.

What is the total size of memory needed at the cache controller to store meta-data (tags) for the cache?

$$\rightarrow \text{Lines} = \frac{CS}{BS} = \frac{2^{13}}{2^3} = 2^8$$

19	8	5
$(19+1+1) = 21$		

$$\begin{aligned} T\text{-Directory} &= (21 * 2^8) \text{ bits} \\ &= 21 * 256 \\ &= 5376 \text{ bits.} \end{aligned}$$

P-2<sup>29</sup>, A-1  
1-40, (Q7)  
g-2012

$$\text{cache size} = 256 \text{ KB} = 2^{18} \text{ B}$$

$$\text{set size} = 4$$

$$B.S = 32 \text{ B} = 2^5 \text{ B}$$

$$P.A = 32 \text{ bits}$$

16	11	5
Tag	Set	BO

$$\begin{aligned} \text{no. of lines} &= \frac{BS \times CS}{BS} \\ &= 2^{13} \end{aligned}$$

① The number of bits in the tag field of an address - (16).

$$\begin{aligned} \text{sets} &= \frac{2^8 \text{ lines}}{2^2} \\ &= \frac{2^{13}}{2^2} = 2^{11} \end{aligned}$$

② The size of the cache tag directory is - Lines \* tag

$$\begin{aligned} 2^{13} * (16 + 2 + 11) &= 2^{13} * 16 \\ &= 2^{13} * 20 \\ &= 2^3 * 20 \text{ KB} \\ &= 160 \text{ KB.} \end{aligned} \quad \begin{aligned} &= 2^{13} * 16 \\ &= 2^{13} * 4 \\ &= 2^{17} \\ &= 2^7 \text{ KB} \\ &= 128 \text{ KB} \end{aligned}$$

g-2019  
(88)

4-way set associativity

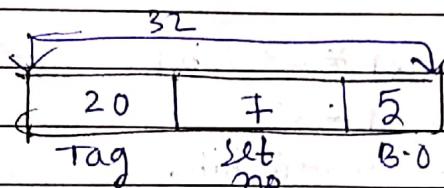
$$\text{Cache size} = 16 \text{ KB} = 2^{14} \text{ B}$$

$$\text{Block size} = 8 \text{ words} = (8 \times 32) \text{ bits} = \frac{2^8}{2^3} \cdot B_0 = 2^5 B$$

$$\text{Word size} = 32 \text{ bits}$$

$$PAS = 4 \text{ sets} = 2^2 \text{ B}$$

$$\text{tag bits} = ?$$

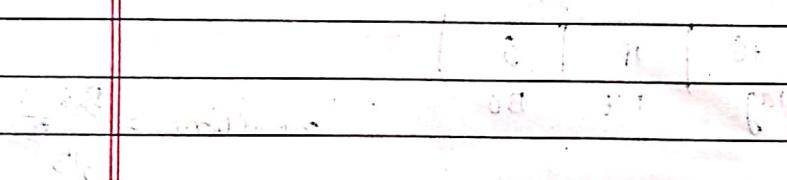


$$\text{Lines} = \frac{C_S}{B_S}$$

$$\text{Tag bits} = 20$$

$$\text{set-lines} = \frac{2^7}{2^2} = 2^5 = 32$$

$$= 2^7$$



Cache organization =  $\frac{\text{Cache size}}{\text{Block size}} \times \frac{\text{Block size}}{\text{Word size}}$

$= \frac{2^{14}}{2^5} \times \frac{2^5}{2^3} = 32 \times 8 = 256$

Cache organization =  $\frac{\text{Cache size}}{\text{Block size}} \times \frac{\text{Block size}}{\text{Word size}}$

$= \frac{2^{14}}{2^5} \times \frac{2^5}{2^3} = 32 \times 8 = 256$

Cache organization =  $\frac{\text{Cache size}}{\text{Block size}} \times \frac{\text{Block size}}{\text{Word size}}$

$= \frac{2^{14}}{2^5} \times \frac{2^5}{2^3} = 32 \times 8 = 256$

Cache organization =  $\frac{\text{Cache size}}{\text{Block size}} \times \frac{\text{Block size}}{\text{Word size}}$

$= \frac{2^{14}}{2^5} \times \frac{2^5}{2^3} = 32 \times 8 = 256$

Cache organization =  $\frac{\text{Cache size}}{\text{Block size}} \times \frac{\text{Block size}}{\text{Word size}}$

$= \frac{2^{14}}{2^5} \times \frac{2^5}{2^3} = 32 \times 8 = 256$

atlantis

Date \_\_\_\_\_

Page \_\_\_\_\_

27

atlantis

Date \_\_\_\_\_  
Page 28

## Memory Interfacing

www.gatenotes.in

atlantis

Date \_\_\_\_\_  
Page 29

### Computer Architecture:

It deals with instructions, addressing modes, ALU, pipelining etc (Internal design)

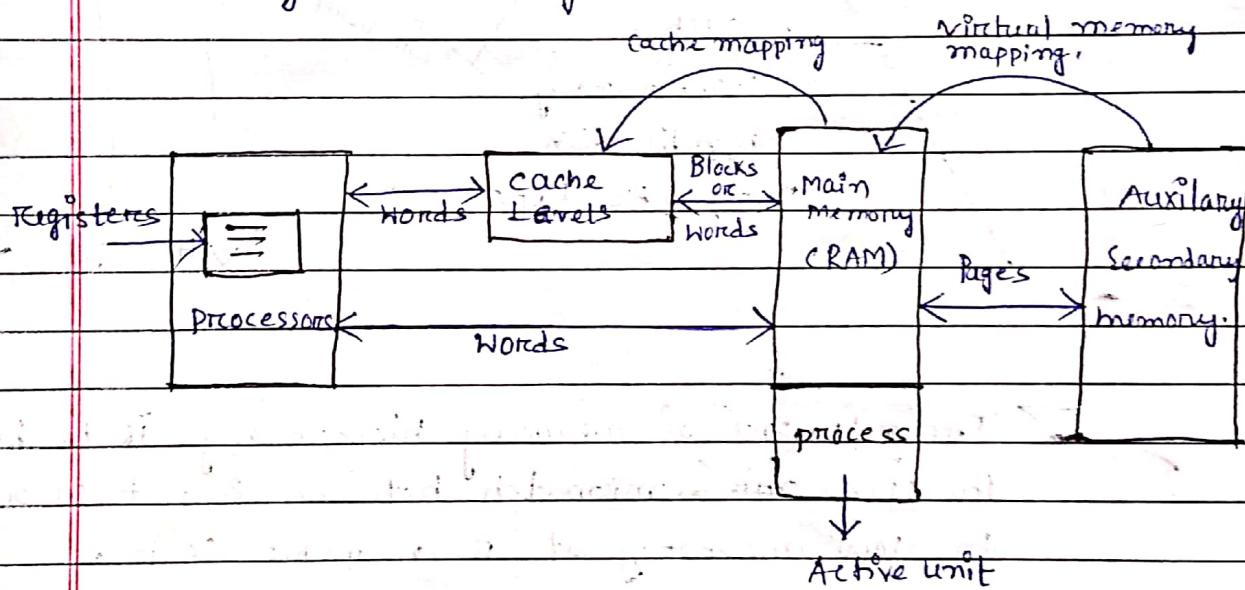
### Computer organization:

It deals with how various memory and I/O interact with a system.

### Computer design:

It deals with hardware design.

### Memory Interfacing -



- Memory Hierarchy :

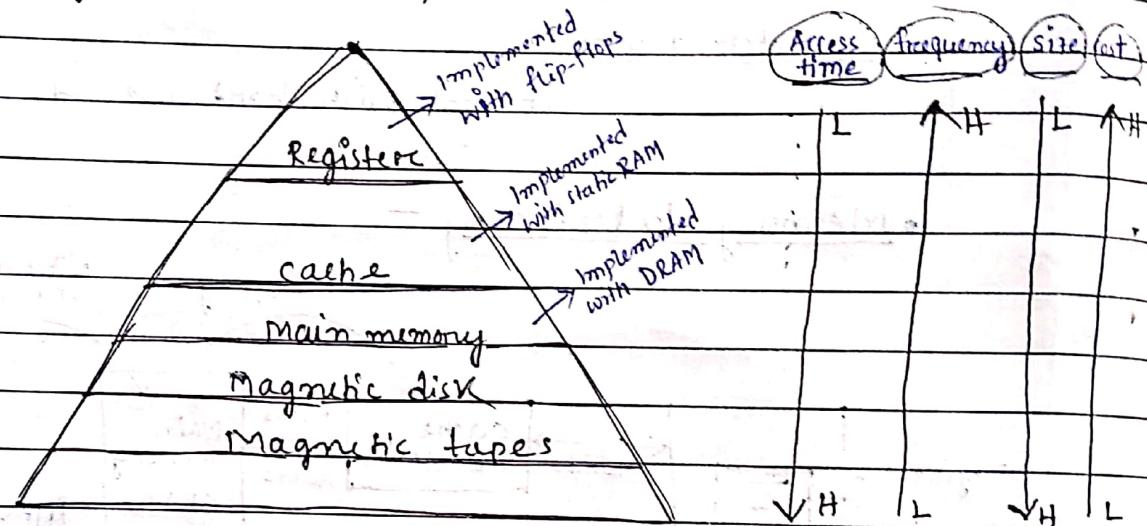
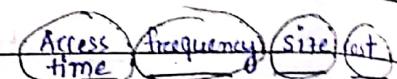
Cache levels }  
 Main memory } Random Access.



Magnetic disk → semi Random Access.



Magnetic tapes → sequential Access.



→ The purpose of memory hierarchy is to bridge the speed mismatch between fastest processor to slow memory at reasonable cost.

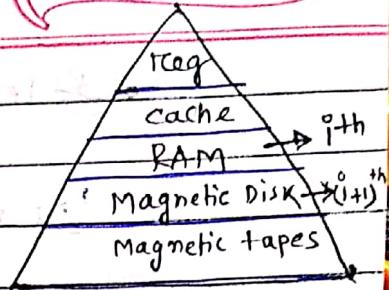
→ The goal of memory hierarchy is to minimize average access time of entire memory system.

- Level memory -

2 level memory -

Information in

$i^{\text{th}}$   $(i+1)^{\text{th}}$  level



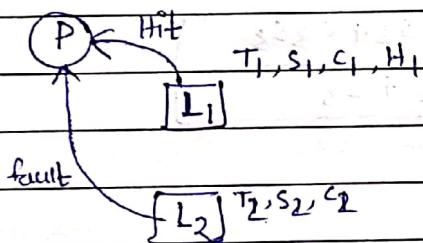
→ If processor refers to  $i^{\text{th}}$  level memory is found them "Hit" otherwise "Miss (or) fault".

→ There are two way in which the processor is connected to various levels of memory

Case 1°

$$T_1 < T_2$$

Cases



$$\text{Hit rate} = \frac{x}{100}$$

$$\text{Miss rate} = \left(1 - \frac{x}{100}\right)$$

$T_1 \rightarrow$  Time to access

$S_1 \rightarrow$  size of level 1 memory.

$c_1 \rightarrow$  cost per bit.

$H_1 \rightarrow$  Hit rate.

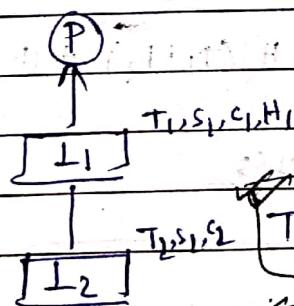
$$T_{\text{avg}} = H_1 T_1 + (1-H_1) T_2$$

$$100 \rightarrow x$$

$$T_{\text{avg}} = \frac{x T_1 + (100-x) T_2}{100}$$

$$= H_1 T_1 + (1-H_1) T_2$$

Case 2°

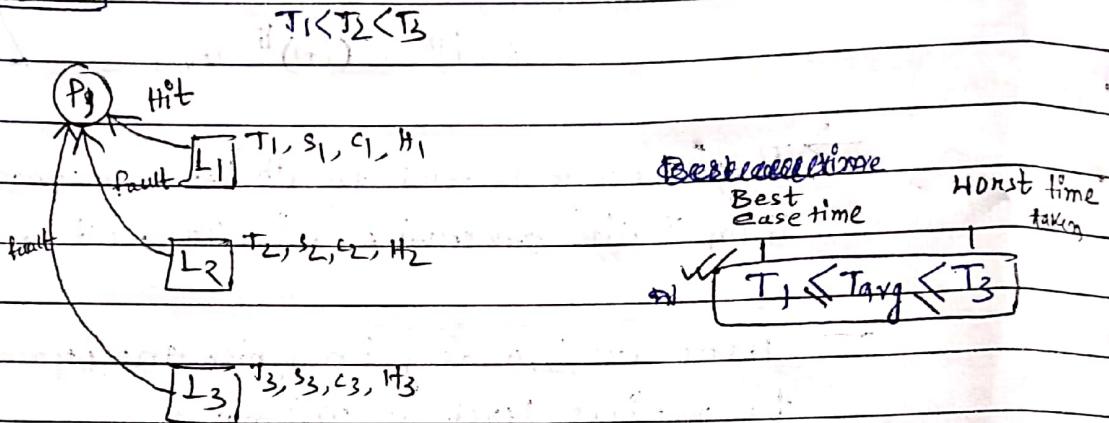


$$T_{\text{avg}} = H_1 T_1 + (1-H_1)(T_1 + T_2)$$

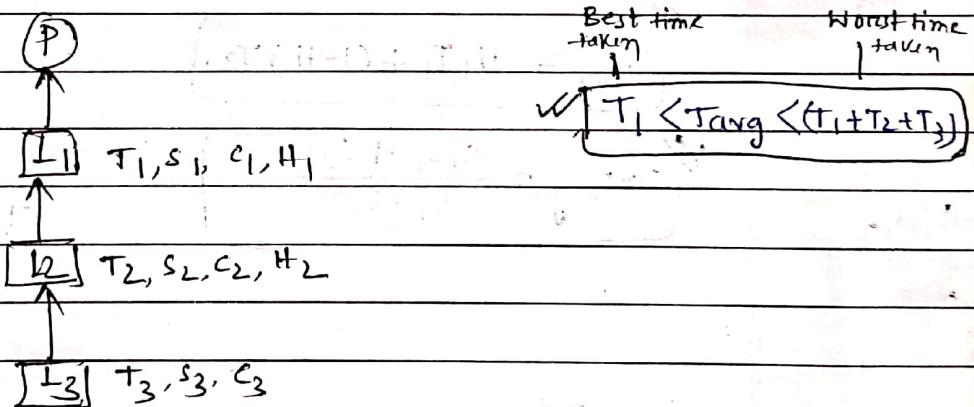
~~$$\text{Cost}_{\text{avg}} = \frac{S_1 c_1 + S_2 c_2}{S_1 + S_2}$$~~

• 3-level memory -

case -1



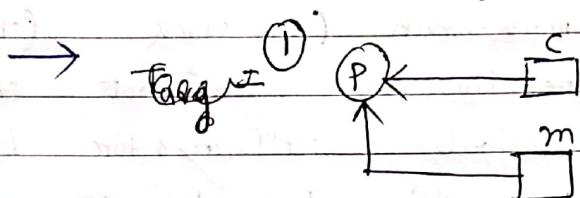
Case -2



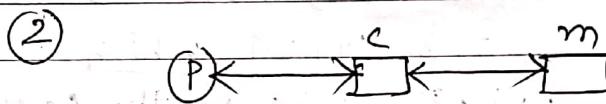
$$\boxed{T_{\text{avg}} = H_1 T_1 + (1-H_2) H_2 (T_1 + T_2) + (1-H_1)(1-H_2)(T_1 + T_2 + T_3)}$$

$$\boxed{C_{\text{avg}} = \frac{S_1 C_1 + S_2 C_2 + S_3 C_3}{S_1 + S_2 + S_3}}$$

Q: The average memory access time for a machine with a cache hit rate of 80% where the cache access time is 5ns and memory access time is 100ns is?



$$\begin{aligned}
 T_{avg} &= H_1 T_C + (1-H_1) T_m \\
 &= (0.8)(5\text{ns}) + (1-0.8)(100\text{ns}) \\
 &= [24 \text{ ns}]
 \end{aligned}$$



$$\begin{aligned}
 T_{avg} &= H_1 T_C + (1-H_1) (T_C + T_m) \\
 &= (0.8)(5\text{ns}) + (1-0.8) (5\text{ns} + 100\text{ns}) \\
 &= 4\text{ns} + (0.2) (105\text{ns}) \\
 &= 4\text{ns} + 21.0\text{ns} \\
 &= [25 \text{ ns}].
 \end{aligned}$$

- Cache replacement policy :-

↳ Replacement policy is required for associative mapping and set associative mapping but not for direct mapping.

↳ Replacement policies are aimed to minimize miss penalty for ~~referencess~~ future references.

## Replacement Policies

**Random**

(No specific criteria to replace the block)

**FIFO**

(The block which entered first is the candidate replacement)

**LRU**

(The block which is not referenced for longest time is replaced)

**LFU**

(the block which references fewer times is replaced)

**Q1**: Consider a direct mapped cache with 8 cache blocks (0-7), if the memory block requests are in the orders 3, 5, 2, 8, 0, 6, 3, 9, 16, 20, 17, 25, 18, 30, 24, 2, 63, 5, 82, 17, 24 which one of the memory blocks will not be in the cache at the end of the sequence?

- A) 3      B) 18      C) 20      D) 30 .**

→	0	8 0 16	24
1	8 17	28 17	
2	2 18	2 82	
3	3 ✓		
4	20 ✓		
5	5		
6	6 30 ✓		
7	63		

no. of Block  
(line) = 8  
cache



$i = j \bmod 8$

o line  
 $i \rightarrow$  block no. of cache.  
 $j \rightarrow$  block no. of MM.

number of lines on cache (8).

main memory

Q: 2 Consider a 1-Way set associative mapping with 16 cache blocks, the more the memory block request are in the order (0, 255, 1, 1, 3, 8, 133, 159, 216, 219, 18, 32, 73, 92, 155) which one of the following memory block will ~~not~~ be in the cache if LRU is used.

~~present~~

✓) 3

✗) 8

✓) 129

✗) 216

⇒,

S <sub>0</sub>	0, 1, 8, 216 8, 18, 32, 92
S <sub>1</sub>	1, 133, 73,
S <sub>2</sub>	
S <sub>3</sub>	(2, 5, 8, 15), 219, 0, 63, 155

(LRU - Least recently used)

Set-number = (i) mod 4

i → Block number.

Cache memory (4-way set associative)

(means ~~each~~ set contain 4 lines)

Q: 3 considers a small 2-way set associative mapping with a total of 4 blocks, for choosing the block to be replaced we LRU scheme, the number of cache misses for the following sequence of block addresses 8, 12, 0, 12, 8 is 4 ?

⇒

S <sub>0</sub>	8, 12, <del>∅</del> , 0, 12, 8
S <sub>1</sub>	

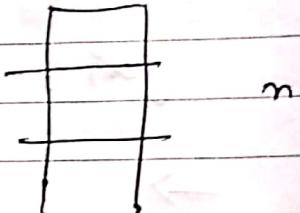
s.no = (i) mod 2

$$\text{miss rate} = \left( \frac{4}{5} \times 100 \right) = 80\%$$

$$\text{hit rate} = \left( \frac{1}{5} \times 100 \right) = 20\%$$

**Q:9** Consider a 2-way set associative mapping consisting of 2 memory blocks and 2c cache blocks, the cache location for the memory block 'K' is \_\_\_\_\_

- a)  $K \bmod 2c$
- b)  $K \bmod 2^c$
- c)  $K \bmod c$
- d)  $K \bmod K$



$$\text{no. of lines in cache} = 2c$$

2-way set-associative.

$$\text{no. of sets} = \frac{2c}{2} = c$$

$$\text{set no} = K \bmod c$$

**Q:5**

Consider the cache has 4 blocks, for the memory reference (5, 12, 13, 17, 4, 12, 13, 17, 2, 13, 19, 13, 43, 61, 19) what is the hit ratio for the following cache replacement algorithms -

- (i) FIFO      (ii) LRU      (iii) Direct mapping .
- (iv) 2-way set Associate (LRU) :

→ ① **FIFO**.

( $\overset{m}{5}, \overset{m}{12}, \overset{m}{13}, \overset{m}{17}, \overset{m}{4}, \overset{m}{12}, \overset{m}{13}, \overset{m}{17}, \overset{m}{2}, \overset{m}{13}, \overset{m}{19}, \overset{m}{13}, \overset{m}{43}, \overset{m}{61}, \overset{m}{19}$ )

5	# 43
12	2 61
13	19
17	13

$$\text{hit ratio} = \left( \frac{5}{15} \times 100 \right)$$

$$\text{miss ratio} = \left( \frac{10}{15} \times 100 \right)$$

→ (II) [LRU] (  $\frac{m}{5}, \frac{m}{12}, \frac{m}{13}, \frac{m}{17}, \frac{m}{4}, \frac{\checkmark}{12}, \frac{\checkmark}{13}, \frac{\checkmark}{17}, \frac{\checkmark}{2}, \frac{m}{13}, \frac{\checkmark}{19}, \frac{m}{13}, \frac{\checkmark}{13}, \frac{m}{43}, \frac{m}{61}, \frac{\checkmark}{19}$  )

(  $\underline{5}, \underline{12}, \underline{13}, \underline{17}, \underline{4}, \underline{12}, \underline{13}, \underline{17}, \underline{2}, \underline{13}, \underline{19}, \underline{13}, \underline{13}, \underline{43}, \underline{61}, \underline{19}$  )

$$\text{hit ratio} = \left( \frac{6}{15} \times 100 \right) \quad \text{miss ratio} = \left( \frac{9}{15} \times 100 \right)$$

→ (III) [ Direct mapping ] (  $\frac{m}{5}, \frac{m}{12}, \frac{m}{13}, \frac{m}{17}, \frac{m}{4}, \frac{m}{12}, \frac{m}{13}, \frac{m}{17}, \frac{m}{2}, \frac{m}{13}, \frac{m}{19}, \frac{m}{13}, \frac{m}{13}, \frac{\checkmark}{43}, \frac{m}{61}, \frac{m}{19}$  )

0	12	X 12 2
1	5	X 5 12 X 13 X 17 61
2		
3	13	43 19
4 (G) 15		
5		

$$\text{line.no} = (\text{B.NO}) \bmod 4$$

$$\text{cache hit ratio} = \left( \frac{1}{15} \times 100 \right)$$

$$\text{miss ratio} = \left( \frac{14}{15} \times 100 \right)$$

→ (IV) [ 2-Way set Associate (LRU) ] —

(  $\frac{m}{5}, \frac{m}{12}, \frac{m}{13}, \frac{m}{17}, \frac{m}{4}, \frac{m}{12}, \frac{m}{13}, \frac{m}{17}, \frac{m}{2}, \frac{m}{13}, \frac{m}{19}, \frac{m}{13}, \frac{m}{43}, \frac{m}{61}, \frac{m}{19}$  )

S <sub>0</sub>	12, 4	12, 2
S <sub>1</sub>	5, 13, 17, 13, 17, 13, 19, 13, 13, 61, 19	

$$\text{line.no} = (i \bmod 2)$$

i → Block no.

$$\text{hit ratio} = \left( \frac{5}{15} \times 100 \right)$$

$$\text{miss ratio} = \left( \frac{10}{15} \times 100 \right)$$

Q6

A hierarchical memory system has the following specification. 20MB main storage with access time of 300ns, 256B cache with access time of 50ns, word size 4B, page size 8 words. What will be the hit ratio if the page address trace of a program has the pattern  $0, 1, 2, 3, 0, 1, 3, 0, 1, 2, 4$  following LRU page replacement technique.



$$\text{Cache size} = 256 \text{ B}$$

$$\text{Word size} = 4 \text{ B}$$

$$\text{page size} = 8 \text{ words} = (8 \times 4) \text{ B} = 32 \text{ B}$$

$$\text{No. of cache page} = \frac{\text{size of cache}}{\text{cache page size}} = \frac{256}{32} = 8$$

$(\emptyset, 1, 2, 3, 0, 1, 3, 0, 1, 2, 4)$

$$\text{hit ratio} = \left( \frac{3}{8} \times 100 \right)$$

Q7

Consider an array  $A[100]$  and each element occupies 4 words. A 32-word cache is used and divided into 8-word blocks.

a) What is the hit ratio for the statement.

for ( $i=0, i < 100, i++$ )

$$A[i] = A[i] + 10.$$

→ Cache size = 32 W

$$\text{Block size} = 8 W$$

$$\text{no. of block in cache} = \frac{32}{8} = 4 \text{ block.}$$

$B_0$	$A_0$	$A_1$
$B_1$	$A_2$	$A_3$
$B_2$	$A_4$	$A_5$
$B_3$	$A_6$	$A_7$

$$\frac{A_0}{m}, \frac{A_0}{H}, \frac{A_1}{H}, \frac{A_1}{H}$$

$$\frac{A_2}{m}, \frac{A_2}{H}, \frac{A_3}{H}, \frac{A_3}{H}$$

Cache.

$$\text{hit ratio} = \left( \frac{3}{4} \times 100 \right) \checkmark = 75\% \text{ hit rate.}$$

Q. 8

Consider an array has 100 elements and each elements occupies 4 words. A 32 bit word cache is used and divided into a blocks of 8 words. What is the hit rate of.

for  $i = 0; i < 10; i++$

for  $j = 0; j < 10; j++$

$$A[i][j] = A[i][j] + 10.$$



Cache size = 32 Word

$i = 0 \ 10 \dots$

Block size = 8 W

$i = 1 \ 11 \dots$

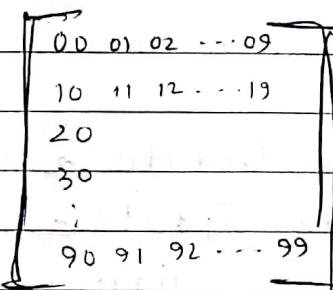
No. of Block = 4.

$i = 2 \ 12 \dots$

$B_0$	$A_{00}$	$A_{04}$
$B_1$		
$B_2$		
$B_3$		

array =  $[A_0, A_4, A_8, \dots, A_{100}]$

RMO



$00 \ 01 \ 02 \ \dots \ 09$

$10 \ 11 \ 12 \ \dots \ 19$

$20 \ 21 \ 22 \ \dots \ 29$

$30 \ 31 \ 32 \ \dots \ 39$

$40 \ 41 \ 42 \ \dots \ 49$

$50 \ 51 \ 52 \ \dots \ 59$

$60 \ 61 \ 62 \ \dots \ 69$

$70 \ 71 \ 72 \ \dots \ 79$

$80 \ 81 \ 82 \ \dots \ 89$

$90 \ 91 \ 92 \ \dots \ 99$

cache.

RMO :

00	01		
m	H	H	H
00	00	01	01

m	H	H	H
00	00	01	01
R	W	R	W

$$TM = (50)1$$

$$TH = (50)43$$

$$TR_{\text{reference}} = (50)1$$

$$\text{Hit ratio} = \left( \frac{3}{4} \times 100 \right) \\ = 75\%$$

(column major order)

CMO :

m	H	m	H
00	00	01	01

$$TM = (50)2$$

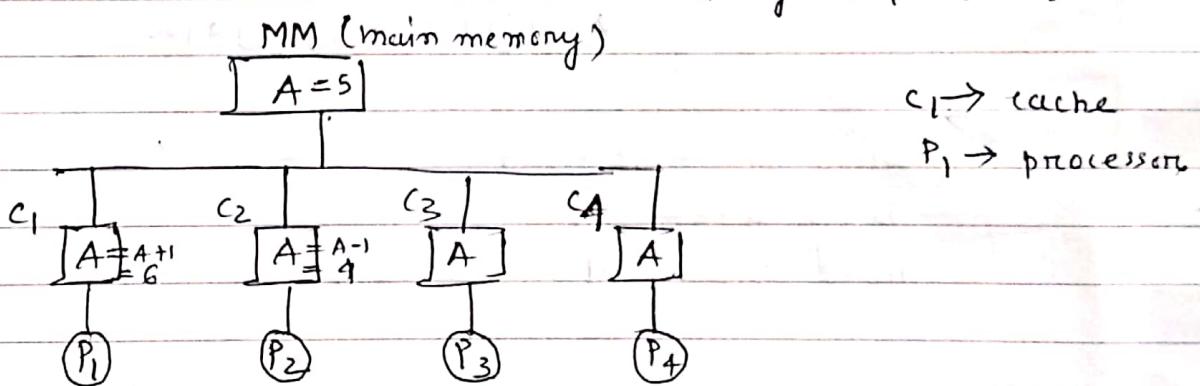
$$TH = (50)2$$

$$TR = (50)4$$

$$\text{Hit ratio} = \left( \frac{2}{4} \times 100 \right) \\ = 50\%$$

- Cache Coherence Problem :

→ cache coherence problem: Multiple copies of same data can exist in different cache simultaneously, and if processors are allowed to update their own copies freely an inconsistent view of memory can result.

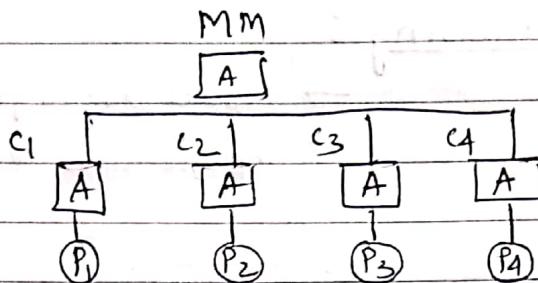


- Methods to avoid cache coherence problem -



There are 4-method to avoid cache coherence problem

- 1) Write Update  $\rightarrow$  write through.
- 2) Write Update  $\equiv$  write back.
- 3) Write Invalidate  $\rightarrow$  write through.
- 4) Write Invalidate  $\rightarrow$  write back.



1) Write update - write through : update simultaneously of a word in cache & - M memory

2) Write update - write back : The updation of MM is postponed until the associated block is replaced.

3) Write Invalidate - write through : changing a value in MM. ~~value in one cache will be invalidated to other cache.~~

4) Write Invalidate - write Back :

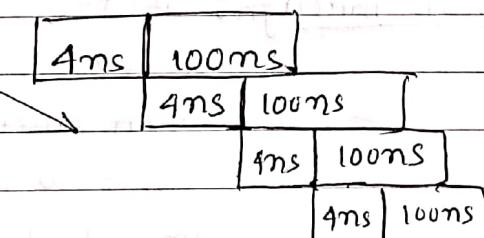
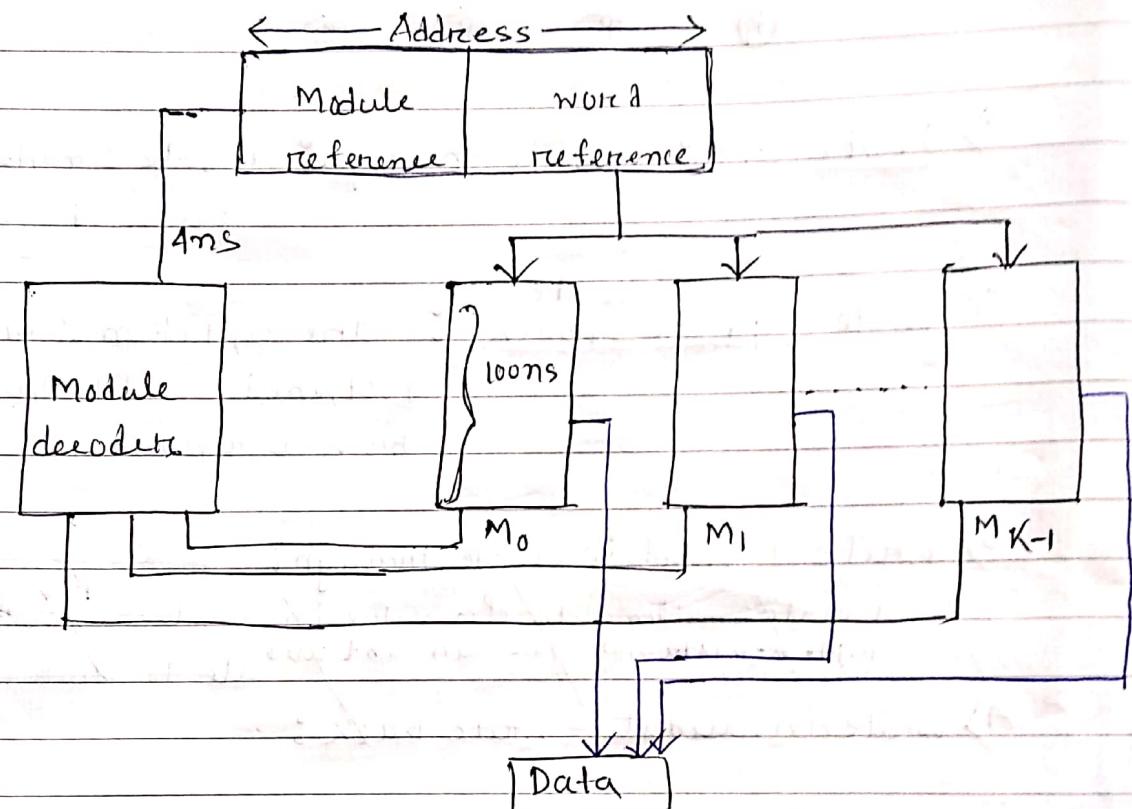
3) write invalidate - write through : if processor  $P_1$  ~~changed~~ a value and change, then other processor  $P_2$  can't do the value will be invalidated for other processor and update simultaneously of value in ~~MM~~ MM.

4) Write Invalidate - write Back :

if  $P_1$  ~~change~~ a value then this value will be Invalid to other processor and updation of MM is postponed until the associated block is replaced.

## • Memory interleaving :

- "Reduce average access time".
- "Improve data transfers rate".



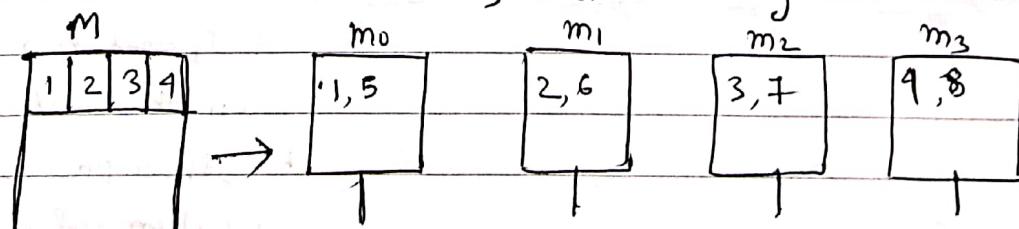
$$4(4\text{ms}) + 3(100\text{ms})$$

if read ' $K$ ' words then, total time taken,  $T = K(4\text{ms}) + 100\text{ms}$ .

AND

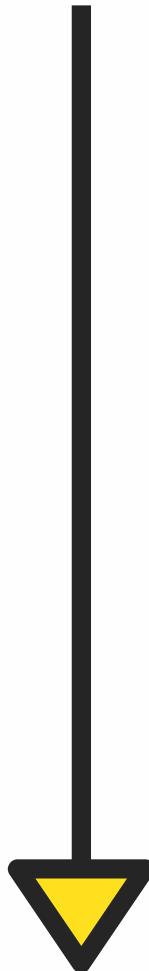
without Interleaving concept, total time,  $T = K(100\text{ms})$ .

### Memory interleaving -



**TO DOWNLOAD THE COMPLETE PDF**

**CLICK ON THE LINK  
GIVEN BELOW**



[WWW.GATENOES.IN](http://WWW.GATENOES.IN)

**GATE CSE NOTES**