CS & IT ENGINEERING

COMPUTER ORGANIZATION AND ARCHITECTURE

Cache Organization

Lecture No.- 03

By- Vishvadeep Gothi sir

Recap of Previous Lecture

Topic Cache Write

Topic Write Through & Write Back

Topic Write Allocate & No Write Allocate

Topics to be Covered

Topic Cache Mapping

Topic Direct Mapping

Topic Tag

Topic Index

#Q. Size of data sent to main memory from CPU:

1. For write hit, when a write through cache is used? ⇒ 1 byte or 1 data item size

2. For write miss, when a write through cache is used? ⇒ 1 byte or 1 data item size

3. For write hit, when a write back cache is used? ⇒ 0

4. For write miss, when a write back cache is used? ⇒ 0

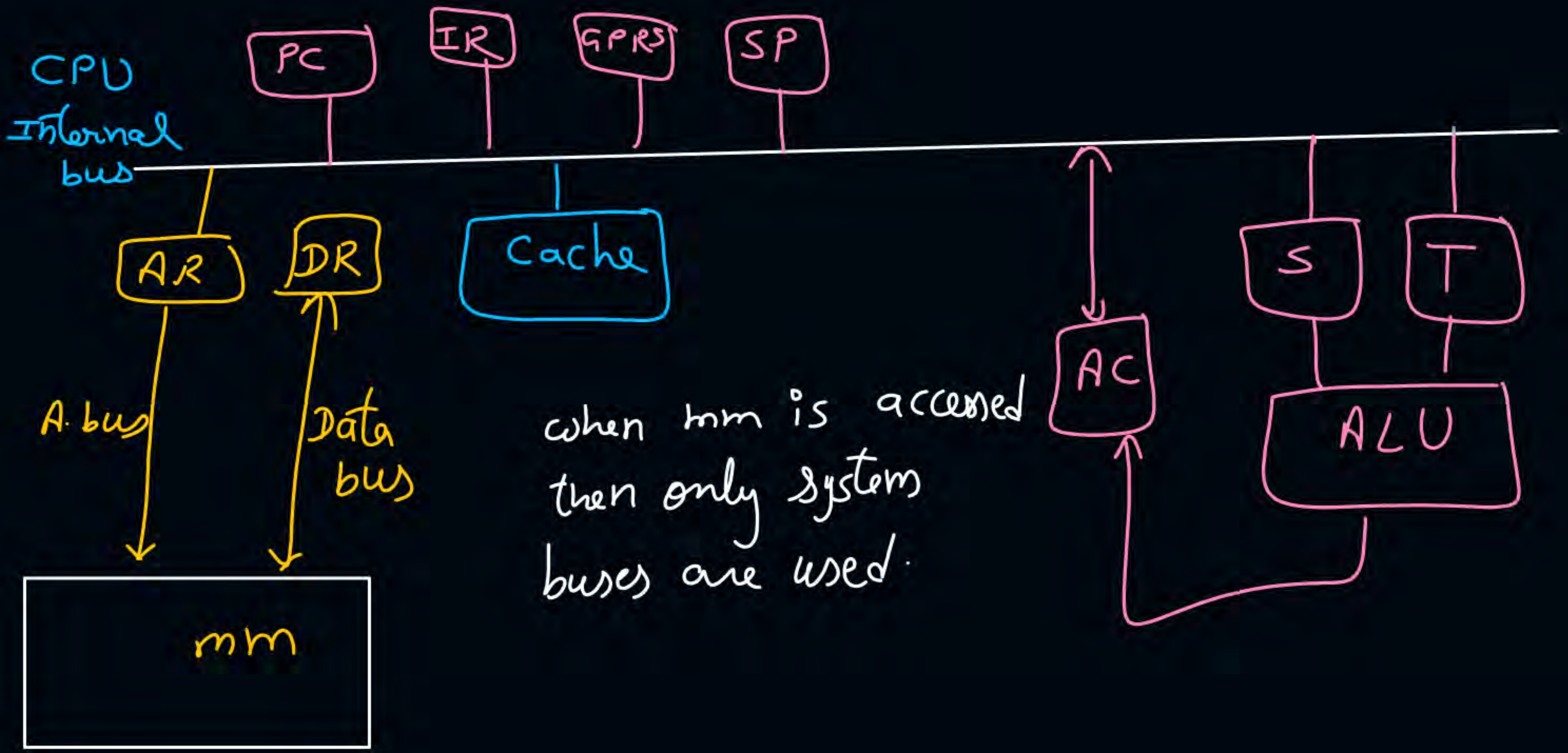
#Q. Size of data sent from main memory to cache:

1. For write hit, when a write through cache is used? ⇒ 0

2. For write miss, when a write through cache is used? ⇒ 0

3. For write hit, when a write back cache is used? ⇒ 0

4. For write miss, when a write back cache is used? ⇒ 1 block



when mm is accessed then only system buses are used.

[Question]

#Q. Consider a computer with the following features:

- 90% of all memory accesses are found in the cache (hit ratio = 0.9)
- The block size is 2 words and the whole block is read on any miss
- The CPU sends references to the cache at the rate of 10^7 words per second
- 25% of the above references are writes (writes = 25%, reads = 75%)
- The bus can support 10^7 words per second, read or writes (total bus bandwidth =10^7)
- The bus reads or writes a single word at a time
- Assume at any one time, 30% of the block frames in the cache have been modified

Calculate the percentage of the bus bandwidth used on the average when:

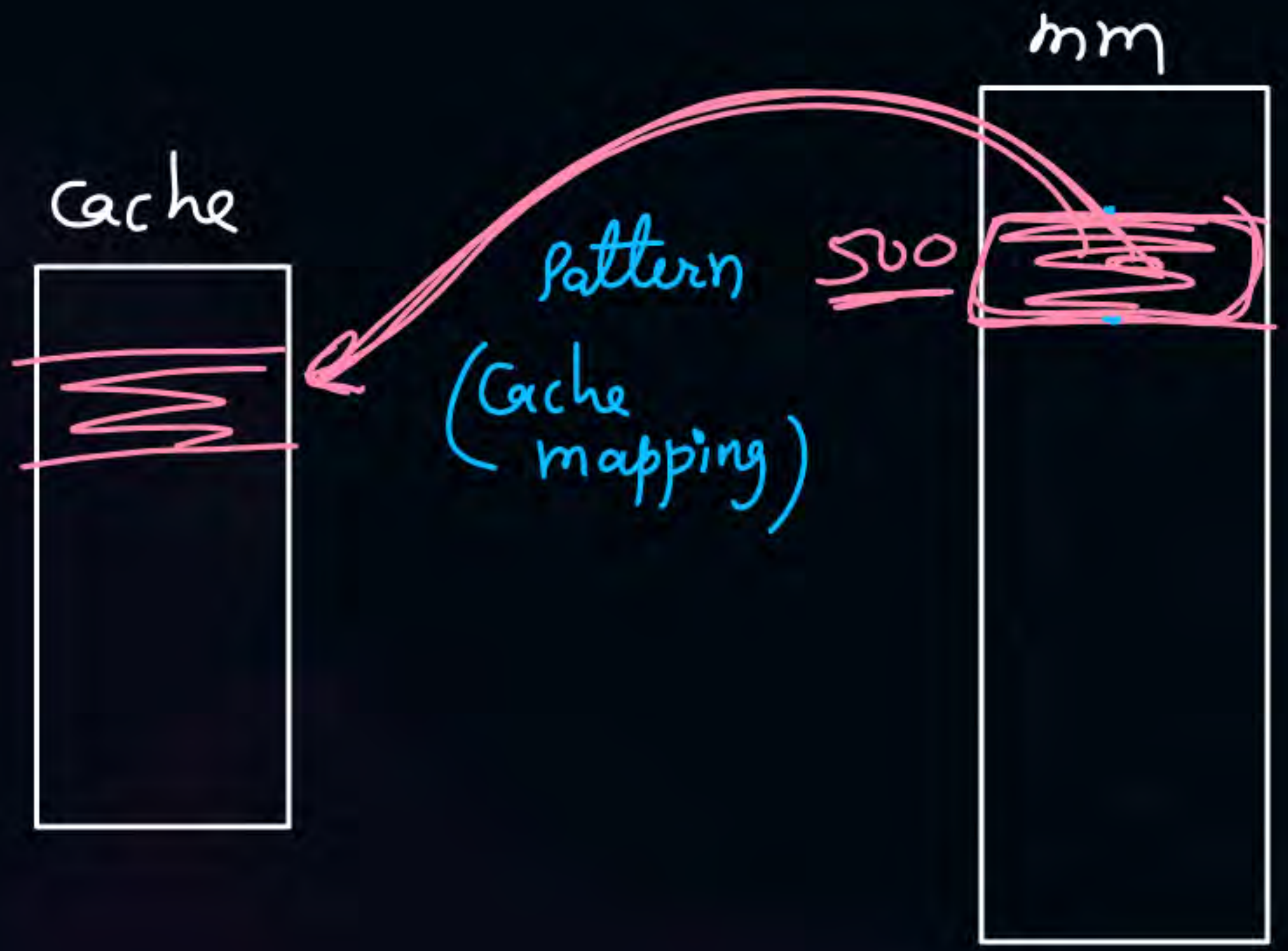
1. Cache is write through with no write allocate
2. Cache is write back with write allocate

Topic : Cache Mapping

CPU always generates mm address.

cm locat^n = f (mm location)

mapping is done on blocks.

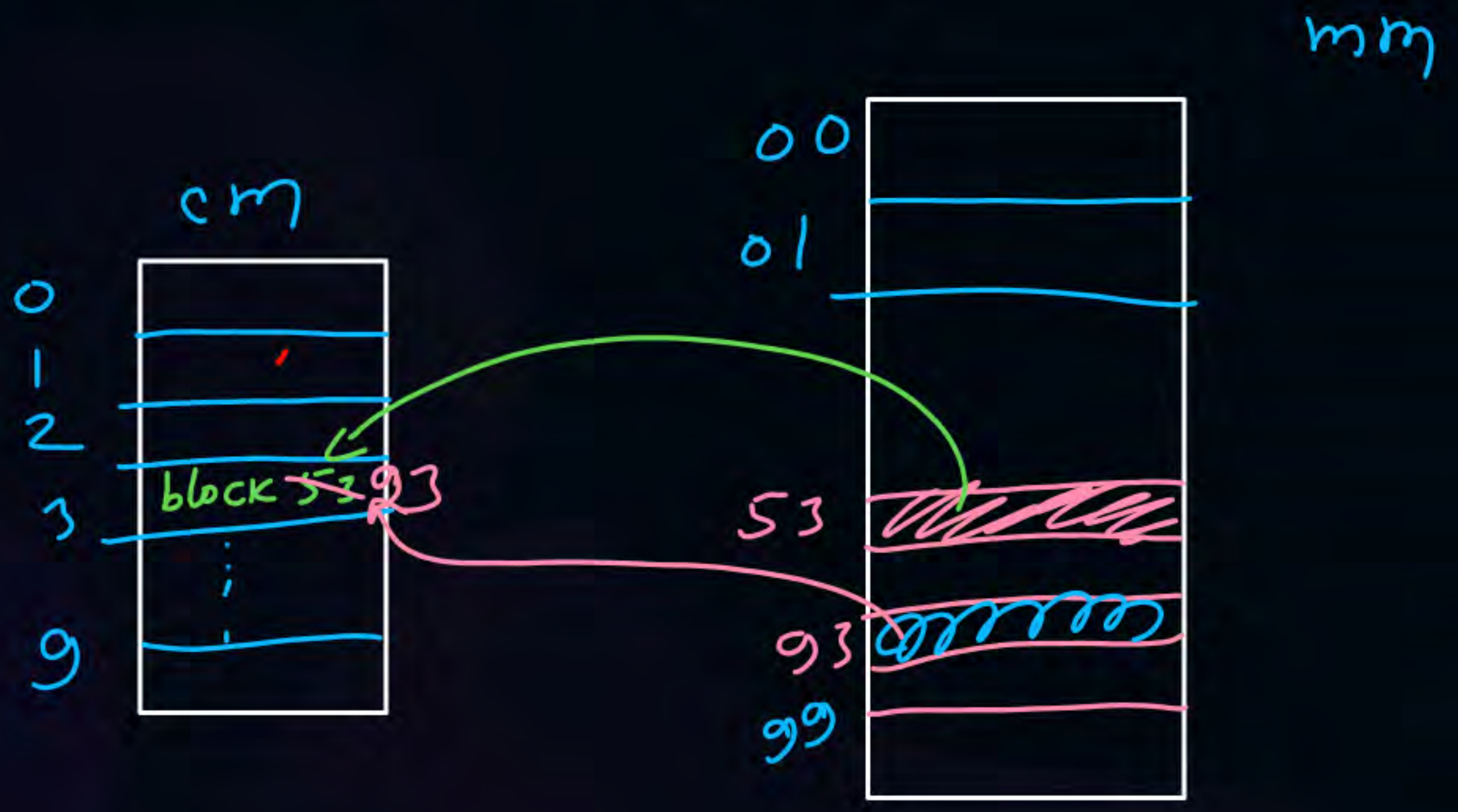


Topic : Cache Mapping

- Direct Mapping
- Set Associative Mapping
- Fully Associative Mapping

Topic : Direct Mapping

- Blocks in cache = 10 (0-9)
- Blocks in Main memory = 100 (00-99)



Topic : Direct Mapping

93

~~53~~

Cache	0	1	2	3	4	5	6	7	8	9
Main Memory	00	01	02	03	04	05	06	07	08	09
	10	11	12	13	14	15	16	17	18	19
	20	21	22	23	24	25	26	27	28	29
	30	31	32	33	34	35	36	37	38	39
	:	:	:	:	:	:	:	:	:	:
	90	91	92	93	94	95	96	97	98	99

Topic : Direct Mapping

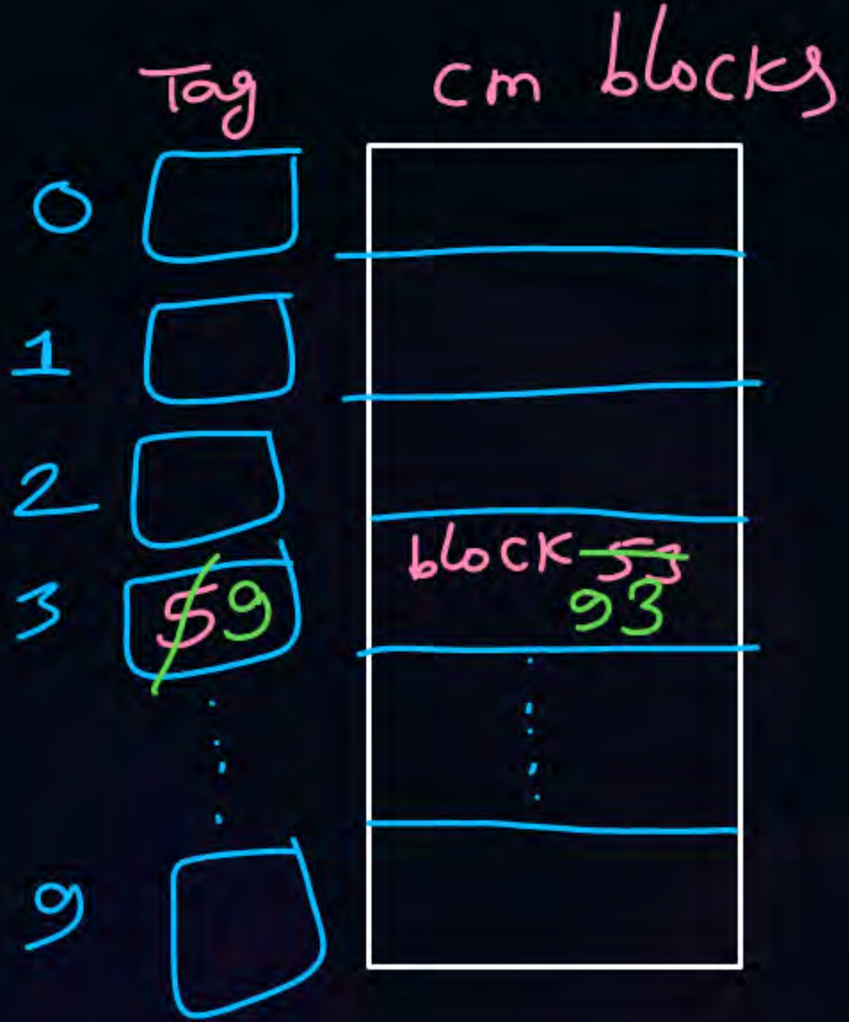
cm block no. = (mm block no.) % (no. of blocks in cache)

$$\text{Tag} = \left\lfloor \frac{\text{mm block no.}}{\text{no. of blocks in cache}} \right\rfloor$$

Topic : Direct Mapping

CPU Request (MM block)	Mapping (CM block no.)	Hit /Miss	Comments
53	$53 \% 10 = 3$ goto block 3 in cache & check	Miss	Bring mm block 53 content in cache at block no. 3.
93	$93 \% 10 = 3$ goto block 3 in cache & check ⇓ block 53 content is in cache	miss	bring mm block 93 content in cache at block no 3, by replacing block 53 content.

Topic : Direct Mapping



mm block no.

Tag	cm block no.

ex:-

mm block no. 53

5	3

Tag

CPU Request mm block 93

9	3

↳ cm block no. 3

⇓

miss

Tag :- It identifies among all competitors which one is present in cache at this block.

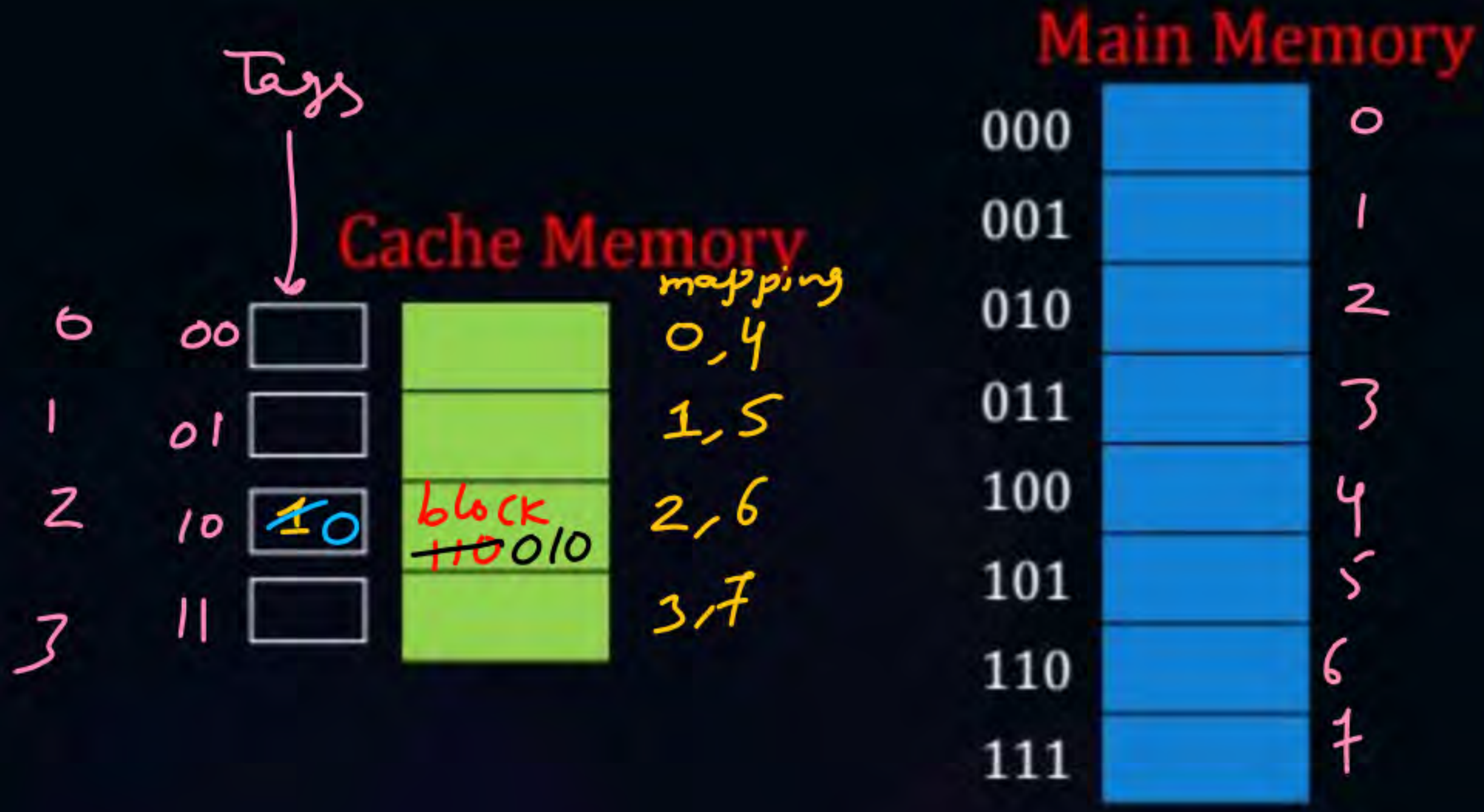
one Tag is stored in cache for each block in cache.

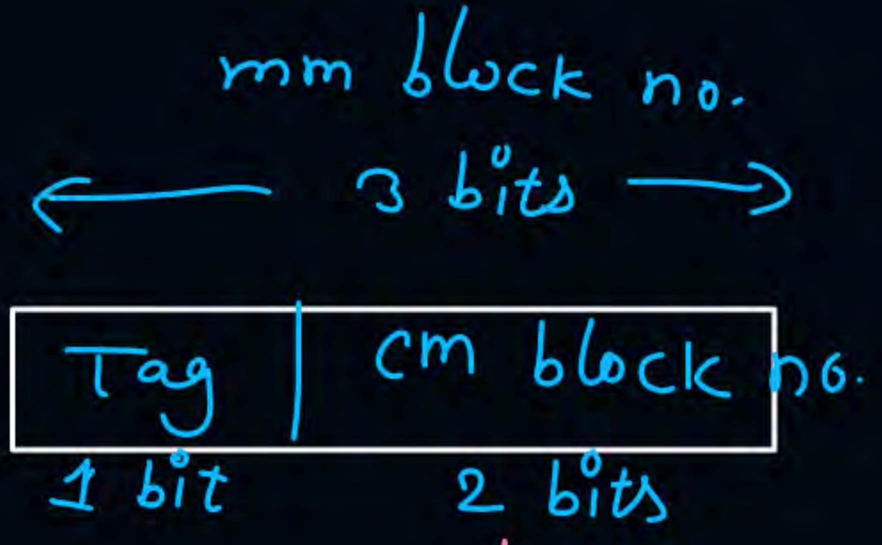
no. of tags = no. of blocks in cache

Topic : Direct Mapping

- Blocks in cache = 4 (00-11) 0 to 3
- Blocks in Main memory = 8 (000-111) 0 to 7

Topic : Direct Mapping





no. of blocks in cache = 4

$= 2^2$

Topic : Direct Mapping

CPU Request (MM block)	Mapping (CM block no.)	Hit /Miss	Comments
$(6)_{10} = (110)_2$	$\underbrace{1}_{\text{Tag}} \mid \underbrace{10}_{\text{cm block no.}}$	miss	bring mm block 110 content in cache at block 10 with tag 1.
$(2)_{10} = (010)_2$	$\underbrace{0}_{\text{Tag}} \mid \underbrace{10}_{\text{cm block no.}}$	miss	bring mm block 010 content in cache at block 10 by replacing mm block 110 and change tag to 0.

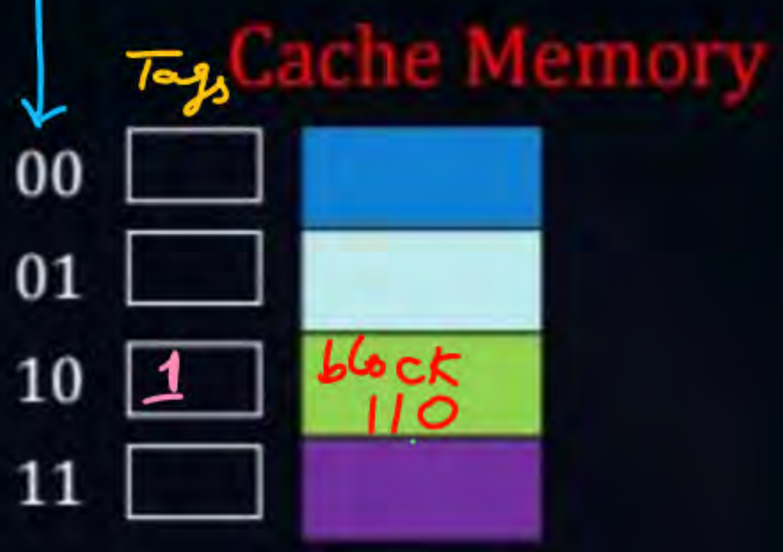
Topic : Direct Mapping

- Blocks in cache = 4 (00-11)
- Blocks in Main memory = 8 (000-111)
- Block Size = 2 Bytes $= 2^1$ B
- Size of Cache memory $= 4 * 2 = 8$ bytes
- Size of Main memory $= 8 * 2 = 16$ bytes $= 2^4$ B
- Size of Main memory address = 4 bits

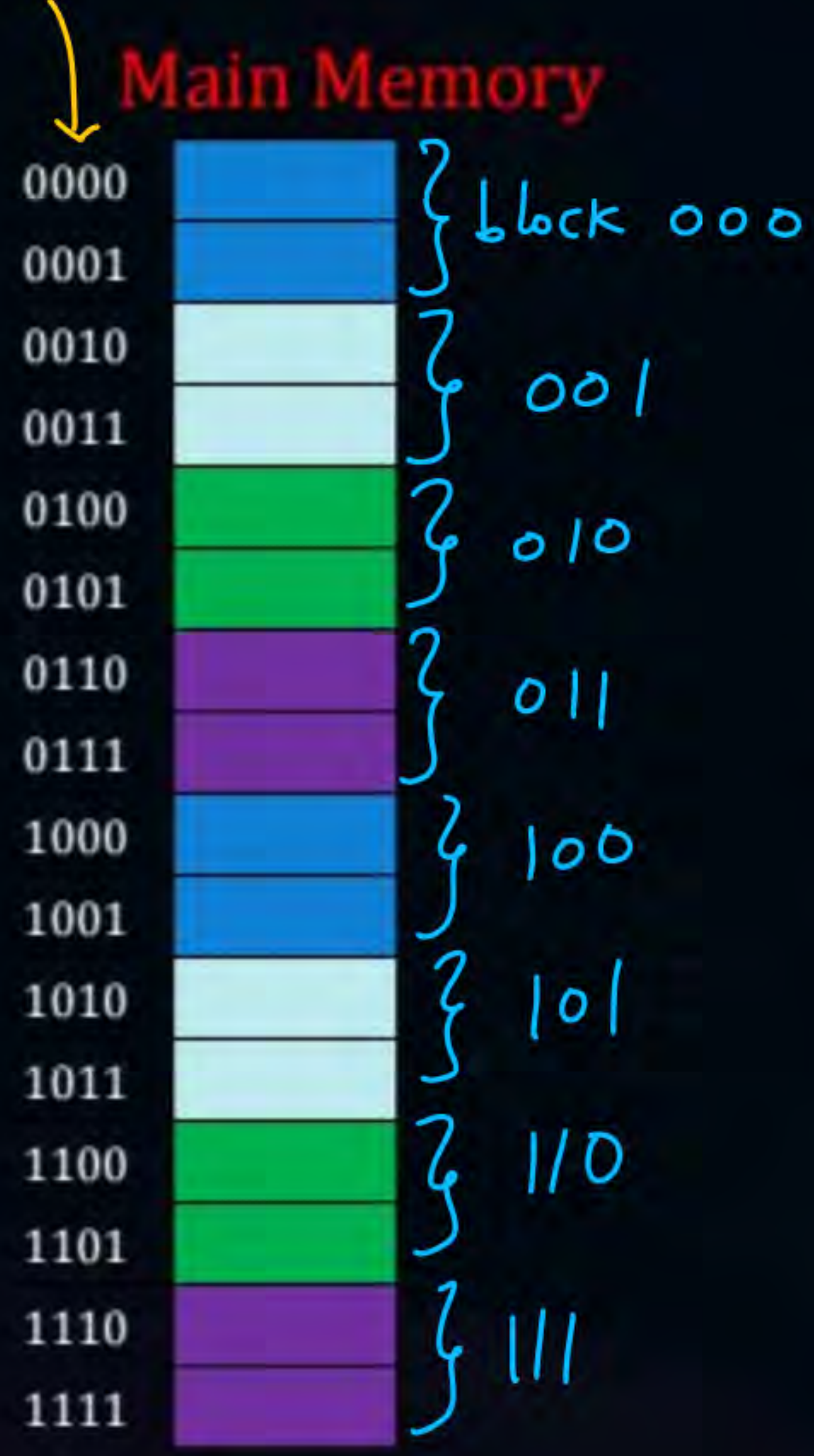
byte addressable

Topic : Direct Mapping

cm block no. (line no.) or Index

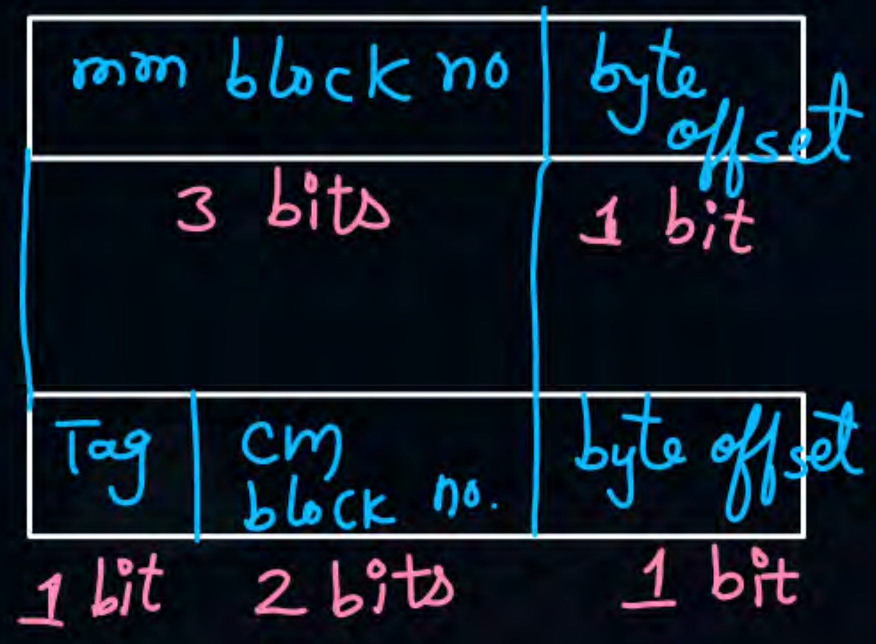


mm addresses



mm add.

4 bits



Topic : Direct Mapping

CPU Request (MM add.)	Mapping(CM block no.)	Hit/Miss	Comments
$(1100)_2$	$\boxed{110 \mid 0}$ mm block no. byte Tag cm block $\boxed{1 \mid 10 \mid 0}$	miss	bring mm block 110 content in cache at block 10 with tag 1
$(1101)_2$	$\boxed{110 \mid 1}$ $\boxed{1 \mid 10 \mid 1}$ Tag cm block no. byte	Hit	CPU accesses byte 1 of this block from cache.

Topic : Direct Mapping

no. of bits in byte offset = $\log_2(\text{block size})$

no. of bits in cm block no. = $\log_2(\text{no. of blocks in cache})$

no. of blocks in cache = $\dfrac{\text{cache size}}{\text{block size}}$

Tag directory size or metadata = no. of blocks in cache * Tag bits

Ques) mm add. = 20 bits

cm size = 512 bytes

block size = 16 bytes $= 2^4 B$

$\Downarrow$

byte offset = 4 bits

Direct mapping

← 20 bits →

Tag	cm block no.	byte
11	5	4

no. of blocks in cache $= \dfrac{512\ B}{16\ B}$

$= \dfrac{2^9}{2^4}$

$= 2^5$

$\Downarrow$

cm block no = 5 bits

Tag directory size $= (2^5 * 11)$ bits

Index = 5 bits

Tag = 11 bits

Ques) mm add. = 32 bits

cm size = 32 KB

block size = 32 B = 2^5 B = byte offset = 5 bits

direct mapping

Index = 10 bits

Tag = 17 bits

Tag directory size = $2^{10} * 17$ bits

= 17 k bits

mm add.

$\longleftarrow 32 \longrightarrow$

Tag	cm block no.	byte offset
17	10	5

$$\text{no. of blocks in cache} = \frac{\cancel{32KB}}{\cancel{32B}}$$

$$= 1K$$

$$= 2^{10}$$

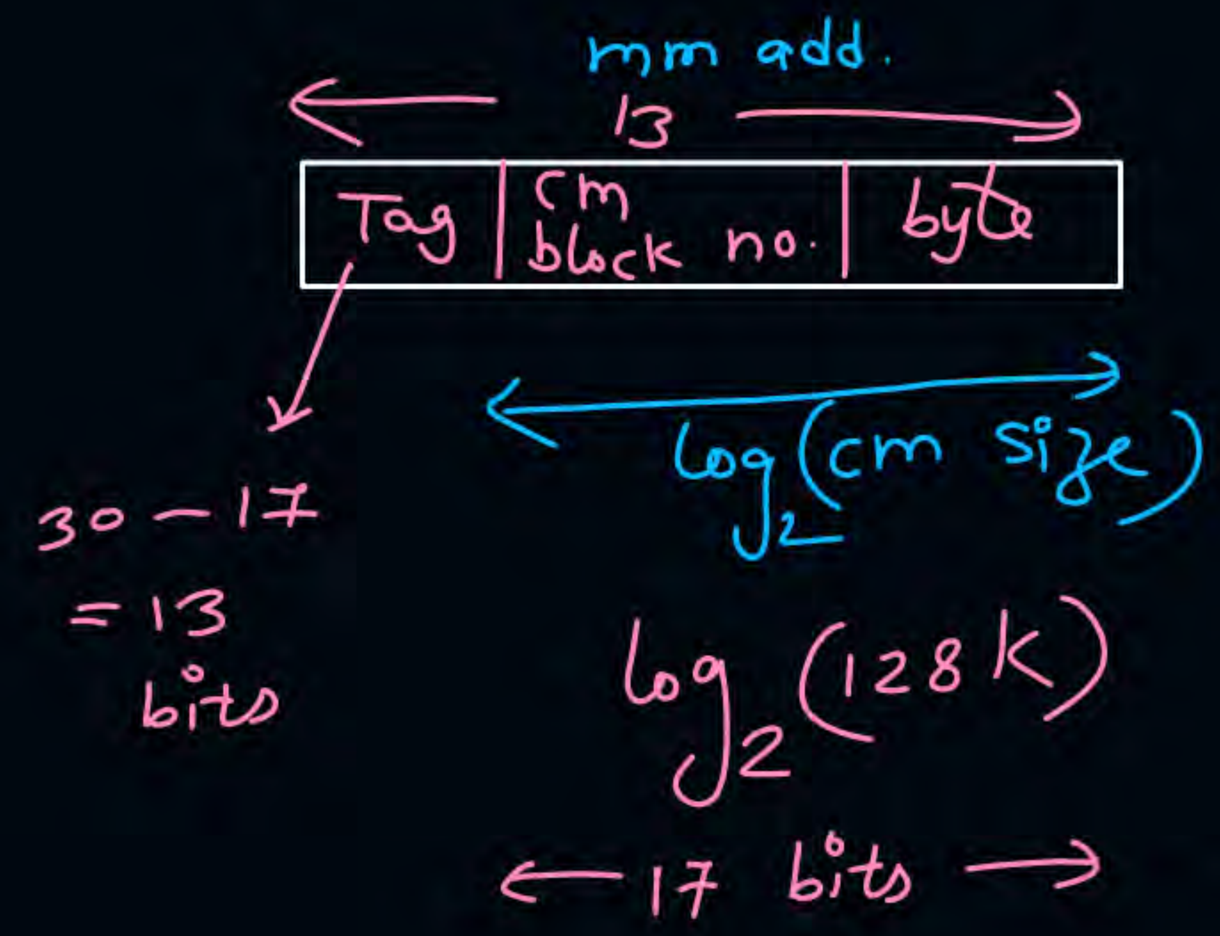
$$\Downarrow$$

cm block no. = 10

Ques) mm add. = 30 bits

cm size = 128 KB

Tag = <u>13</u> bits



$\log_2(\text{cm size})$

$30 - 17$

$= 13$ bits

$\log_2(128\,K)$

17 bits

Ques) Direct mapping

cm size = 256 K bytes

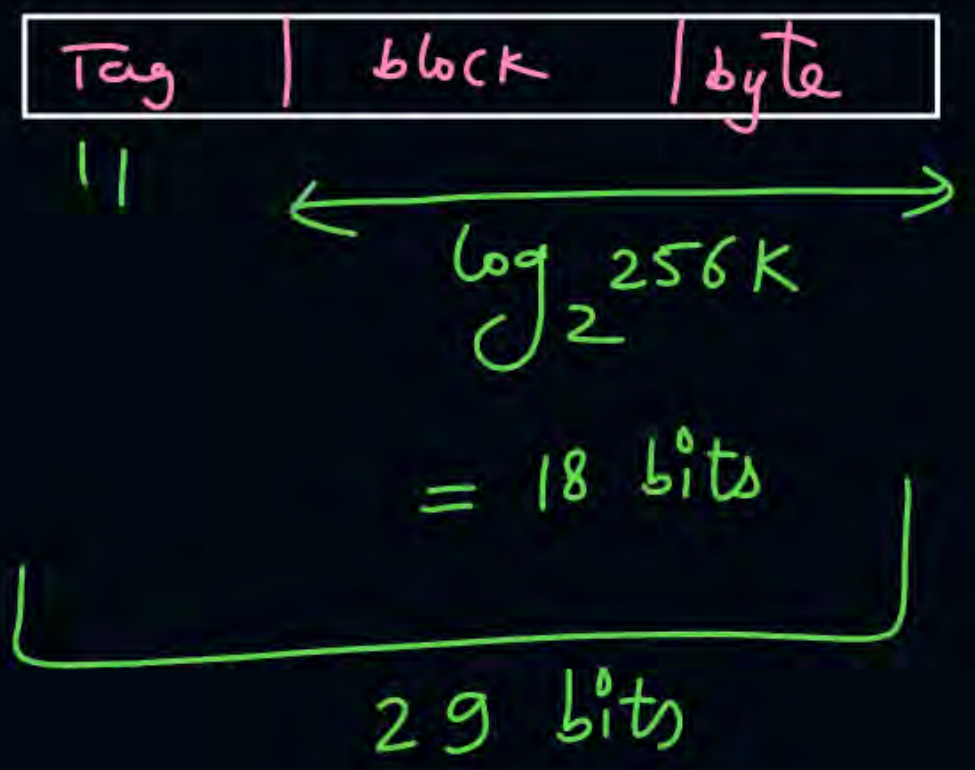
Tag = 11 bits

mm size = 0.5 GB ?

↓

byte addressable

mm add.



Tag	block	byte

11

$\log_2 256K$

= 18 bits

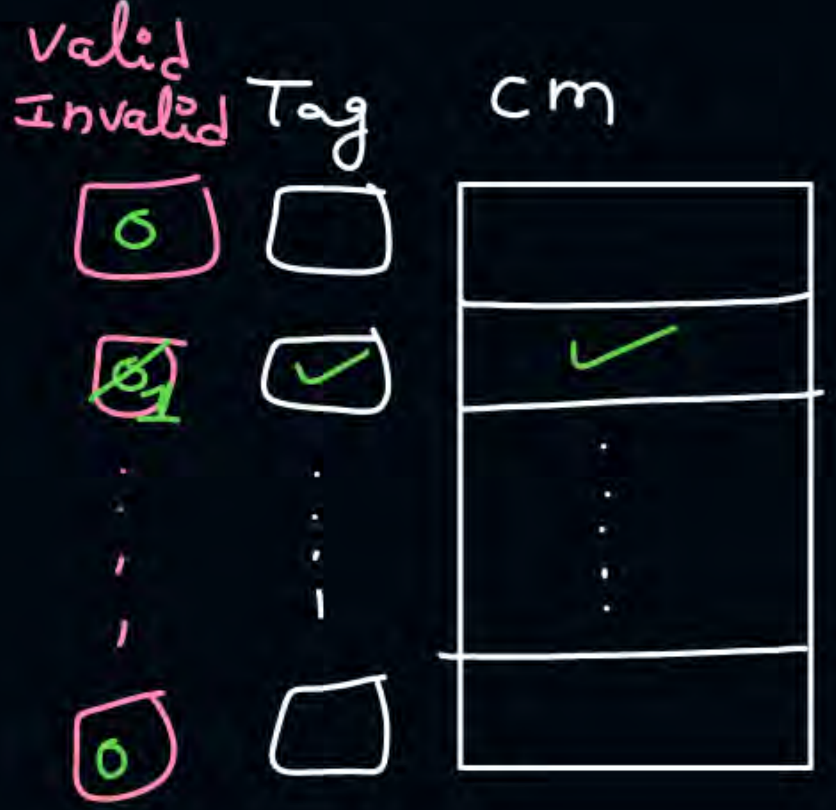
29 bits

mm size = 2^{29} bytes

= 512 MB

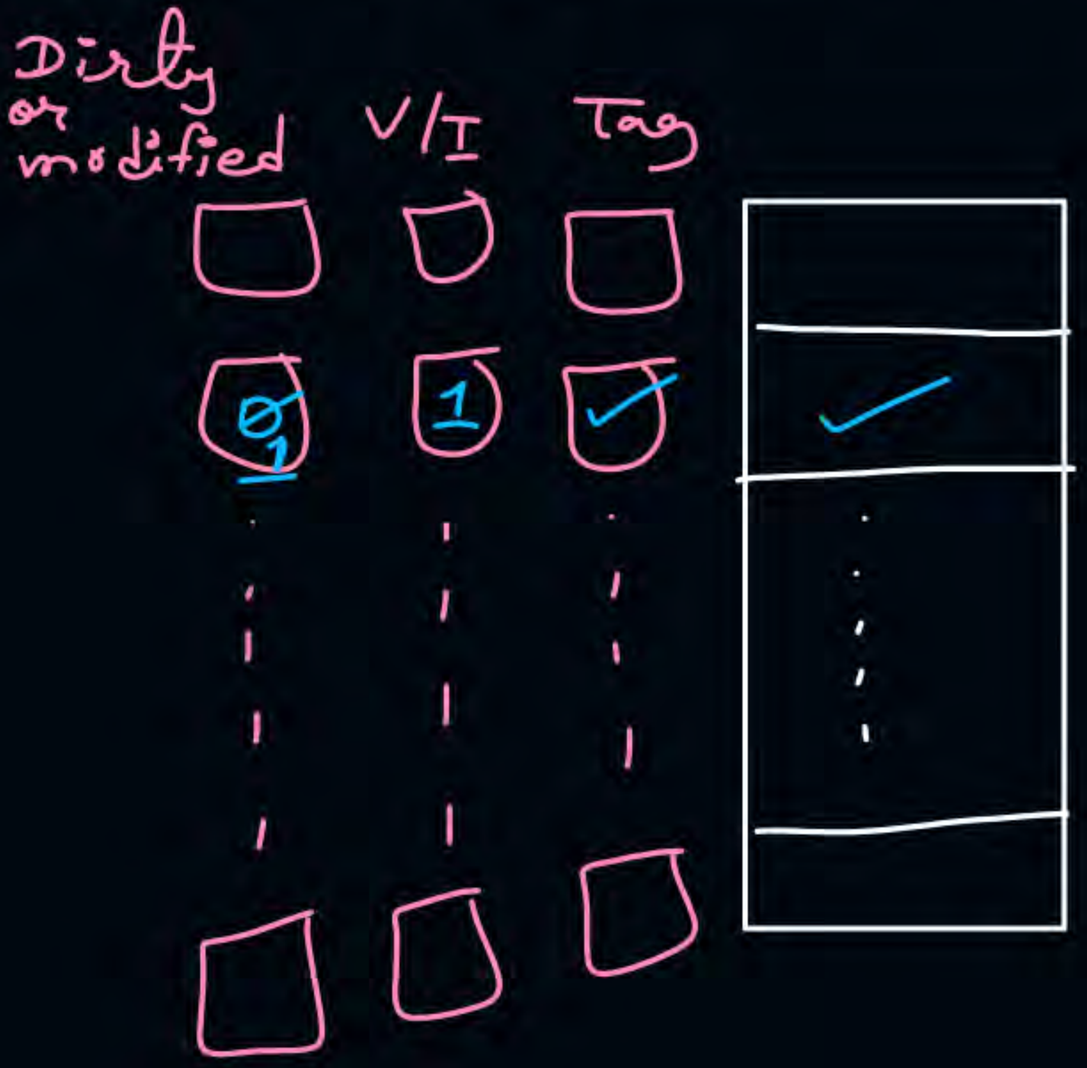
= 0.5 GB

Cache Initialization :-



$$\text{valid/Invalid}\begin{cases} 0 & \text{Invalid} \\ 1 & \text{valid} \end{cases}$$

write back cache performance improvement :-



$$\text{Dirty}\begin{cases}0 & \text{if block is not modified}\\ 1 & \text{if block is dirty}\end{cases}$$

write back the block from cache to mm, which have dirty bit 1.

$$\text{Tag directory size} = \underset{\text{in cache}}{\text{no. of blocks}} * \left(\text{Tag} + \text{extra bits}\right)$$

← 30 →

Tag	block	byte
18	8	4

$\longleftrightarrow$ $\log_2 4k = 12$ bits

Ques) Direct mapping

mm add = 30 bits

cm size = 4K B

block size = 16 B = 2^4 B

for each block in cache tag, 1 valid/Invalid, 1 modified bit stored.

Tag directory size = 5120 bits?

↳ $2^8 * (18 + 1 + 1)$ bits

$= 256 * 20$ bits $=$ 5120 bits Ans.

2 mins Summary

Topic Cache Mapping

Topic Direct Mapping

Topic Tag

Topic Index

Happy Learning

THANK - YOU