CS & IT ENGINEERING

COMPUTER ORGANIZATION AND ARCHITECTURE

Cache Organization

Lecture No.- 06

By- Vishvadeep Gothi sir

Recap of Previous Lecture

Topic Cache Mapping

Topic Direct Mapping

Topic Tag

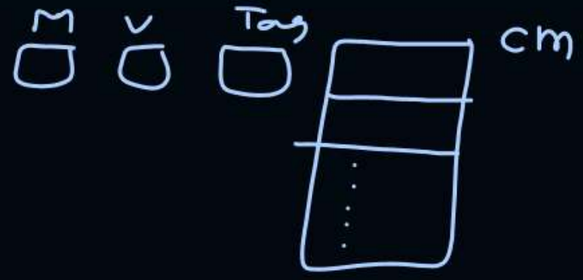
Topics to be Covered

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

- Topic: Direct Mapping
- Topic: Set Associative Mapping
- Topic: Fully Associative Mapping

[MCQ] GATE-2011

#Q. An 8KB direct-mapped write-back cache is organized as multiple blocks, each of size 32-bytes. The processor generates 32-bit addresses. The cache controller maintains the tag information for each cache block comprising of the following.



→ 1 Valid bit

→ 1 Modified bit

Tag → As many bits as the minimum needed to identify 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?

(A) 4864 bits (B) 6144 bits

(C) 6656 bits (D) 5376 bits

32-bits

Tag	cm block no.	byte
19	8	5

$\log_2 8k = \log 2^{13} = 13$ bits

no. of blocks in cm $= \frac{8\,kB}{32B}$

$= \frac{2^3 \cdot 2^{10}}{2^5}$

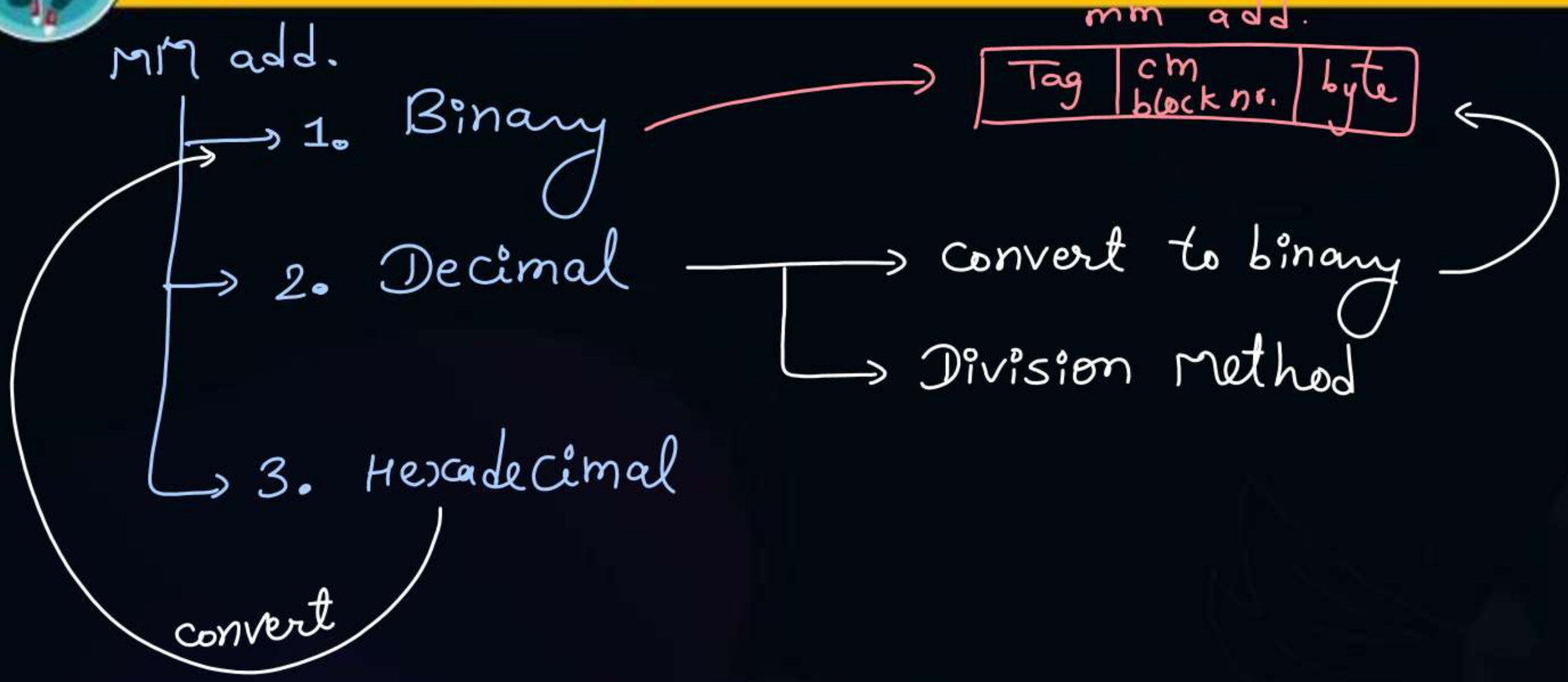
$= 2^8$

Tag directory size $= 2^8 * (19 + 1 + 1)$ bits

$= 2^8 * 21$ bits

$= 5376$ bits

Topic : Calculating CM Block Number from MM Address



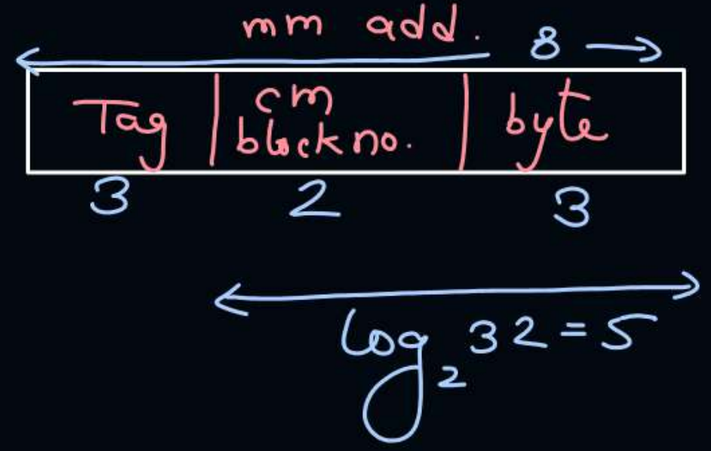
ex:- mm add. = 8 bits

cache size = 32 bytes

block size = 8 bytes

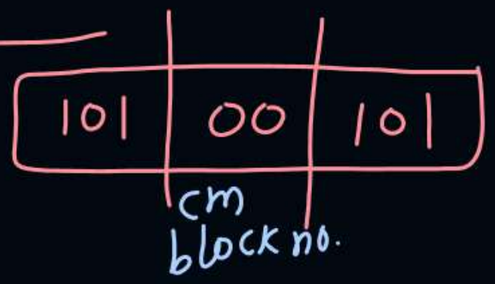
Direct mapping

given mm addresses are mapped to which cm block?

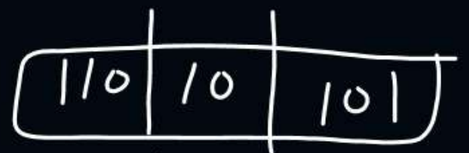


---

mm add
10100101



cm block
$(00)_2 = (0)_{10}$



11010101

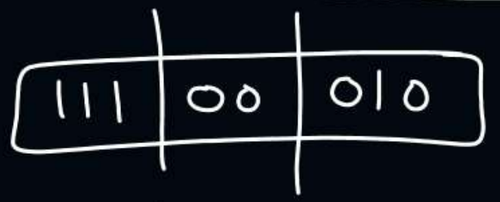
$(10)_2 = (2)_{10}$

11111101

$(11)_2 = (3)_{10}$

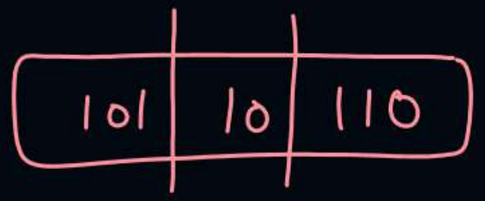
MM add. | cm block no.

$(E2)_{16} = (11100010)_2$



$(00)_2 = (0)_{10}$

$(B6)_{16} = (10110110)_2$



$(10)_2 = (2)_{10}$

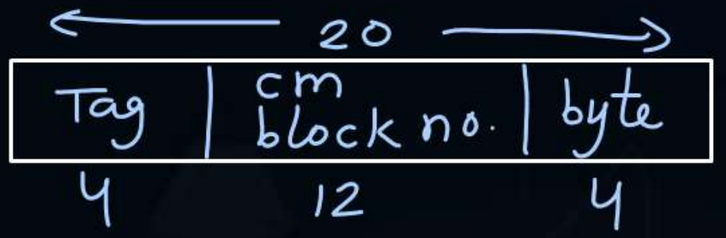
[MCQ]

byte = 4 bits

add. = 20 bits

#Q. Consider a machine with a byte addressable main memory of 2^{20} bytes, block size of 16 bytes and a direct mapped cache having 2^{12} cache lines. Let the addresses of two consecutive bytes in main memory be $(E201F)_{16}$ and $(E2020)_{16}$. What are the tag and cache line address (in hex) for main memory address $(E201F)_{16}$?

Tag | cm block no. | byte no.



A ✓ E, 201

B F, 201 E 201 F

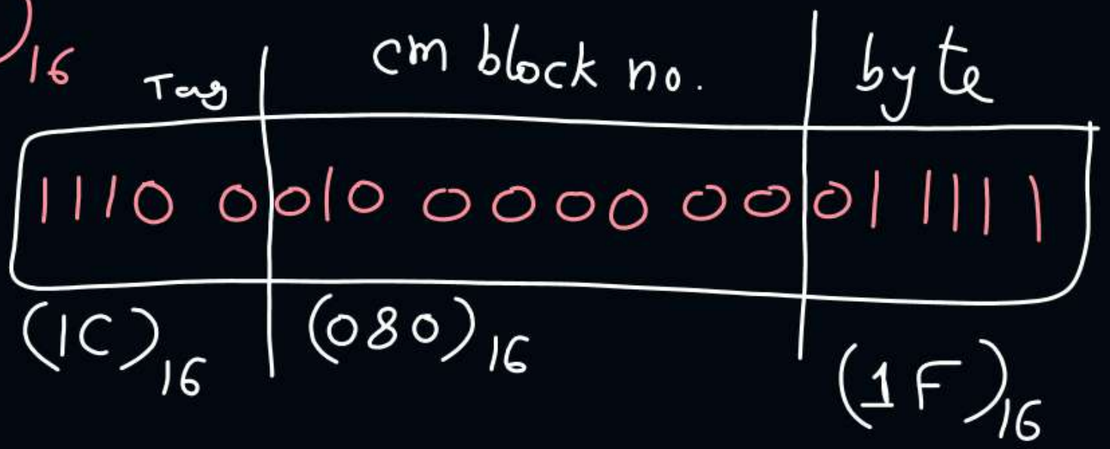
C E, E20

D 2, 01F

Ex:-


20 bits
Tag
cm block no.
byte
5
9
6

$(E201F)_{16}$

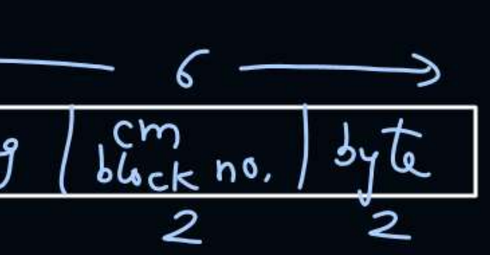

Tag
cm block no.
byte
= 1110 0010 0000 0001 1111
$(1C)_{16}$
$(080)_{16}$
$(1F)_{16}$

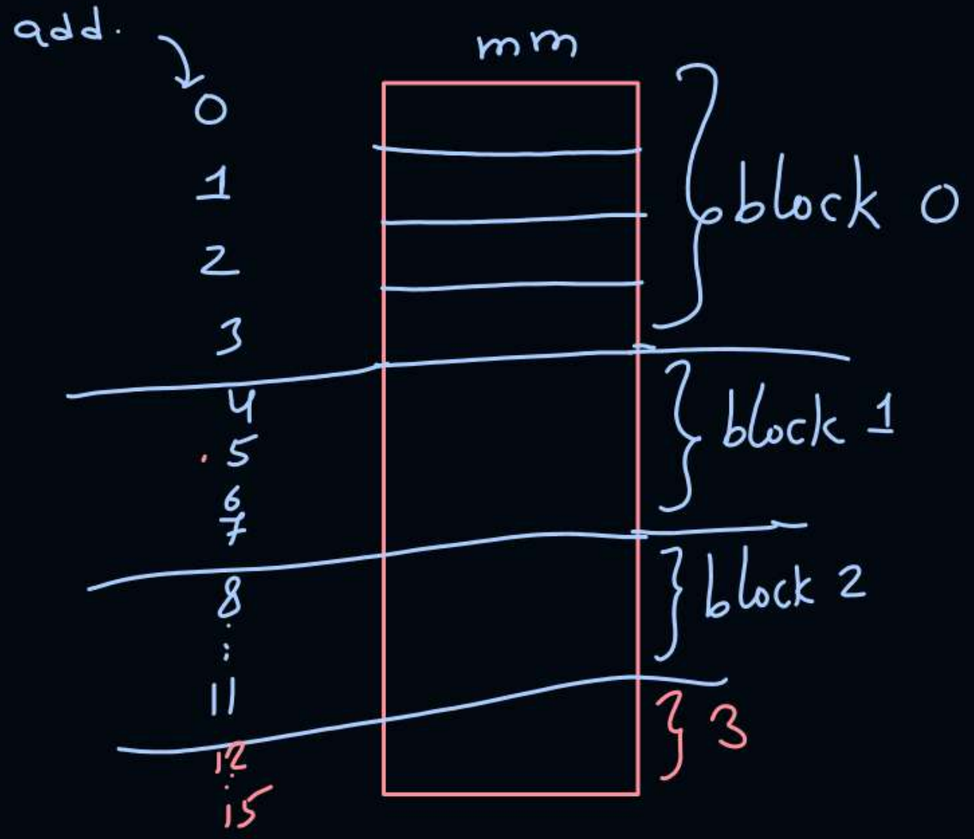
mm add. given in decimal :-

ex:- mm size 64 bytes ⇒ mm add. = 6 bits

cm size 16 bytes } no. of blocks in cache = $\frac{16}{4} = 4$

block size 4 bytes }



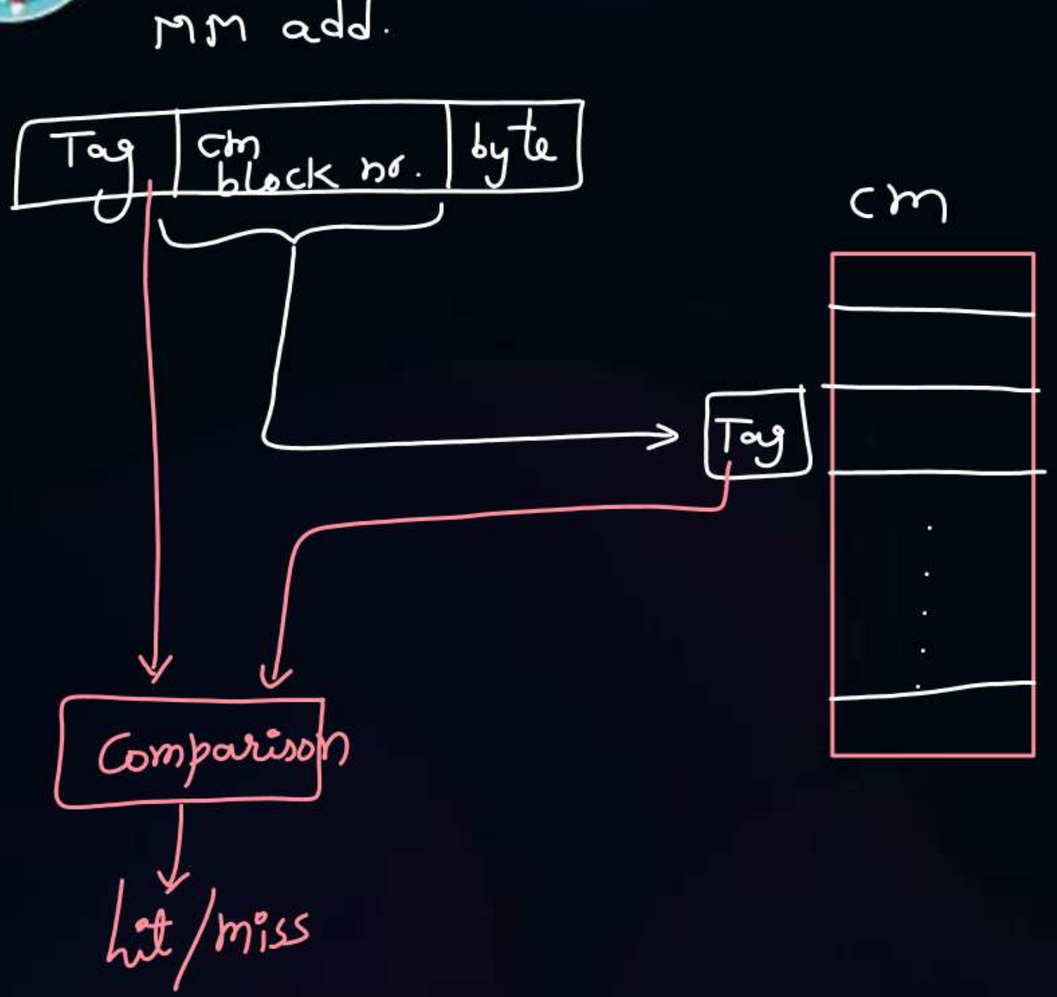


mm add.	mm block no.
11	2
12	3
13	3

$$\text{mm block no.} = \left\lfloor \frac{\text{mm add.}}{\text{block size}} \right\rfloor$$

mm add.	mm block no.	cm block no. (mapping)
50	$\lfloor \frac{50}{4} \rfloor = 12$	$12 \% 4 \Rightarrow 0$
43	$\lfloor \frac{43}{4} \rfloor = 10$	$10 \% 4 \Rightarrow 2$
29	$\lfloor \frac{29}{4} \rfloor = 7$	$7 \% 4 \Rightarrow 3$
63	$\lfloor \frac{63}{4} \rfloor = 15$	$15 \% 4 \Rightarrow 3$

Topic : Checking Hit/Miss in Direct Mapped Cache

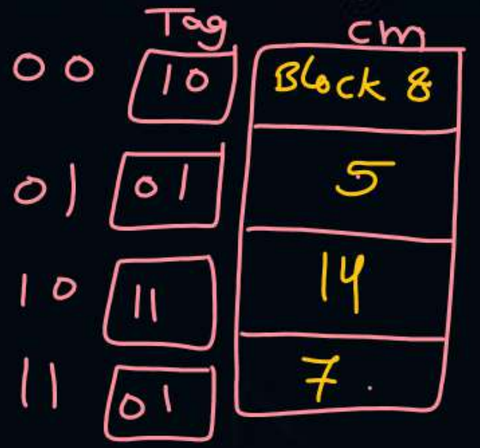


[MCQ]

no. of blocks in cm = $\frac{64B}{16B} = 4$

#Q. Consider a 64 bytes direct mapped cache with a block size of 16 bytes. Main memory size is 256bytes. Currently in the cache, the blocks are having tags as follows:

Block	Tag
00	10
01	01
10	11
11	01



mm block

$(1000)_2 = (8)_{10}$

$(0101)_2 = (5)_{10}$

$(1110)_2 = (14)_{10}$

$(0111)_2 = (7)_{10}$

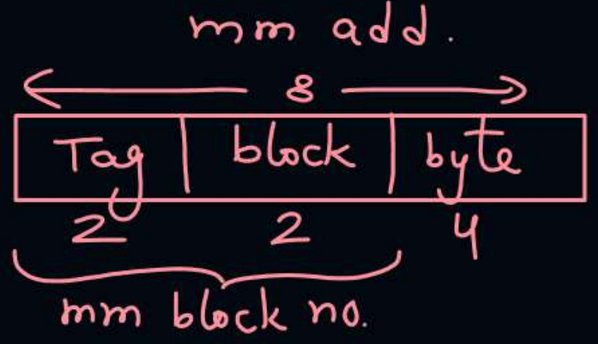
Identify the correct statement with respect to the availability of the main memory data into cache?

a) Main memory byte number 243 present in cache → miss

b) Main memory byte number 143 present in cache → Hit

c) Main memory byte number 43 present in cache → miss

d) Main memory byte number 119 present in cache → Hit



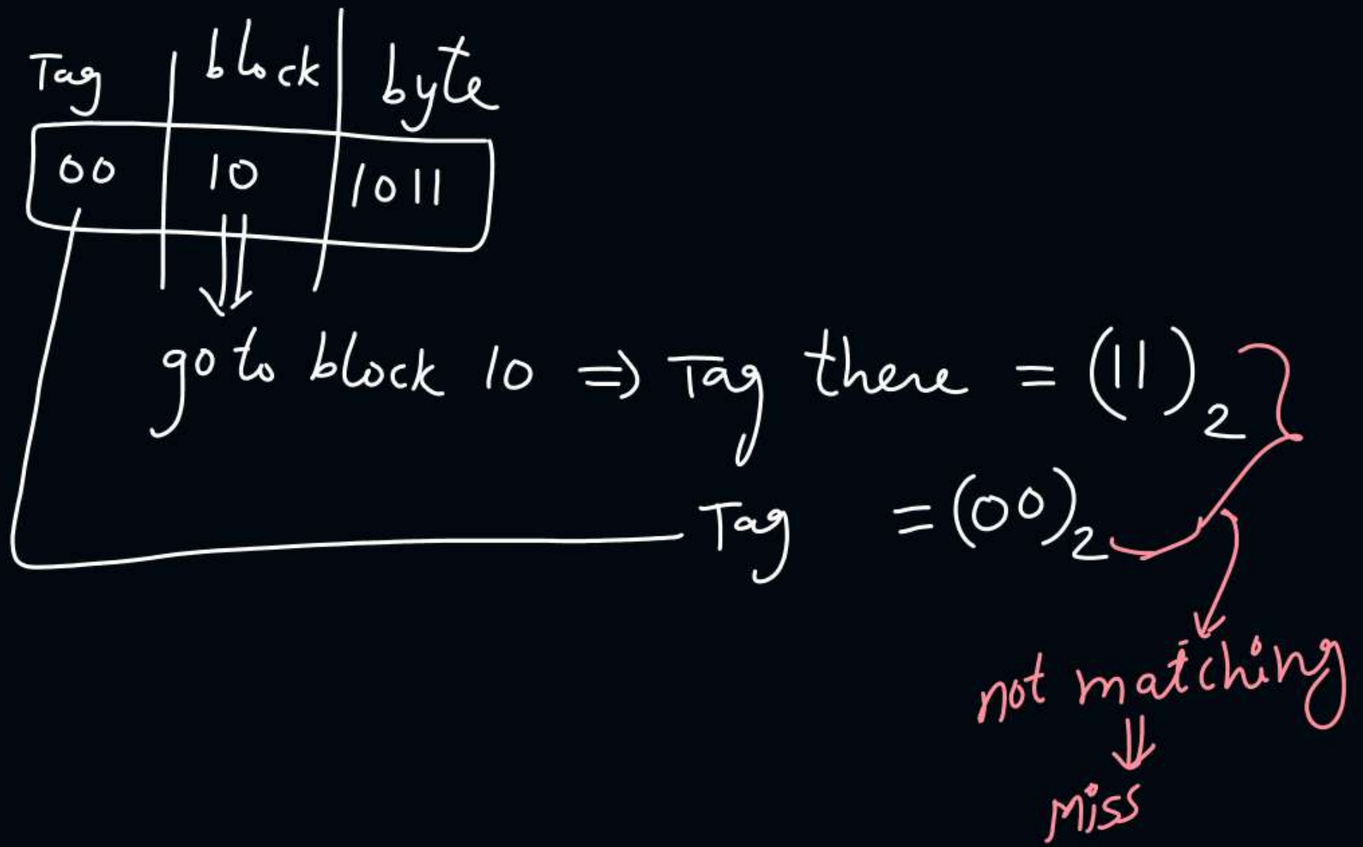
option c :-

$(43)_{10} = (00101011)_2$

⇓

mm block no. $= \left\lfloor \frac{43}{16} \right\rfloor$

$= (2)_{10}$



option (a) :-

$(243)_{10}$

mm block no. $= \left\lfloor \frac{243}{16} \right\rfloor = (15)_{10}$

---

option (b) :-

mm block no. $= \left\lfloor \frac{143}{16} \right\rfloor = (8)_{10}$

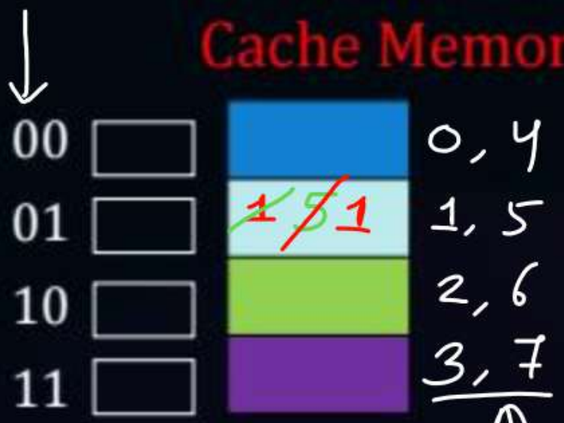
---

option d :-

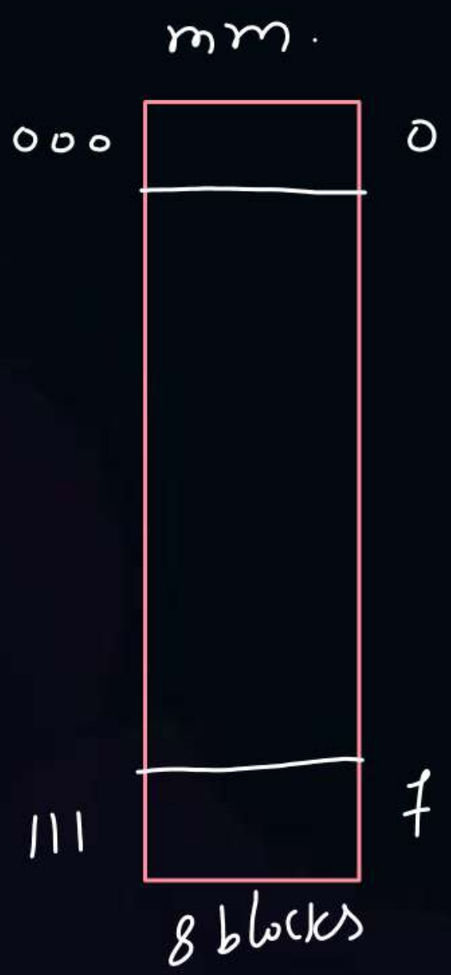
mm block no. $= \left\lfloor \frac{119}{16} \right\rfloor = (7)_{10}$

Topic : Problem With Direct Mapping

cm block no. (index)



mappings



CPU Requests mm blocks

1, 5, 1, 5, 1, 5, 1, 5

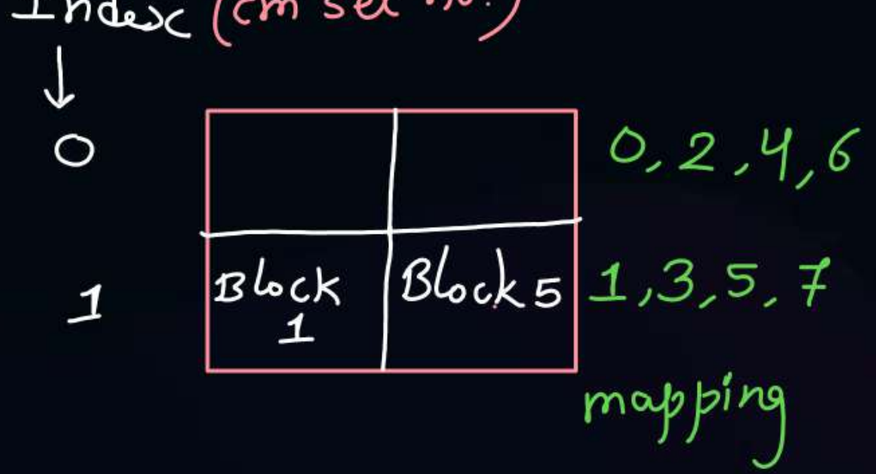
miss

miss

miss

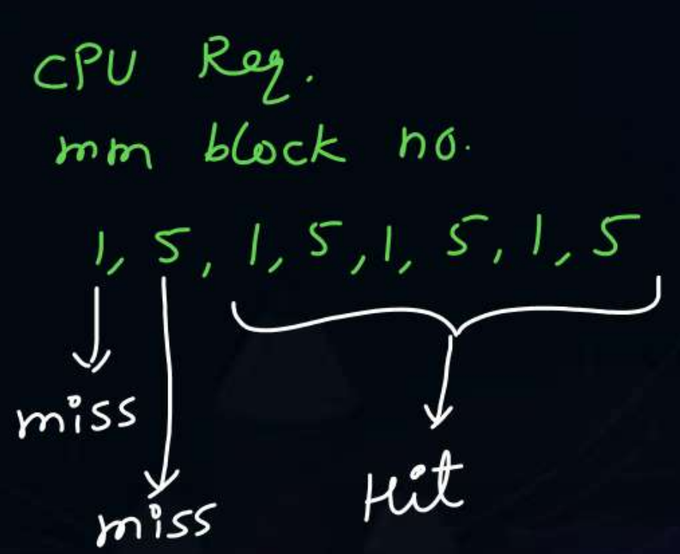
Topic : Set Associative Mapping

on each index in cache multiple blocks of main memory can be stored.



2-way set associative cache

(because on each index 2 blocks are organized)



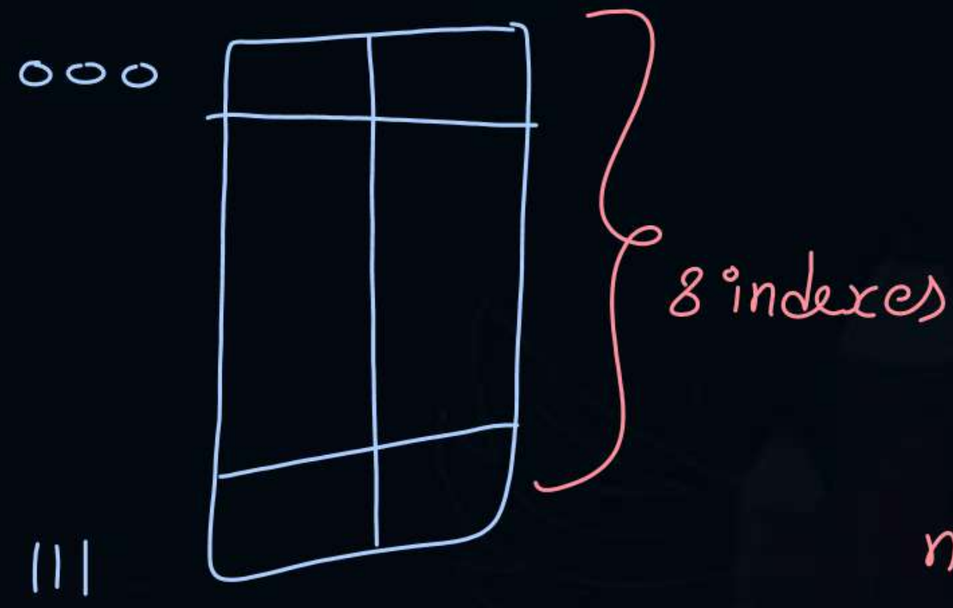
Topic : Set Associative Mapping

ex:-

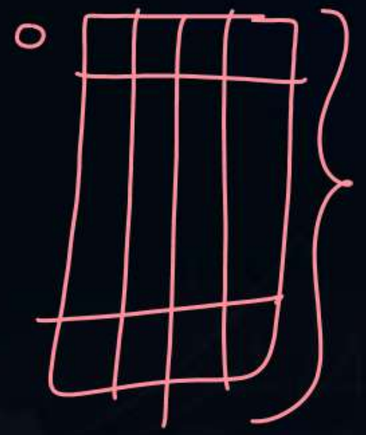
Direct mapped

16 blocks

2 - way set ass.



4 - way set ass.



no. of indexes = 4

no. of indexes in a k-way set associative cache (no. of sets) $= \frac{\text{no. of blocks in cache}}{k}$

---

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

---

mm add.

no. of bits in set no. $= \log_2$ (no. of sets in cache)

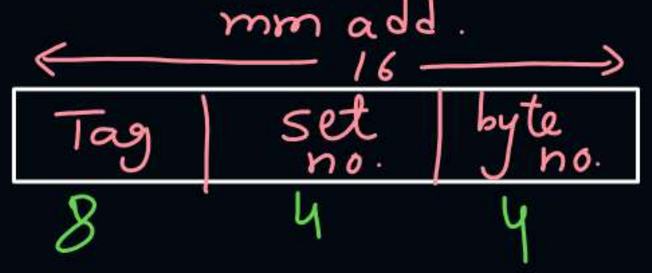
Tag directory size = no. of blocks in cache * (Tag + extra bits)

ex:- mm add. = 16 bits

cm size = 512 B

block size = 16 B = 2^4 B ⇒ byte no. = 4 bits

2-way set associative cache



no. of blocks in cache = $\frac{512\ B}{16\ B} = \frac{2^9}{2^4} = 2^5 = 32$

no. of sets in cache = $\frac{2^5}{2} = 2^4 = 16$

⇓

no. of bits for set offset = 4 bits

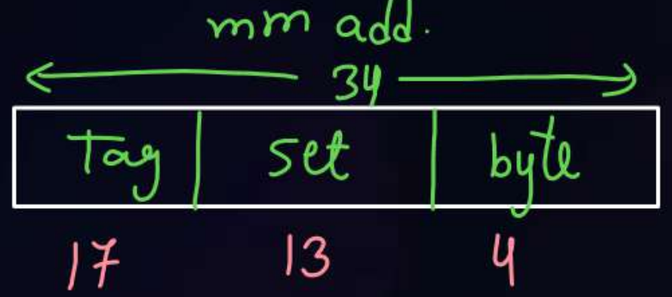
Tag directory size = $2^5 * 8$ bits = 256 bits

[NAT]

#Q. A computer has a 512Kbyte, 4-way set associative, write back data cache with block size of 16 Bytes. The processor sends 34 bit addresses to the cache controller. Each cache tag directory entry contains, in addition to address tag, 2 valid bits, 1 modified bit and 1 replacement bit

2^4 B

1. The number of bits in the tag field of an address is → 17 bits
2. The size of the cache tag directory is → $2^{15} * (17+2+1+1) = 2^{15} * 21$ bits



no. of blocks in cache $= \frac{512\,kB}{16\,B} = \frac{2^9 \cdot 2^{10}}{2^4} = 2^{15}$

no. of sets in cache $= \frac{2^{15}}{4} = \frac{2^{15}}{2^2} = 2^{13} \Rightarrow$ set no. = 13 bits

2 mins Summary

Topic Direct Mapping

Topic Set Associative Mapping

Topic Fully Associative Mapping

Happy Learning

THANK - YOU