

COMPUTER SCIENCE

Computer Organization and Architecture

Cache Memory

Mapping Techniques-3

Lecture_06



Vijay Agarwal sir



A graphic of a construction barrier made of two orange and white striped panels with black bases and yellow top caps, positioned to the left of the main title.

**TOPICS
TO BE
COVERED**

01

Mapping Techniques

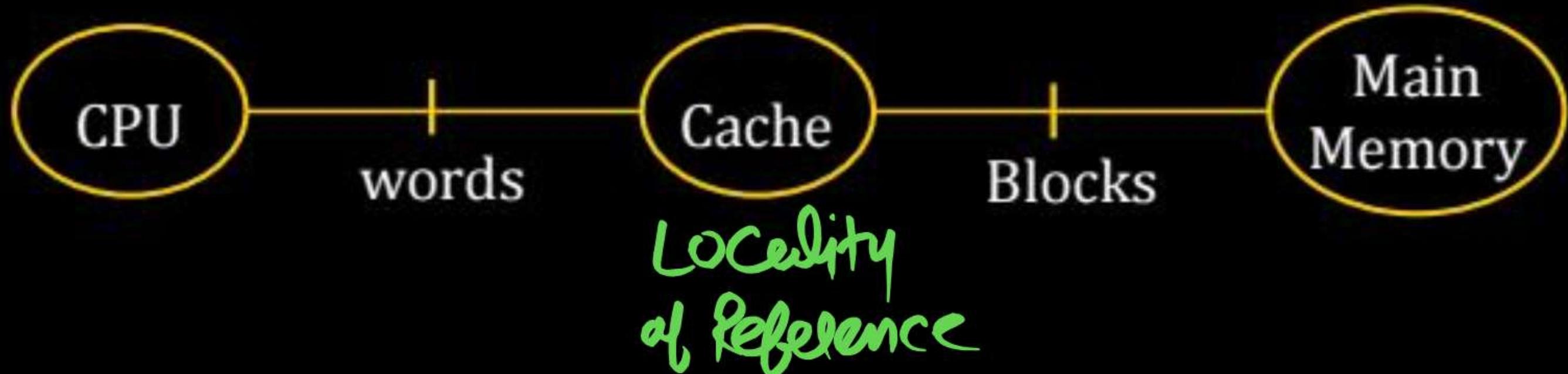
Cache org
Main Memory org.

Mapping Technique

MM → CM.

- ① Direct Mapping
- ② Set Associative
- ③ Fully Associative
Mapping.

Memory Organization



Mapping

The process of transfer the Data from Main Memory to Cache

Memory is called mapping. There are 3 Type of Mapping

Technique

1) Direct Mapping

2) Set Associative Mapping

3) Fully Associative Mapping

Mapping Function

- ❑ Because there are fewer cache lines than main memory blocks, an algorithm, is needed for mapping main memory blocks into cache lines.
- ❑ Three techniques can be used:

Direct

- The simplest technique
- Maps each block of main memory into only one possible cache lines.

Associative

- permits each main memory block to be loaded into any line of the cache.
- The cache control logic interprets a memory address simply as a Tag a word field
- To determine whether a block is in the cache, the cache control logic must simultaneously examine every line's Tag for a match

Set Associative

- A compromise that exhibits the strength of both the direct and associative approaches while reducing their disadvantage.

Direct Mapping

In this Direct Cache Controller interprets the CPU generated Request as follows:



$$\# \text{LINE} = \frac{\text{CM Size}}{\text{BLOCK Size}}$$

$$\text{Word Offset} = \log_2(\text{Block Size})$$

$$\text{LINE Offset} = \log_2(\#\text{LINE})$$

$$\text{TAG} = \text{Physical Address} - (\text{Line offset} + \text{Word offset})$$

$$\text{TAG Memory Size} = \#\text{LINE}'s \times \text{Tag bits} \text{ (Depend on the mapping technique)}$$

Q.

- Consider a Direct Mapping if the size of Cache memory is 512KB & Main Memory 512 KB & Cache line size (Block) is 64KB the calculate the number of bit required for
- (i) P.A
 - (ii) TAG
 - (iii) B.O
 - (iv) W.O
 - (v) #LINES
 - (vi) TAG Memory Size.

Q.

Consider a Direct Mapping, Cache size = 64 byte, Line Size = 8

P
W

Byte. MM = 256 Byte then #bits for P.A, TAG, L.O, W.O Tag
memory size?

Q.

Consider a Direct Mapping, Cache size = 128 KB, Line Size = 64

P
W

Byte. Main Memory is 1MB then what is the line number of physical address $(ABCDE)_{16}$?

Q.

P
W

Consider a machine with a byte addressable main memory of 2^{20} bytes, block size pf 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 $(E201F)_{16}$?

- (a) E, 201
- (b) F, 201
- (c) E, E20
- (d) 2, 01F

[GATE-2015]

Q.

Consider a machine with a byte addressable main memory of 2^{32} bytes divided into blocks of size 32 bytes. Assume that a direct mapped cache having 512 cache lines is used with this machine. The size of the tag field in bits is _____.

[GATE - 2017]

P
W

NAT

Consider a computer system with a byte-addressable primary memory of size 2^{32} bytes. Assume the computer system has a direct-mapped cache of size 32 KB ($1 \text{ KB} = 2^{10}$ bytes), and each cache block is of size 64 bytes.

The size of the tag field is _____ bits.

[GATE-2021(Set-1)-CS: 1M]

1) Direct Mapping

In this Technique mapping function is used to transfer the data from Main Memory to Cache Memory. The Mapping Function is

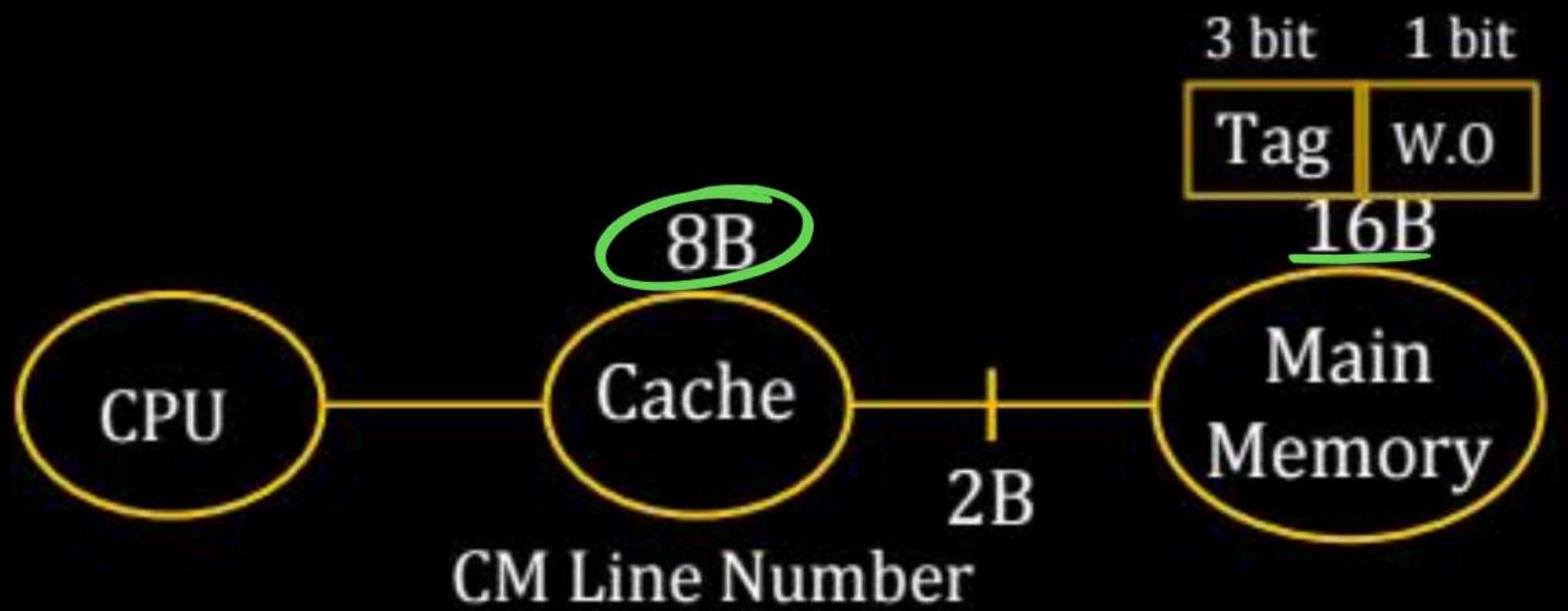
$$\text{Cache address} = \text{Main Memory request} \quad \text{MOD} \quad \# \text{ CM LINES}$$

(Or)

$$K \text{ MOD } N = i$$

K: MM Block No.

- N: # of Cache Line
- i: CM Line Number



$$0 \bmod 4 = 0$$

$$1 \bmod 4 = 1$$

$$2 \bmod 4 = 2$$

$$3 \bmod 4 = 3$$

$$4 \bmod 4 = 0$$

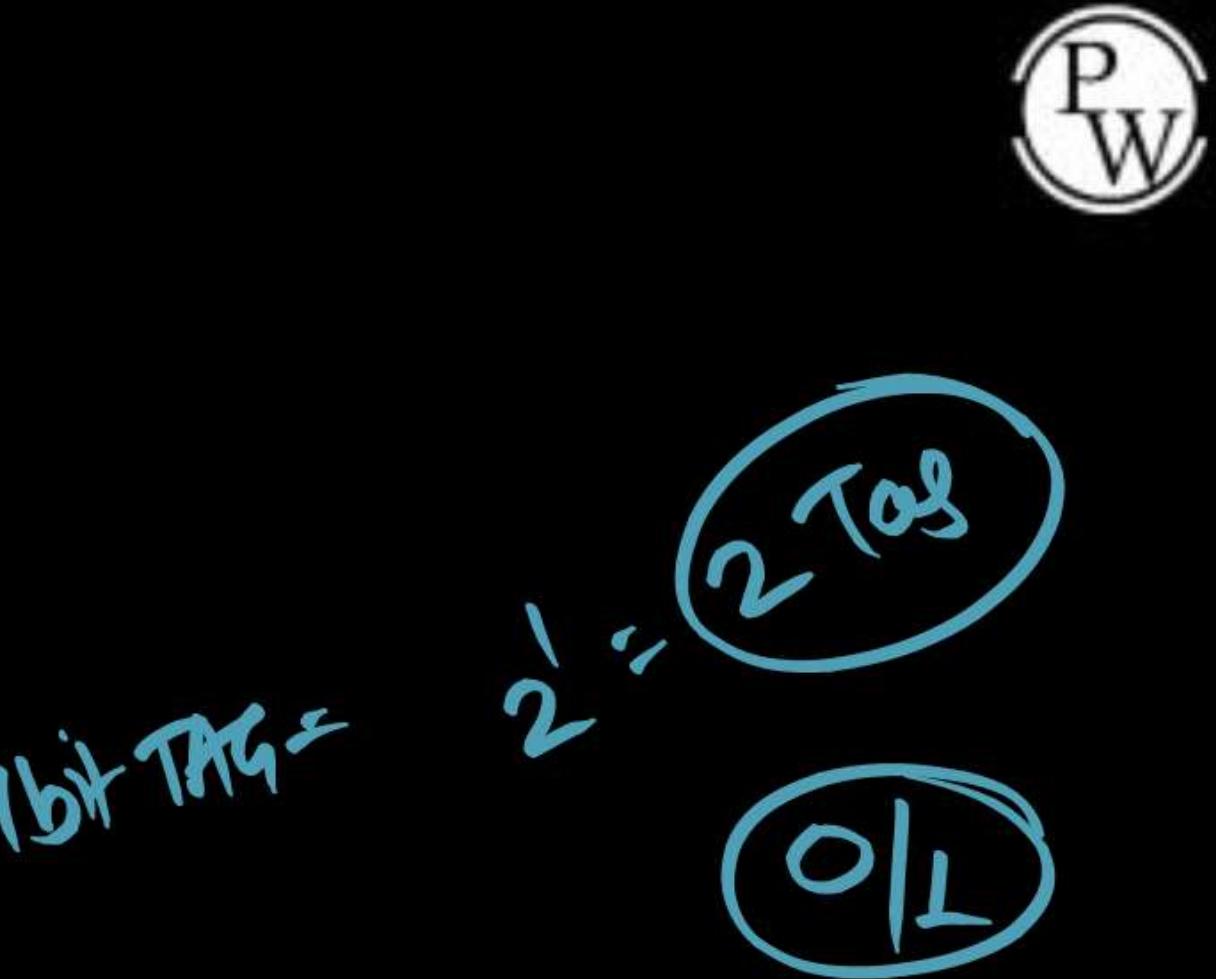
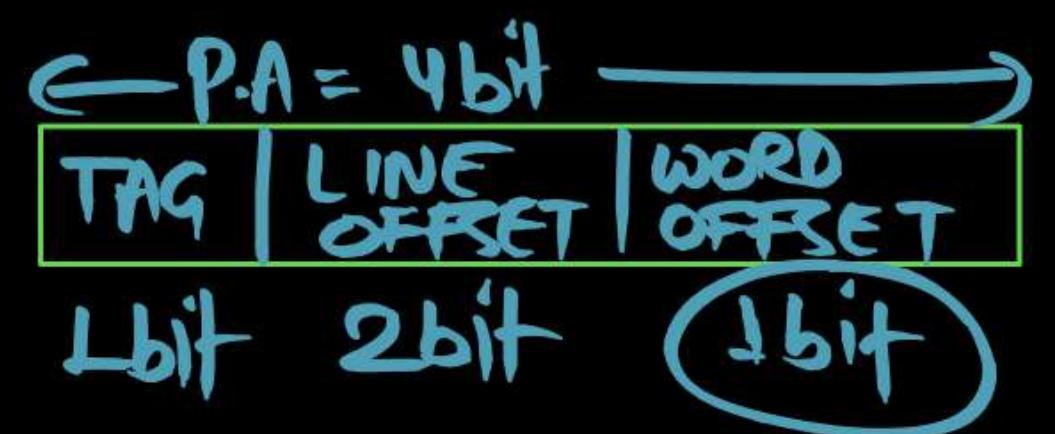
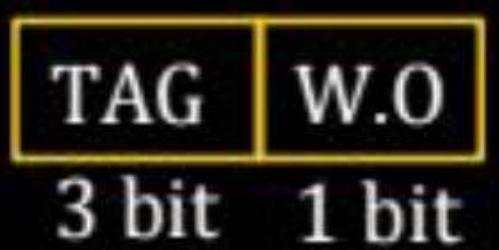
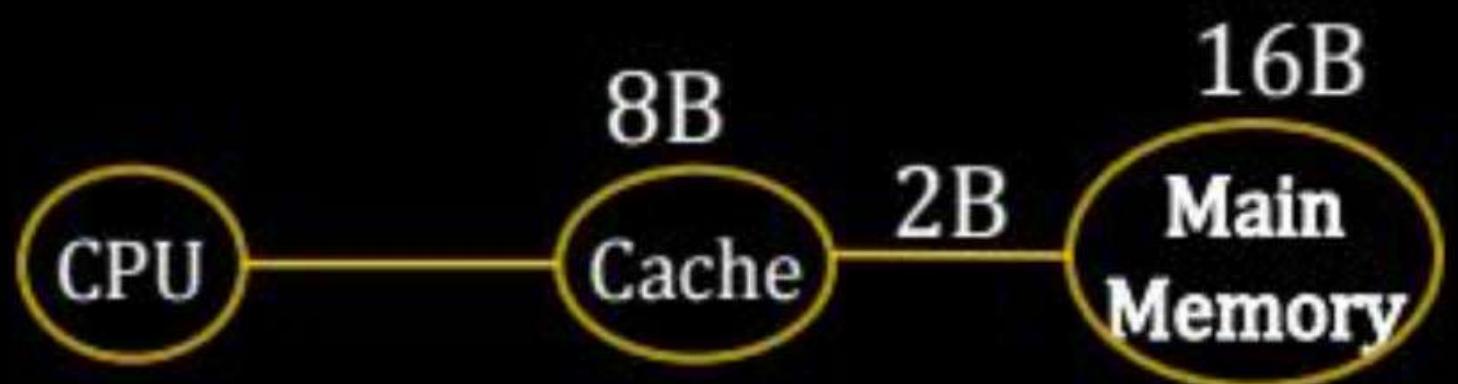
$$5 \bmod 4 = 1$$

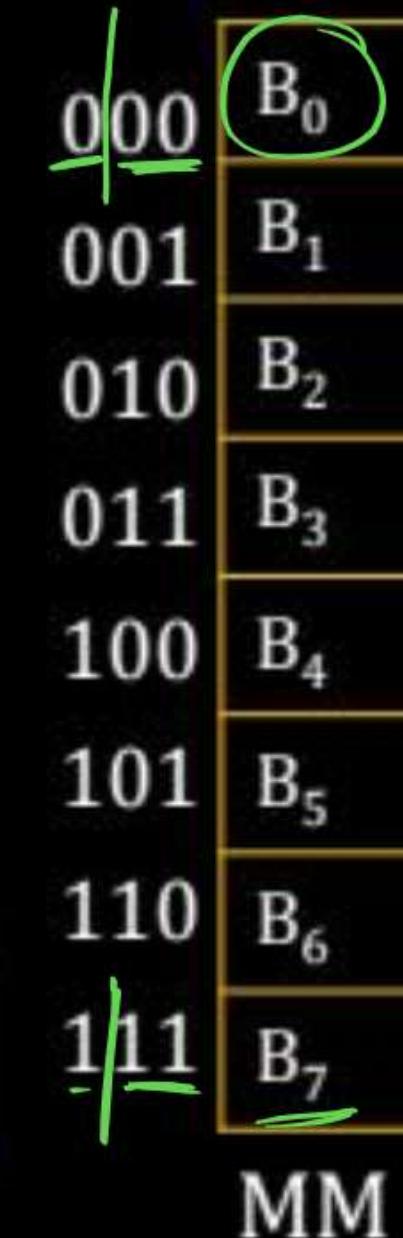
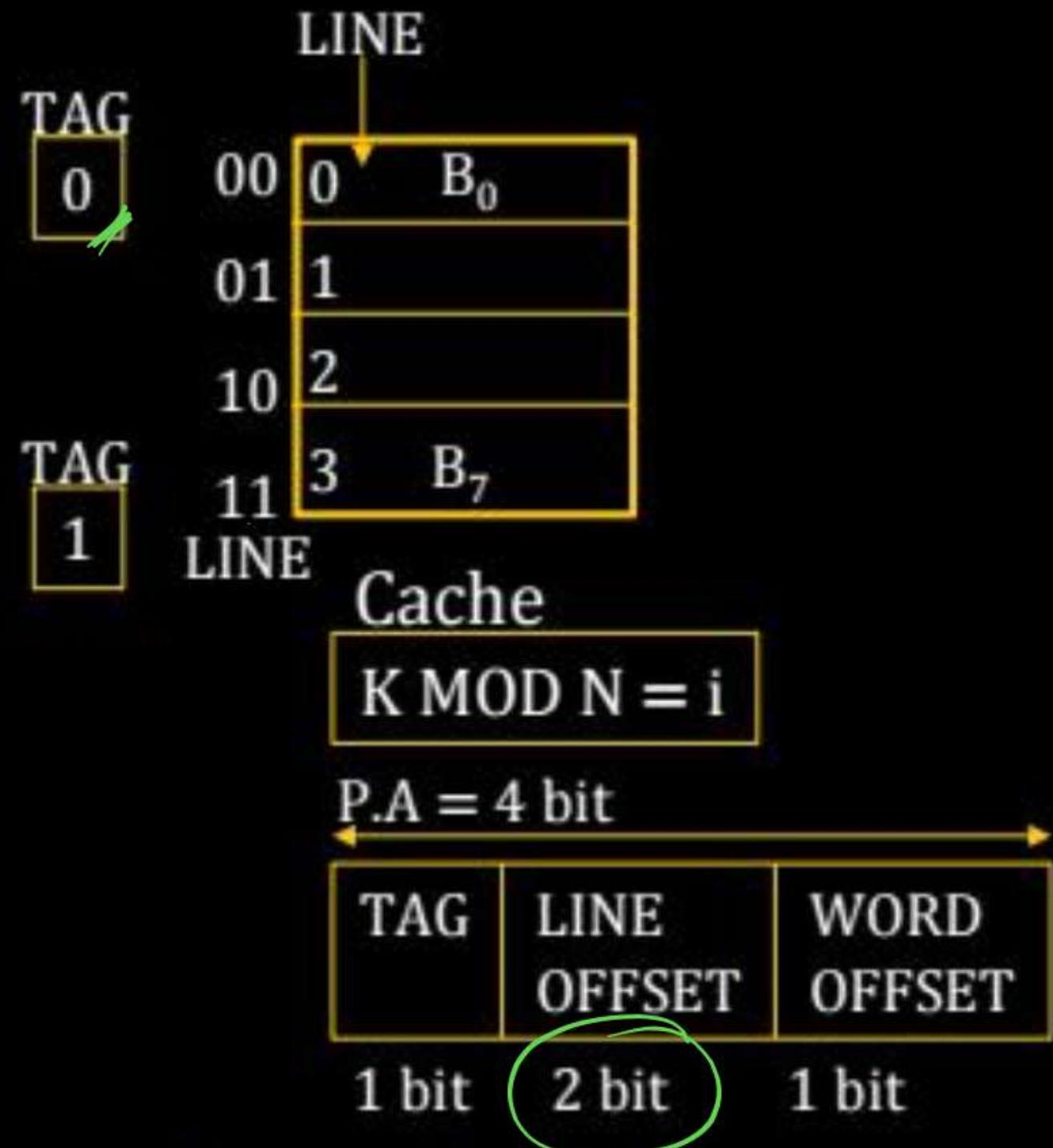
$$6 \bmod 4 = 2$$

$$7 \bmod 4 = 3$$

CM

 $B_0 \& B_4$: LINE 0 $B_4 \& B_5$: LINE 1 $B_2 \& B_6$: LINE 2 $B_3 \& B_7$: LINE 3





$0 \bmod 4 = \text{LINE } '0'$
 $7 \bmod 4 = \text{LINE } '3'$

Direct Mapping

MM Block



Direct Mapping

$$\frac{K \bmod N = i}{0 \bmod 4 = '0'}$$

CM LINE

LINE '0'



$$\frac{K \bmod N = i}{7 \bmod 4 = '3'}$$

LINE '3'

$$\text{Tag Memory Size} = \# \text{LINE's} \times \text{Tag bits}$$

Depends
On the
Mapping
technique

In the above example: # LINE = 4
Tag bit = 1 bit
(Direct Mapping)

$$\text{Tag Memory Size} = 4 \times 1 = 4 \text{ bits}$$

$$\begin{aligned} & \# \text{LINE} \times \text{Tag bits} \\ & 2^2 \times 1 \text{ bit} \\ & 4 \times 1 = \text{4bit Tag Memory} \end{aligned}$$

How the Cache Hit & Cache Miss Identify?

Consider the following program

I₁: MOV r₀ [0000]

I₂: MOV r₁ [1000]

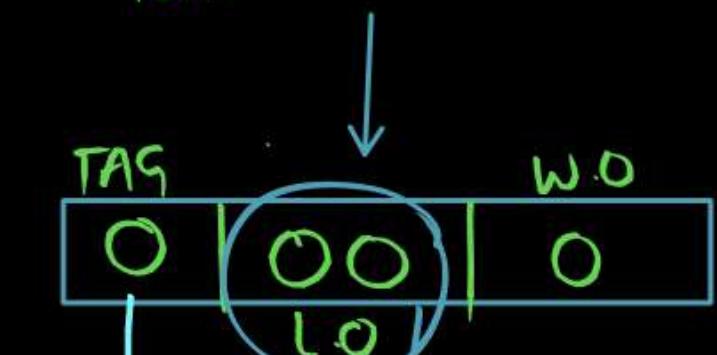
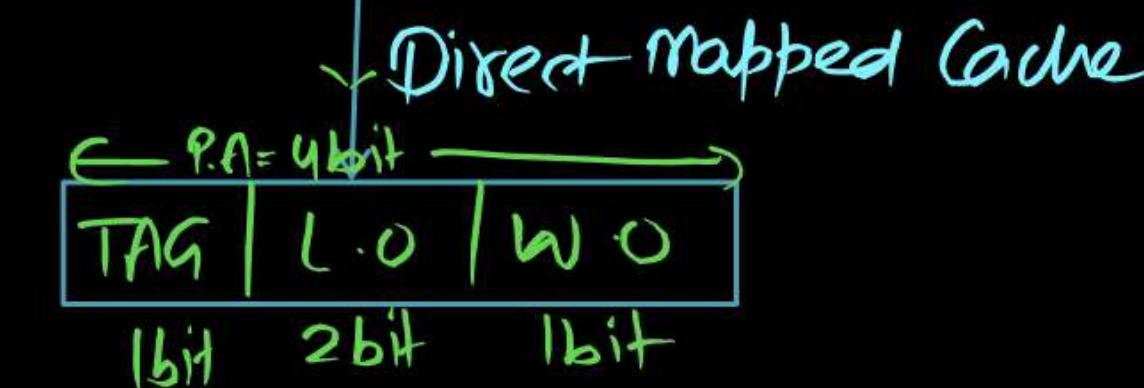
I₃: ADD r₀r₁

CPU generate the Mem Req.
Cache

[0000] RP → Memory Read

T₁: 0000

0000 Cache



Consider the following program

I₁: MOV r₀ [0000]

I₂: MOV r₁ [1000]

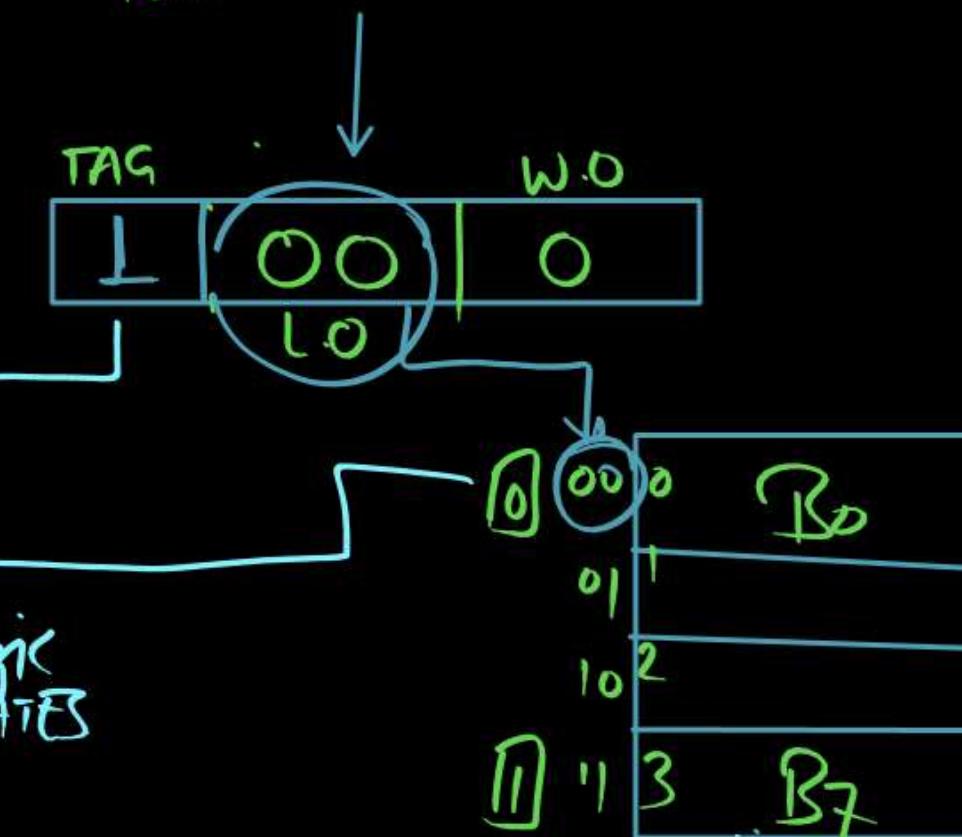
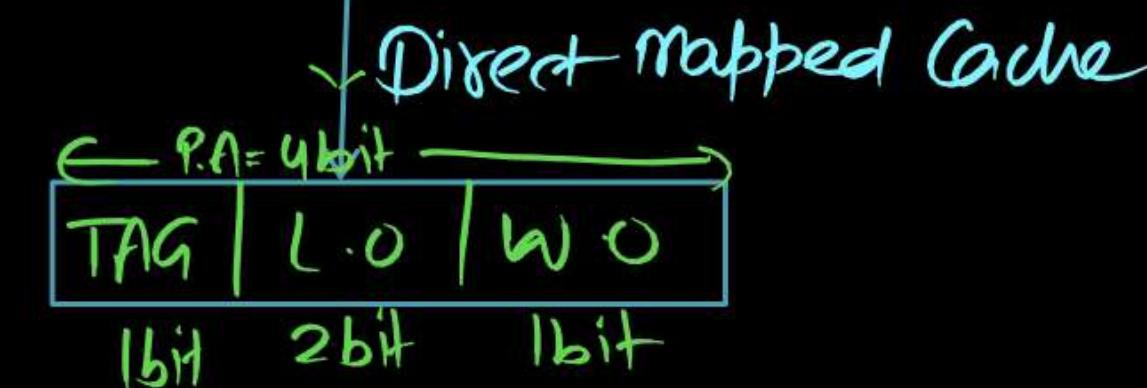
I₃: ADD r₀r₁

CPU generate the Mem Req.
Cache.

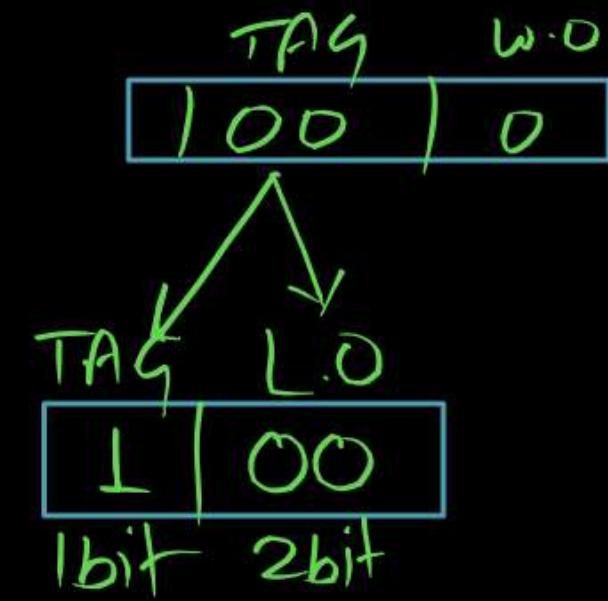
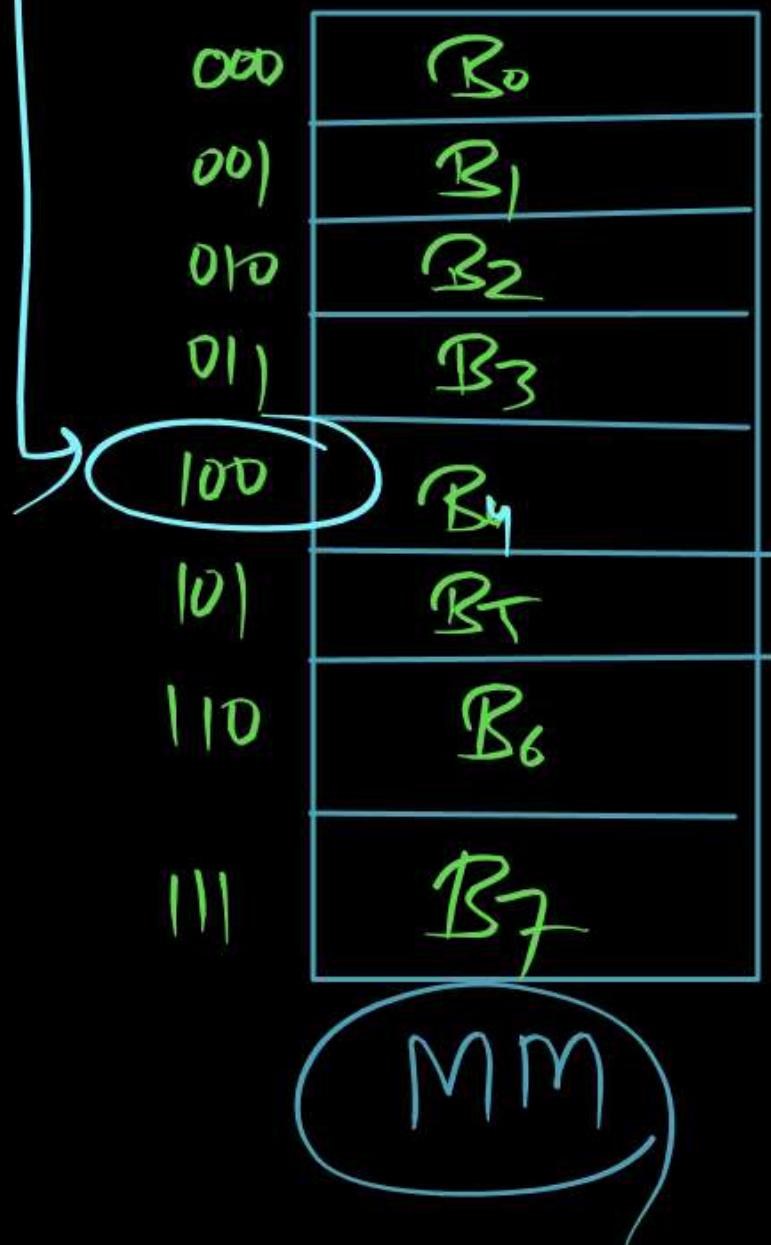
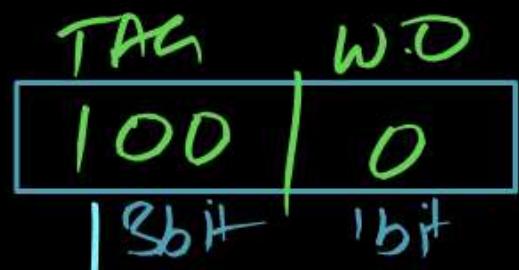
[0000] RD → Memory Read

T₂: [1000]

[1000] Cache



P
W



$$\frac{K \bmod N = i}{4 \bmod 4 = 0} \rightarrow \text{LINE } '0'$$

Direct Mapping
MM to CM



LOR [Locality of Reference]

Consider the following program

$I_1: \text{MOV } r_0 [000|0] \Rightarrow 0^{\text{th}} \text{ Byte of Block } B_0$

$I_2: \text{MOV } r_1 [000|1] \quad 1^{\text{st}} \text{ Byte of Block } B_0$

$I_3: \text{MOV } r_2 [100|0] \quad 0^{\text{th}} \text{ Byte of Block } B_1$

$I_4: \text{MOV } r_3 [100|1] \quad 1^{\text{st}} \text{ Byte of Block } B_1$

$T_1: \underline{\text{Cache Hit}} \quad (B_0 \text{ is in Cache}) \quad 0^{\text{th}} \text{ Byte of Block } 0 \quad (B_0) \text{ given to CPU}$

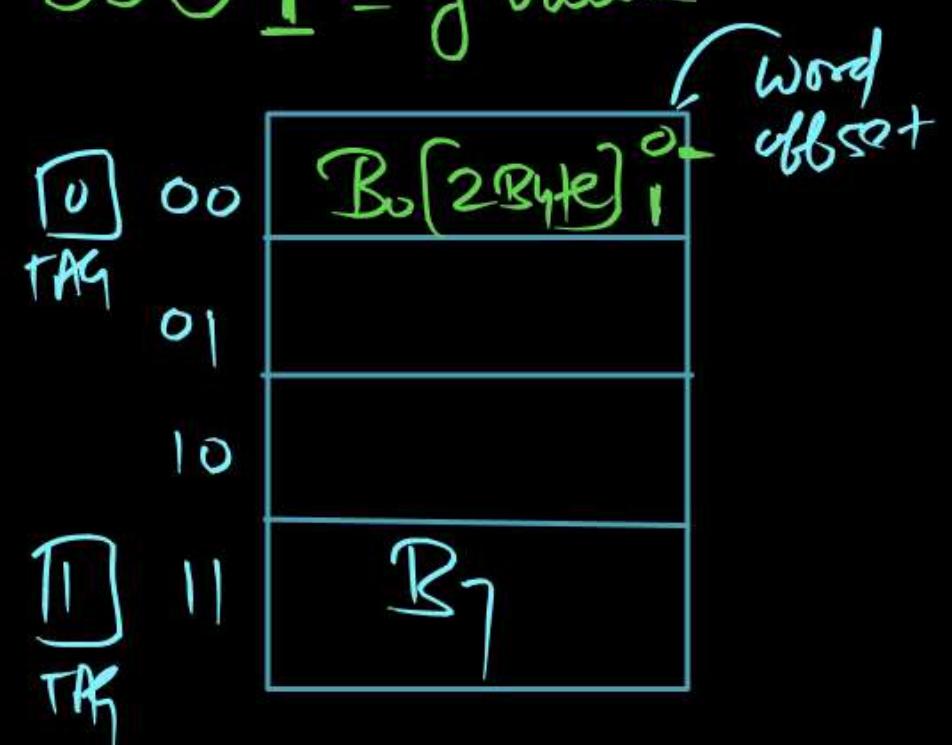
$\rightarrow I_2: \underline{\text{Cache Hit}} \quad (B_0 \text{ is in Cache}) \quad 1^{\text{st}} \text{ Byte of Block } 0 \text{ given to CPU}$

$T_3: 1000 \Rightarrow 0^{\text{th}} \text{ Byte of } B_1 \quad (\text{Cache Miss.})$

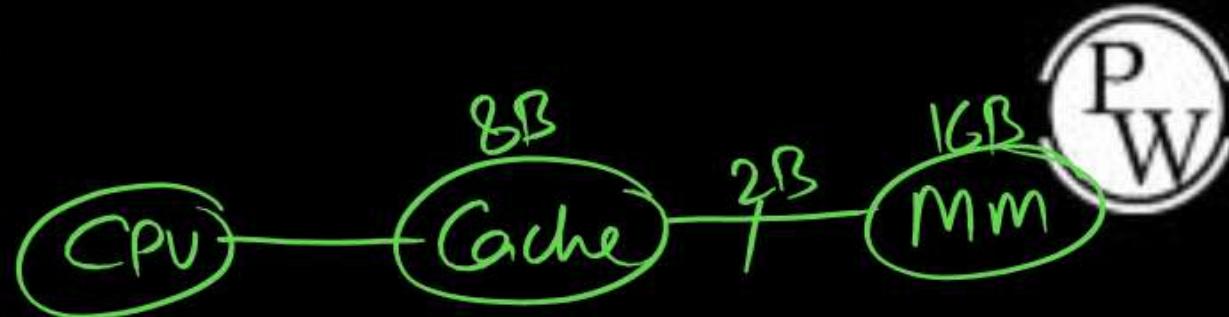


$0000\ \underline{0} = x \text{ value}$

$0000\ \underline{1} = y \text{ value}$



LOR [Locality of Reference]



Consider the following program

$I_1: \text{MOV } r_0 [000|0] \Rightarrow 0^{\text{th}} \text{ Byte of Block } B_0$

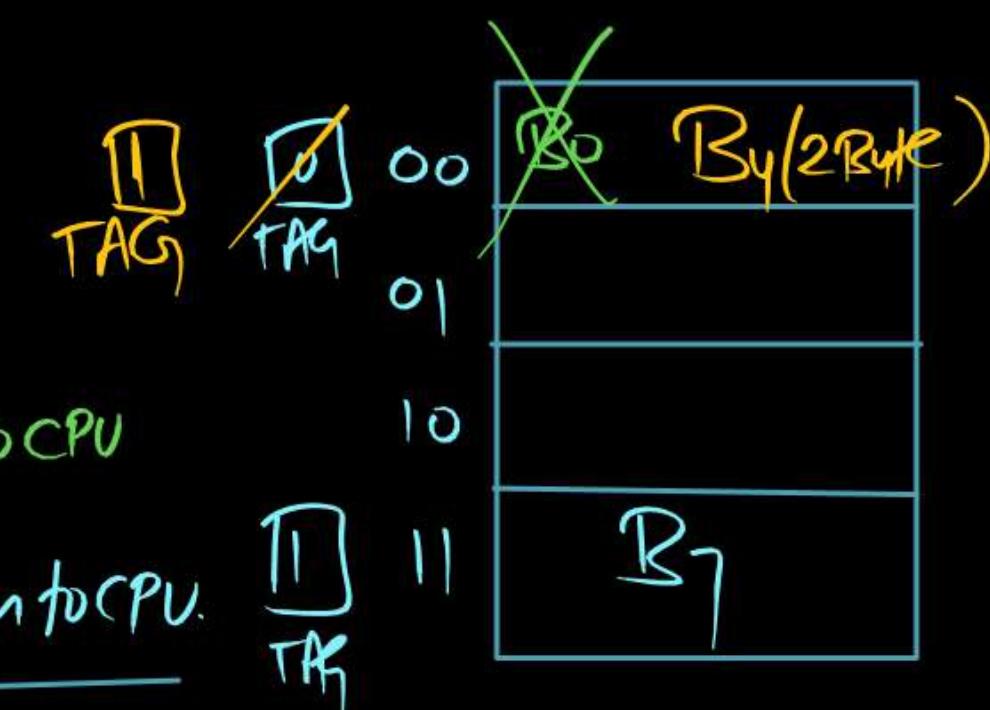
$I_2: \text{MOV } r_1 [000|1] \quad 1^{\text{st}} \text{ Byte of Block } B_0$

$I_3: \text{MOV } r_2 [100|0] \quad 0^{\text{th}} \text{ Byte of Block } B_1$

$I_4: \text{MOV } r_3 [100|1] \quad 1^{\text{st}} \text{ Byte of Block } B_1$

$I_1: \underline{\text{Cache Hit}} \quad (B_0 \text{ is in Cache}) \quad 0^{\text{th}} \text{ Byte of Block } 0 (B_0) \text{ given to CPU}$

$\rightarrow I_2: \underline{\text{Cache Hit}} \quad (B_0 \text{ is in Cache}) \quad 1^{\text{st}} \text{ Byte of Block } 0 \text{ given to CPU.}$



$I_3: 1000 \Rightarrow 0^{\text{th}} \text{ Byte of } B_1 \quad (\text{Cache Miss.})$

$I_4: 1001 \Rightarrow 1^{\text{st}} \text{ Byte of } B_1 \quad (\text{Cache HIT}) \quad \text{so } B_1 \text{ is given to CPU by Cache.}$

Limitation of Direct Mapping

- In Direct Mapping each Cache Line is able to Hold Only One Tag (One mm Block) at a time. So Number of Conflict Miss Increase.
- If CPU is Referencing the Multiple Blocks which are Mapped to the Same Cache Line (Assume All Blocks are Put into Cache LINE Number n) then Number of Conflict Miss very High, even other Cache Blocks are empty. (Like Thashing)

Disadvantage of Direct Mapping

I₁: MOV r₀ [0000] → 0th Byte of B₀

I₂: MOV r₁ [1000] → 0th Byte of B₄

I₃: MOV r₂ [0001] → 1st Byte of B₀

I₄: MOV r₃ [1001] → 1st Byte of B₄

I₅: MOV r₄ [0000] → 0th Byte of B₀

I₆: MOV r₅ [1000] → 0th Byte of B₄

Disadvantage of Direct Mapping

I₁: MOV r₀ [0000] → B₀

I₂: MOV r₁ [1000] → B₄

I₃: MOV r₂ [0001] → B₀

I₄: MOV r₃ [1001] → B₄

I₅: MOV r₄ [0000] → B₀

I₆: MOV r₅ [1000] → B₄

Execution:

T₁: B₀ Cache Hit

T₂: B₄ Cache Miss, so Block Replacement TAG

T₃: B₀ Cache miss, so replace

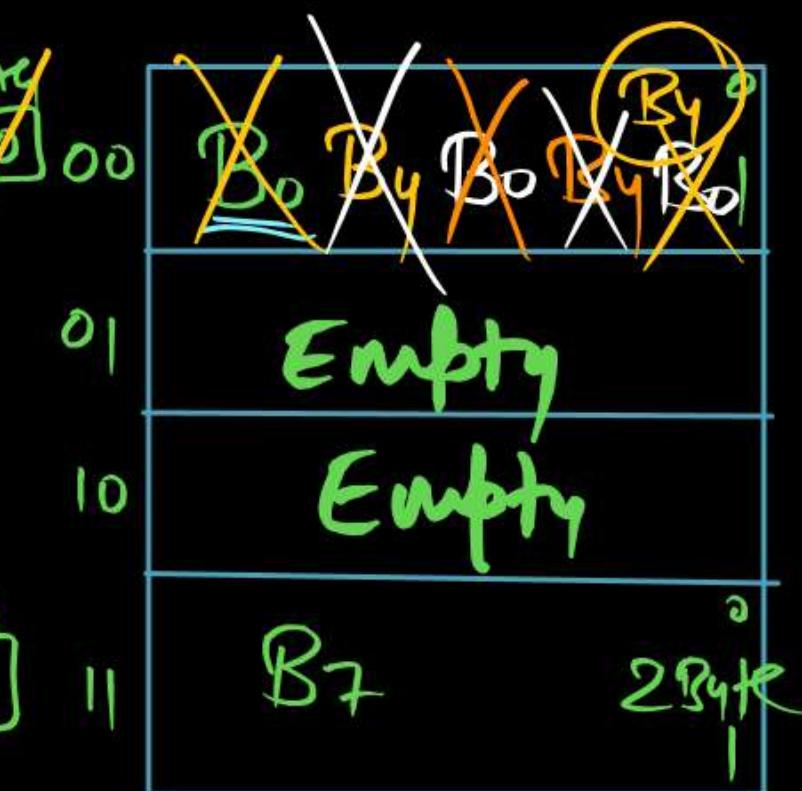
T₄: B₄ Cache miss

T₅: B₀ Cache miss

T₆: B₄ Cache miss

(5 Miss (out of 6 Access))

So Solution is Set Associative Mapping.



② Set Associative Mapping :



$$TAG = PA - [S.O + W.O]$$

$$\#LINES = \frac{C.M\ Size}{\text{Block Size}}$$

$$\#SET = \frac{\#LINE}{N\text{-way}}$$

$$\text{Set offset} [S.O] = \log_2 \#SET$$

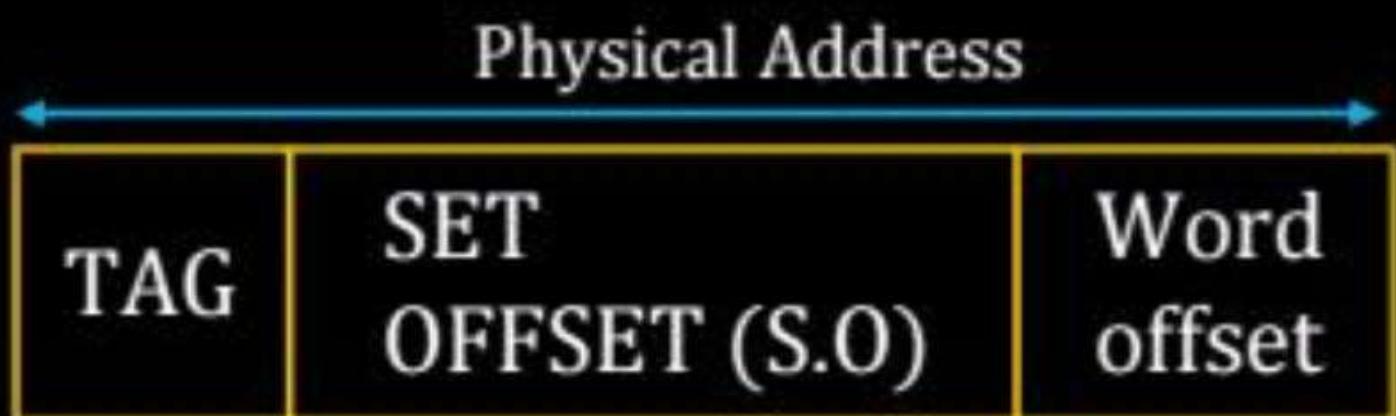
$$\text{Tag Memory} = \#LINES \times \text{Tag bits}$$

OR

$$\text{Tag memory} = \#SET \times \frac{\text{Block size}}{\text{Block per set}} \times \text{Tag bits}$$

2) Set Associative Cache

SET associative cache controller, Interpreter the CPU generated request as follows:



$$\text{Word Offset} = \log_2 \text{Block Size}$$

$$\# \text{lines} = \frac{\text{CM Size}}{\text{Block Size}}$$

$$\# \text{SETS} = \frac{\# \text{Lines}}{\text{N-way}}$$

$$\text{SET OFFSET} = \log_2 \# \text{SETS}$$

$$\text{TAG} = \text{Physical address} - (\text{S.O} + \text{W.O})$$

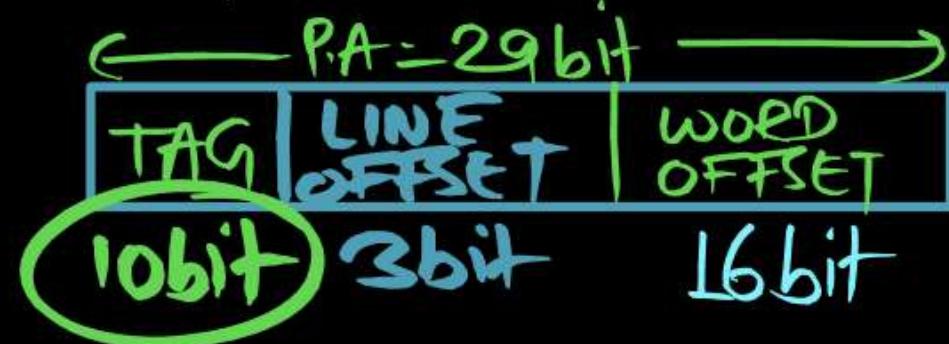
Q.1

Consider a Direct Mapped Cache if the size of cache memory is 512KB & Main Memory 512MB & Cache line size is 64KB then calculate the Number of bit Required for

19

1. (i) P.A (29) (ii) TAG (10) (iii) L.O (3) (iv) W.O (16)
 (2) #lines (8) (3) TAG memory size (80 bit) If Direct Mapping

Direct Mapped Cache



$$\begin{aligned} \text{Tag Memory Size} &= \# \text{LINES} \times \text{Tag bits} \\ &\Rightarrow 2^3 \times 10 \text{ bits} \\ &= 80 \text{ bits} \quad \text{Ans} \end{aligned}$$

$$\# \text{LINES} = \frac{\text{CM Size}}{\text{Block size}} = \frac{2^{19}}{2^{16}} = 2^3 = 8 \text{ Lines}$$

$$\begin{aligned} \text{LINE OFFSET} &= \log_2 \# \text{LINES} \\ &\Rightarrow \log_2 8 = 3 \text{ bit} \end{aligned}$$

$$\begin{aligned} \text{TAG} &= \text{PA} - (\text{L.O} + \text{W.O}) \\ &= 29 - (3 + 16) = 10 \text{ bit} \end{aligned}$$

Q.9 Consider a 2-way set associative if the size of cache memory is 512KB & Main Memory 512MB & Cache line size is 64KB then calculate the Number of bit Required for

19

(i) P.A (29)

(ii) TAG (11)

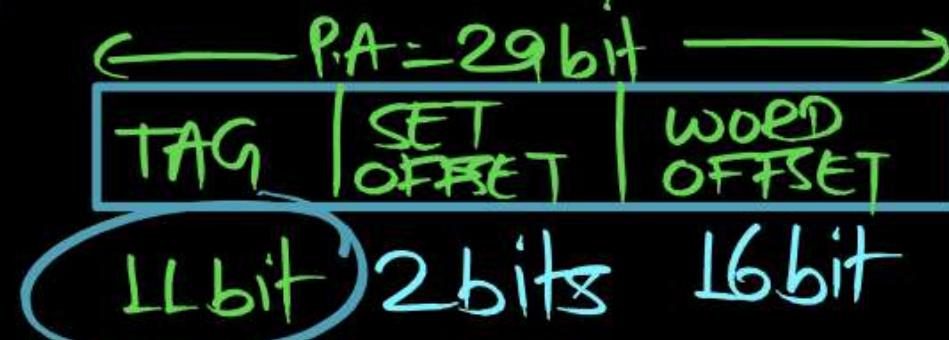
(iii) S.O = 2 bits

(iv) W.O (16)

(2) #lines (8)

(3) TAG memory size

16bit

#SET
(4 Set)& SO
2bit2 way Set Associative

$$\text{Tag Memory} = \# \text{LINES} \times \text{Tag bits}$$

$$\Rightarrow 8 \times 11$$

$$= 88 \text{ bits}$$

(2 way)
N Way
2 Block per set

$$\# \text{LINES} = \frac{\text{CM Size}}{\text{Block size}} = \frac{2^9}{2^6} = 2^3 = 8 \text{ Lines}$$

$$\# \text{SET} = \frac{\# \text{LINE}}{\text{N-Ways}} = \frac{8}{2} = 4 \text{ Sets}$$

$$\text{Set offset} : \log_2 \# \text{SET} \Rightarrow \log_2 4 = 2 \text{ bits}$$

$$\text{TAG} = \text{PA} - (\text{S.O} + \text{W.O}) = 29 - 18 = 11 \text{ bits}$$

$$\text{Tag Memory} = \text{\# LINES} \times \text{Tag bits}$$
$$= 8 \times \text{LL bits}$$

#SET X Block per set

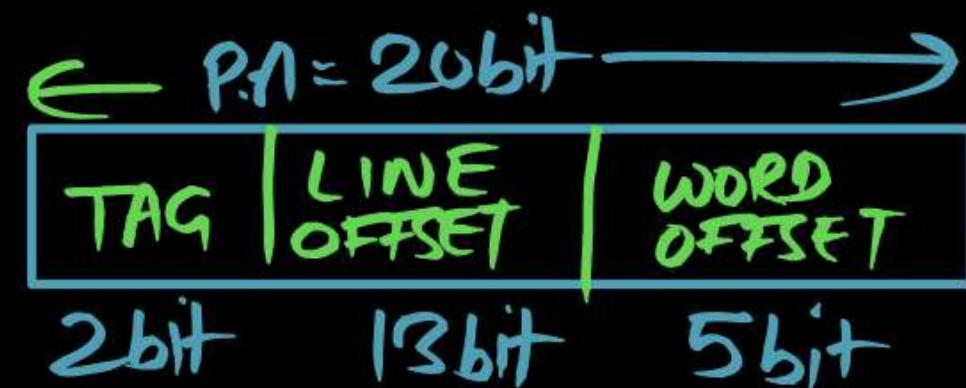
$$\boxed{\text{Tag memory size} = 88 \text{ bits}}$$

(OR)

$$\boxed{\text{Tag memory size} = \text{\#SETS} \times \frac{\text{Block bits}}{\text{Set}} \times \text{Tag bits}}$$
$$= 4 \times 2 \times \text{LL}$$
$$= 88 \text{ bits } \underline{\text{Ans}}$$

Q. Consider a **Direct Mapped** Cache Size = 256 KB, Line size = $\frac{P}{W}$
 32 Byte, MM = 1 MB, then what is the set number of Physical address (ABCDE)₁₆? $\leftarrow 20\text{bit}$

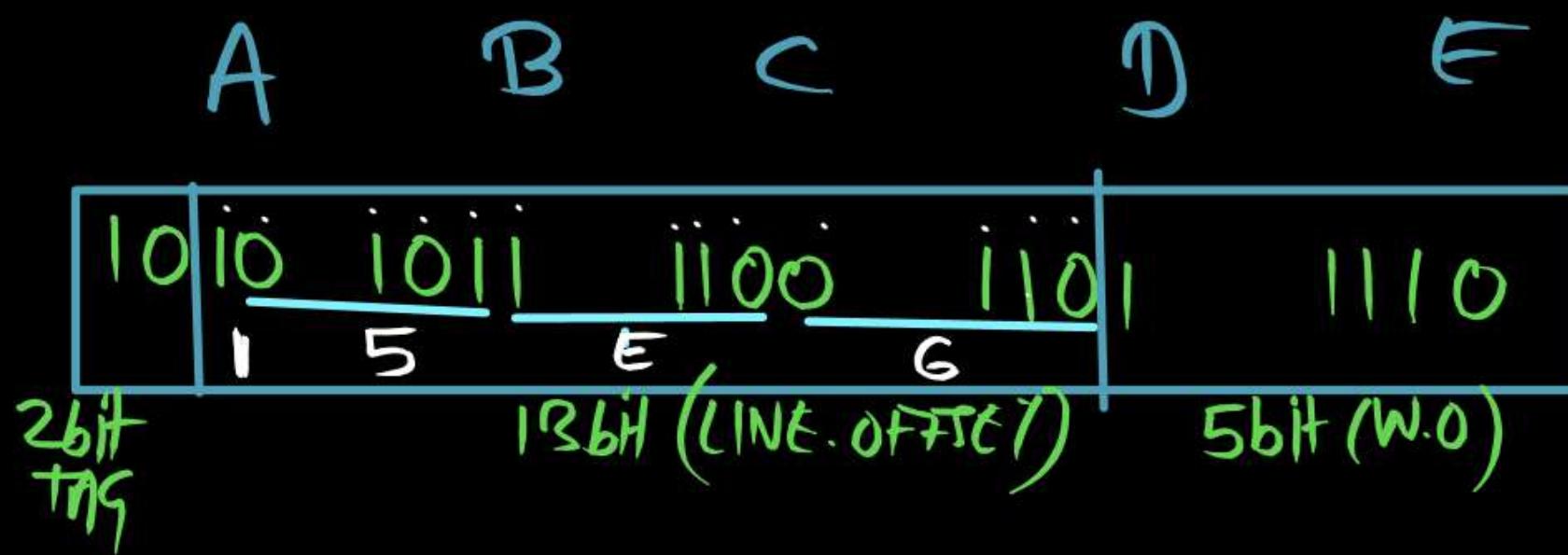
(S.M) **Direct Mapped Cache**



$$\# \text{LINES} = \frac{2^{\text{16}}}{{25\text{B}}}$$

$$(L5E6)_{16} \underset{\text{Ans}}{=} 2^{13}$$

$$L0 = 13\text{bit}$$



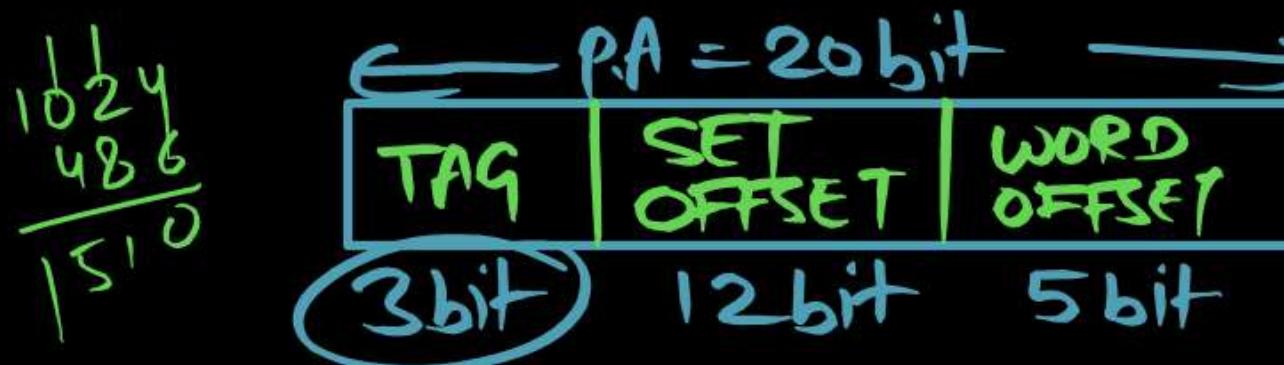
A: 10 : 1010
 B: 11 : 1011
 C: 12
 D: 13
 E: 14
 F: 15

Q.

Consider a 2-way set associative Cache Size = 256 KB, Line size = $\frac{P}{W}$
 32 Byte, MM = 1MB, then what is the set number of Physical
 address (ABCDE)₁₆?

S

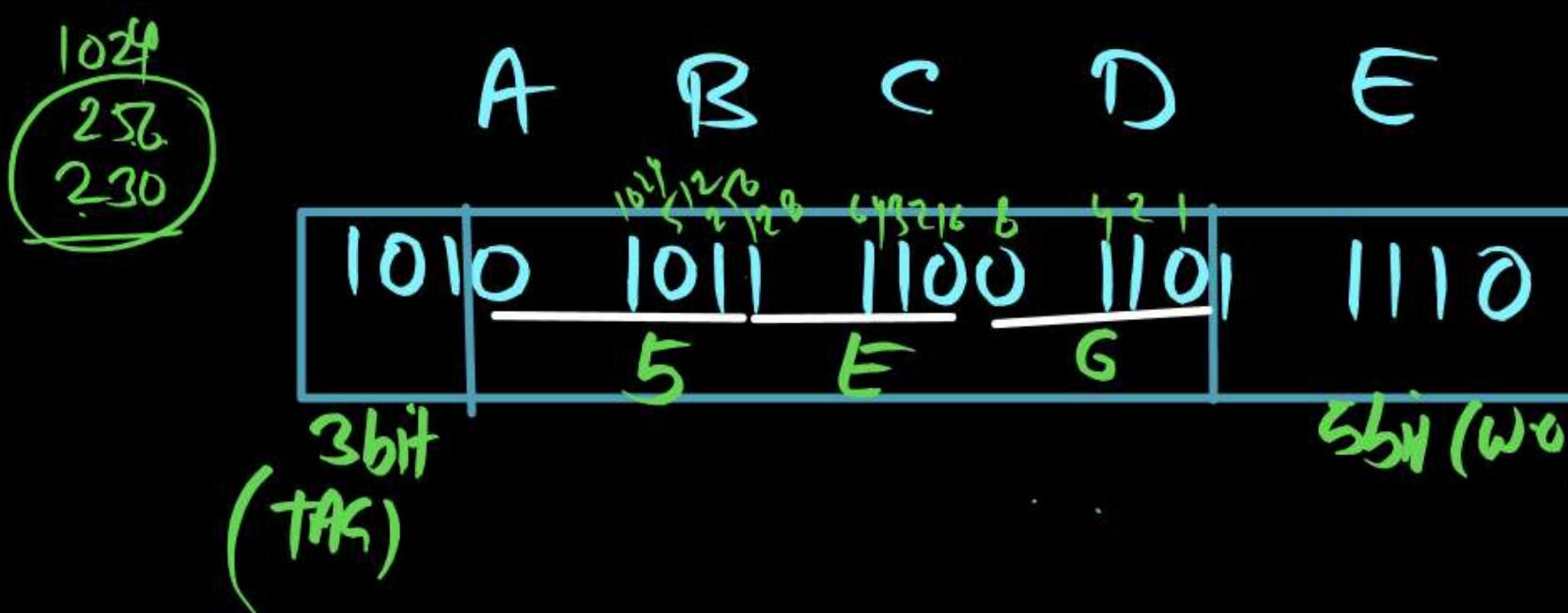
2 Way Set Associative



$$\# \text{LINE} = \frac{\text{CacheSize}}{\text{Blocksize}} = \frac{2^{18}}{2^5} = 2^{13}$$

$$\# \text{SET} = \frac{\# \text{LINES}}{N\text{-way}} = \frac{2^{13}}{2^1} = 2^{12}$$

$$S.O = 12 \text{ bit}$$



$(5E6)_{16}$ Ans

1510 in Decimal

Q.

The main memory of a computer has 2^m blocks while the cache has 2^c blocks. If the cache uses the set associative mapping scheme with 2^s blocks per set, then block k of the main memory maps to the set.

[GATE - 1999]

- (a) $(k \bmod m)$ of the cache
- (b) $(k \bmod c)$ of the cache
- (c) $(k \bmod 2^s)$ of the cache
- (d) $(k \bmod 2^{cm})$ of the cache

Q.

P
W

Consider a 4-way set associative cache consisting of 128 lines with a line size of 64 words. The CPU generates a 20-bit address of a word in main memory. The number of bits in the TAG, SET and WORD fields are respectively.

- (a) 9, 6, 5
- (b) 7, 7, 6
- (c) 7, 5, 8
- (d) 9, 5, 6

[GATE - 2007]

Q.

[Common Data for this and next question]

P
W

Consider a computer with a 4-way set-associative mapped cache of the following characteristics; a total of 1 MB of main memory, a word size of 1 Byte; a block size of 128 words and a cache size of 8 KB.

The number of bits in the TAG, SET and WORD fields, respectively are:

- (a) 7, 6, 7
- (b) 8, 5, 7
- (c) 8, 6, 6
- (d) 9, 4, 7

Q.

[Common Data]

P
W

Consider a computer with a 4-way set-associative mapped cache of the following characteristics; a total of 1 MB of main memory, a word size of 1 Byte; a block size of 128 words and a cache size of 8 KB.

While accessing the memory location 0C795H by the CPU, the contents of the TAG field of the corresponding cache line is

- (a) 000011000
- (b) 110001111
- (c) 00011000
- (d) 110010101

[GATE - 2008]

Q.

A 4-way set-associative cache memory unit with a capacity of 16KB is built using a block size of 8 words. The word length is 32 bits. The size of the physical address space is 4 GB. The number of bits for the TAG filed is

P
W

[GATE - 2014]

Q.

The size of the physical address space of a processor is 2^P bytes. The word length is 2^W bytes. The capacity of cache memory is 2^N bytes, the size of each cache block is 2^M words. For a k-way set-associative cache memory, the length (in number of bits) of the tag field is

(a) $P - N - \log_2 K$

(b) $P - N + \log_2 K$

(c) $P - N - M - W - \log_2 K$

(d) $P - N - M - W + \log_2 K$

[GATE - 2018]

P
W

MCQ

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.

1Valid bit

1Modified bit

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? [GATE-2011-CS: 2M]

A 4864 bits

B 6144 bits

C 6656 bits

D 5376 bits

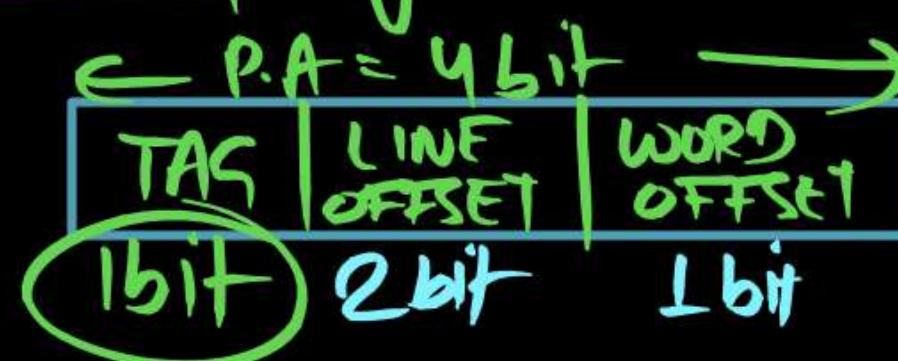
① Direct Mapping



② MM Size = 16Byte Cache Size = 8 Byte Block Size = 2 Byte

#bits for (i) PA (ii) W.O (iii) L.O (iv) TAG bit (v) #LINES (vi) Tag memory

Direct Mapping



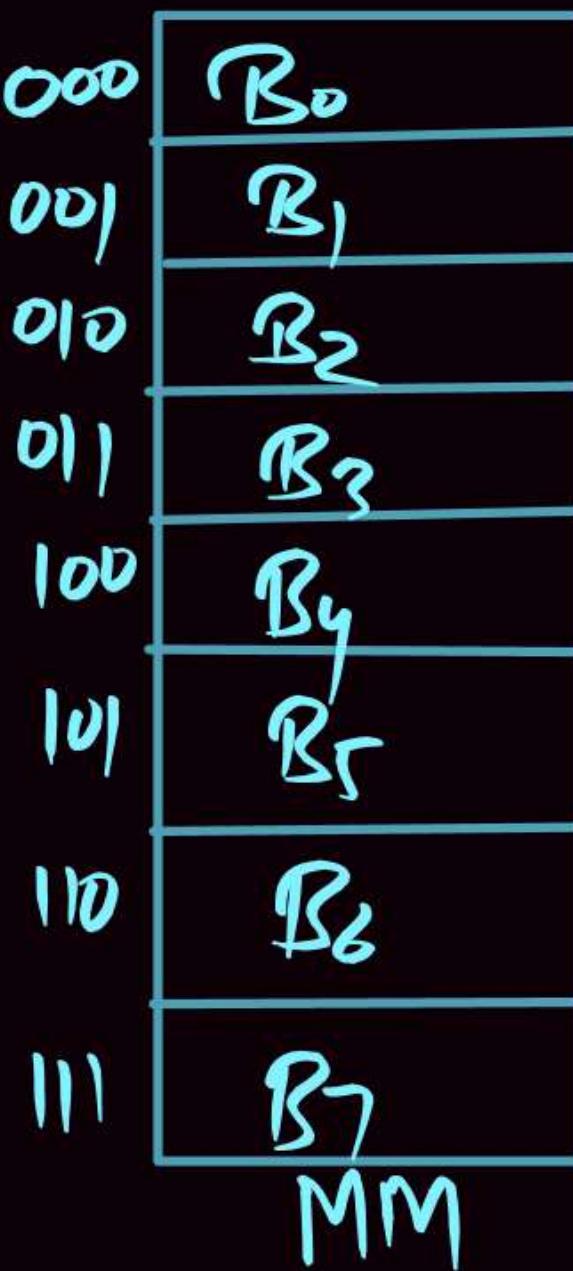
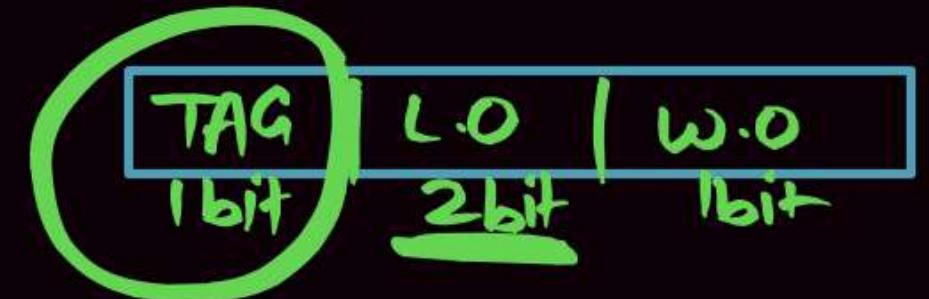
$$\# \text{LINE} = \frac{\text{MM Size}}{\text{Block Size}} = \frac{8}{2} = 4$$

$$L.O = 2 \text{ bit}$$

$$TAG = P.A - (L.O + W.O)$$

$$= 4 - (2 + 1) = 1 \text{ bit}$$

$$\begin{aligned} \text{Tag Memory} &= \# \text{LINES} \text{ Tag bits} \\ &= 4 \times 1 = 4 \text{ bits} \end{aligned}$$



② SET ASSOCIATIVE Mapping:

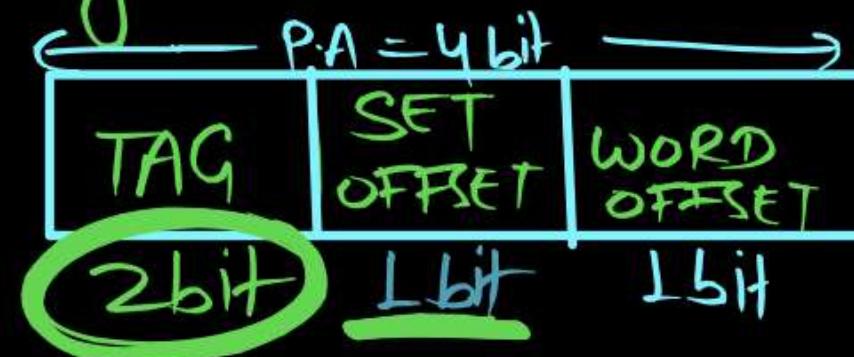


Consider 2 way Set Associative.

MM Size = 1GB, Cache Size = 8 Byte Block size = 2 Byte

#bits for (i) PA (ii) W.O (iii) S.O (iv) TAG bit (v) #LINES (vi) Tag memory

Setⁿ 2 way Set Associative Cache



$$\begin{aligned} \text{Tag Memory} &= \frac{\# \text{LINES}}{\text{Size}} \times \text{Tag bits} \\ &= 4 \times 2 = 8 \text{ bits} \end{aligned}$$

Ans

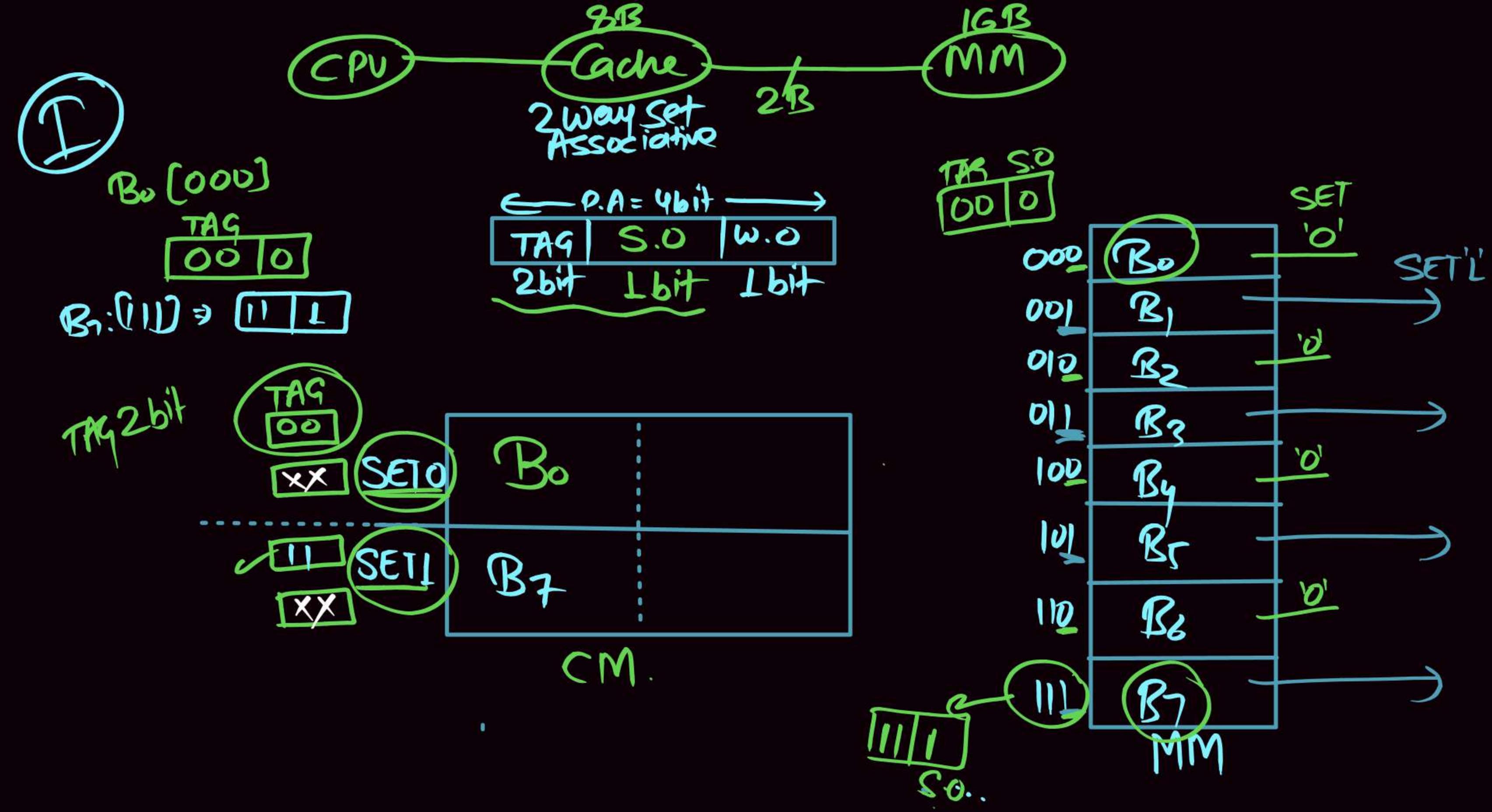
$$\begin{aligned} \text{Tag Memory} &= \# \text{SET} \times \frac{\text{Block size}}{\text{Set}} \times \text{Tag bit} \\ &= 2 \times 2 \times 2 \text{ bit} \\ &= 8 \text{ bits} \end{aligned}$$

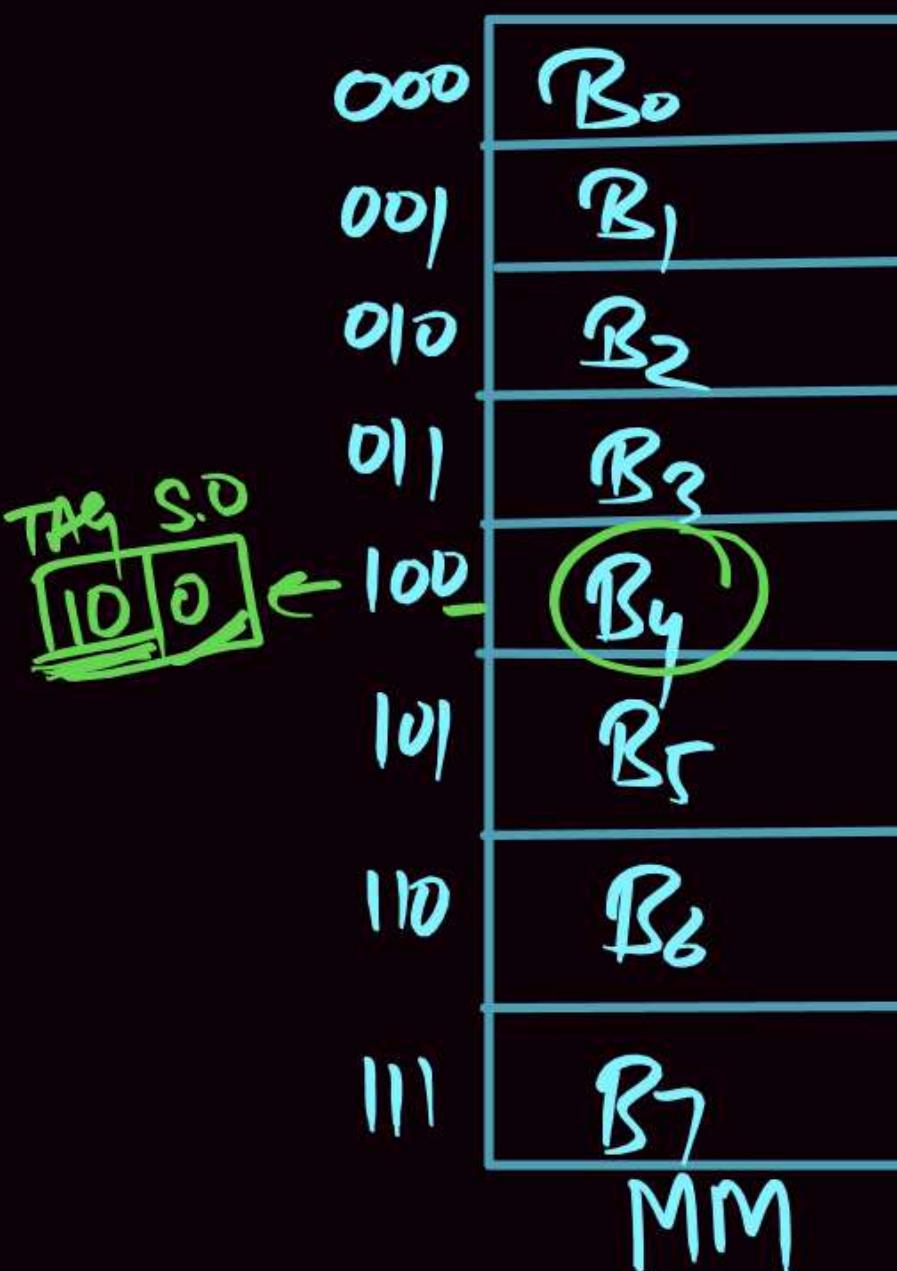
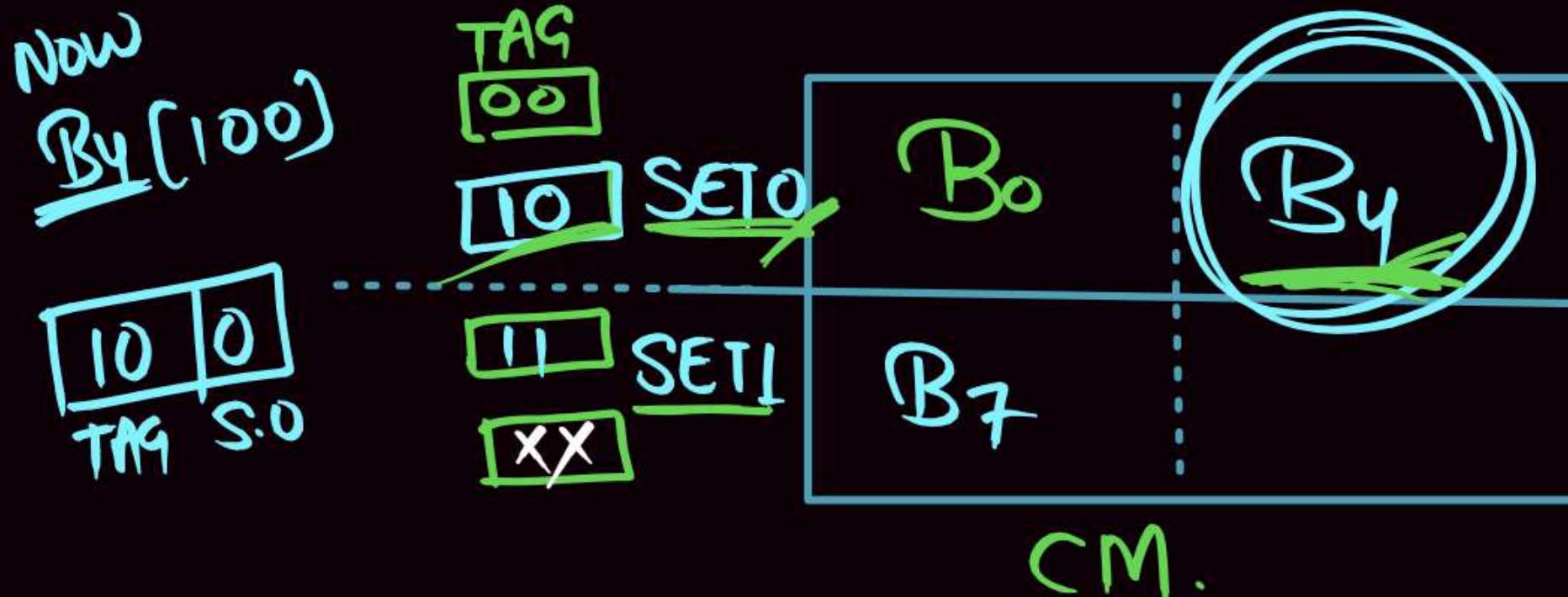
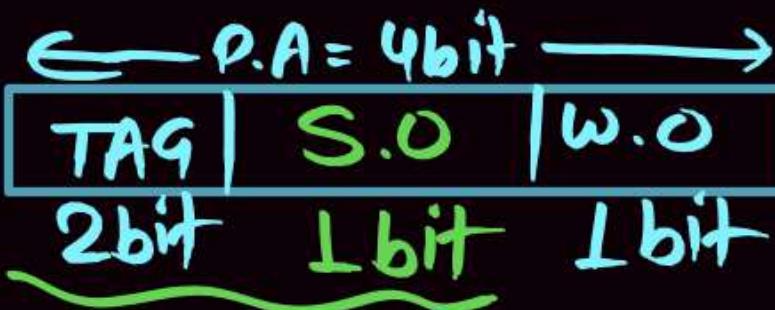
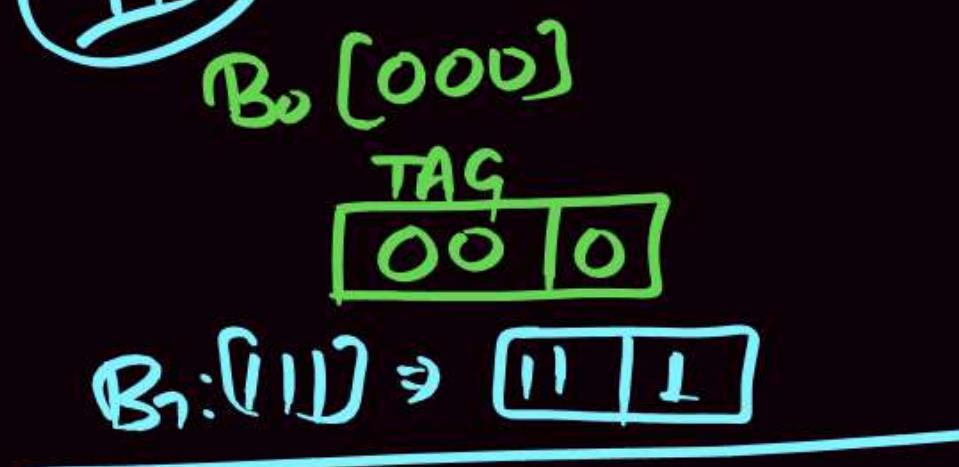
$$\# \text{LINES} = \frac{\text{MM Size}}{\text{Block size}} = \frac{8}{2} = 4 \text{ Lines}$$

$$\# \text{SET} = \frac{\# \text{LINES}}{\text{N ways}} = \frac{4}{2} = 2 \text{ SET}$$

$$\text{S.O} = \text{L bit}$$

$$\begin{aligned} \text{TAG} &= \text{P.A} - (\text{S.O} + \text{W.O}) \\ &= 4 - (1 + 1) = 2 \text{ bit} \end{aligned}$$



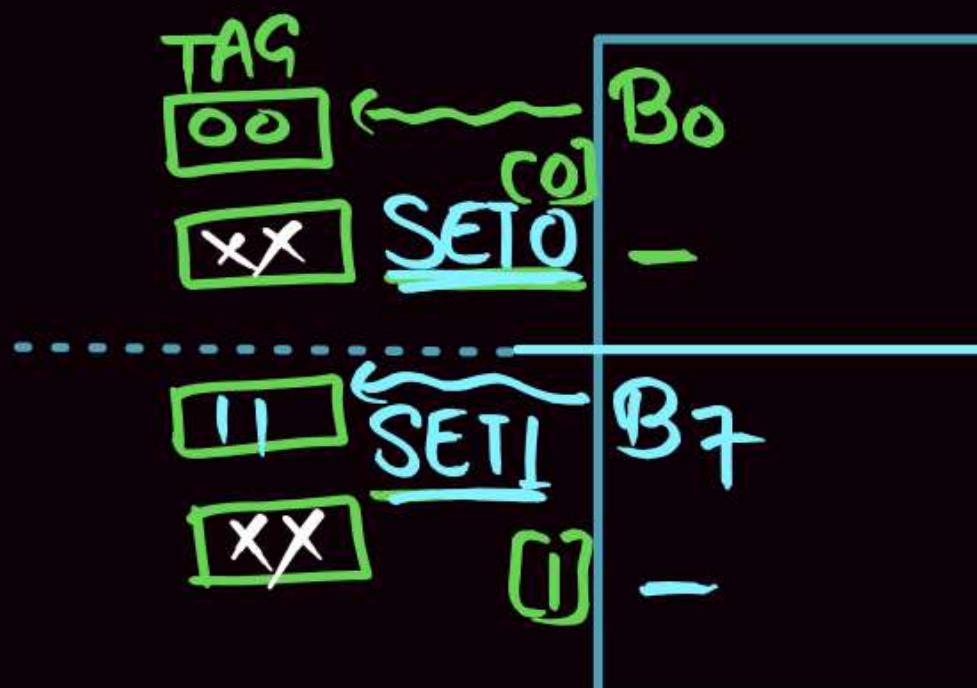
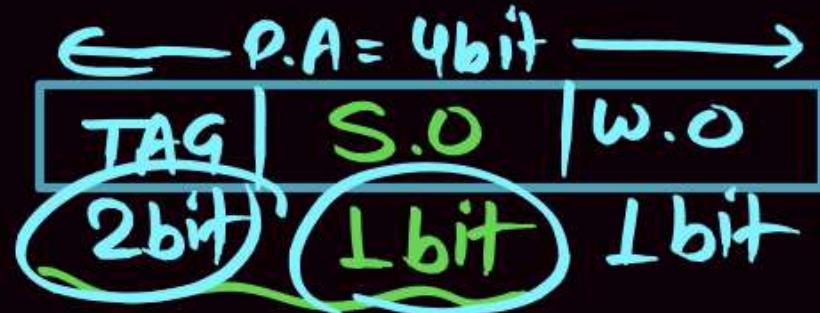




$B_0 [000]$

TAG
00 0

$B_1: [11] \Rightarrow [11]$



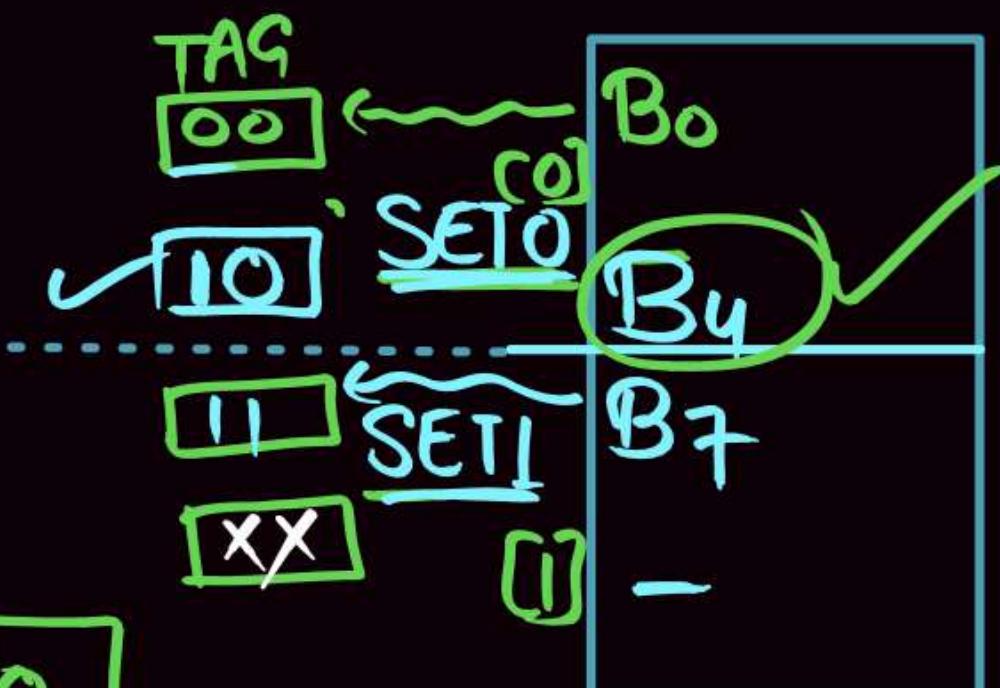
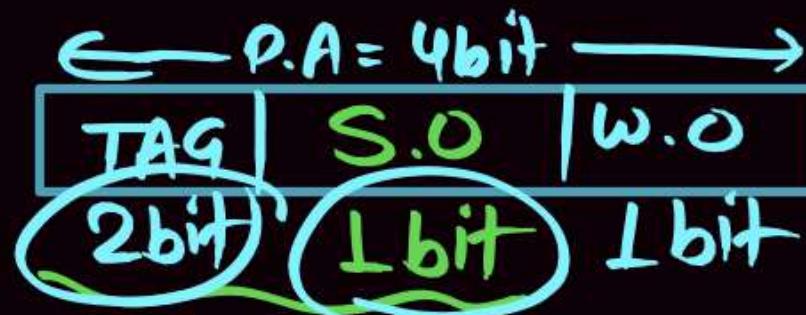
000	B_0
001	B_1
010	B_2
011	B_3
100	B_4
101	B_5
110	B_6
111	B_7
	MM



$B_0 [000]$

TAG
00 0

$B_1: [11] \Rightarrow [11 1]$



000	B_0
001	B_1
010	B_2
011	B_3
100	B_4
101	B_5
110	B_6
111	B_7
	MM

Advantage of Set Associative Mapping:

D₁: B₀ ✓ HIT

T₂: B_y → MISS (then load from MM to CM)

T₃: B₀ → HIT

T₄: B_y → HIT

T₅: B₀ → HIT

T₆: B_y → HIT

| Miss (out of 6 Access)



Set Associative Mapping function

Cache

Set = MM Request MOD #SET is in Cache

Address

(OR)

$K \text{ MOD } S = i$

k: MM Block Number

s: # Cache Set

i: Cache Set Number

$K \text{ MOD } S = i$

S: Number of Set

$\#SET = \frac{\#LINES}{N\text{-Way}}$

Q) What is the Meaning of

- (i) 2 way Set Associative
- (ii) 4 way Set Associative
- (iii) 8 way Set Associative
- (iv) N way Set Associative.

Example1: # Line's = 16 & 2 way set Associative

$$\# \text{SET} = \frac{\# \text{LINES}}{\text{N-way}} = \frac{16}{2} = 8$$

#SET = 8 S = 8

K MODS = i

K MOD 8 = i

✓	✓	✓	✓	✓	✓	✓	✓
---	---	---	---	---	---	---	---

SET₀ SET₁ 2 3 4 5 SET 6 SET 7

K MOD 8 = i 0 - 7

Example2:

#LINE = 16 & 4 way set Associative

$$\# \text{SET} = \frac{16}{4} \Rightarrow 4 \quad S = 4$$

K MOD 4 = i

K MOD S = i

SET 0

SET 1

SET 2

SET 3



Cache

Example3:

#LINE = 16 & 8 way set Associative

$$\# \text{SET} = \frac{16}{8} \Rightarrow 2$$

$$S = 2$$

$$K \bmod 2 = i$$

$$\# \text{SET} = 2$$

$$K \bmod S = i$$

$$K \bmod 2 = i$$

SET 0

SET 1



Q What is the Meaning of

(Seln)

- (i) 2 Way Set Associative → In a One Cache Set we can store 2 mm Block.
- (ii) 4 Way Set Associative → In a One CM set we can store 4 mm Block.
- (iii) 8 Way Set Associative → 8 mm Block store in one CM set.
- (iv) N Way Set Associative → N-mm Block we can store in one set.

Q What is the Meaning of

Seln

- (i) 2 way Set Associative → In a One CM set we can store mm Block by 2 way
- (ii) 4 way Set Associative → By 4 ways
- (iii) 8 way Set Associative → " By 8 ways
- (iv) N way Set Associative → " By N ways

Example4:

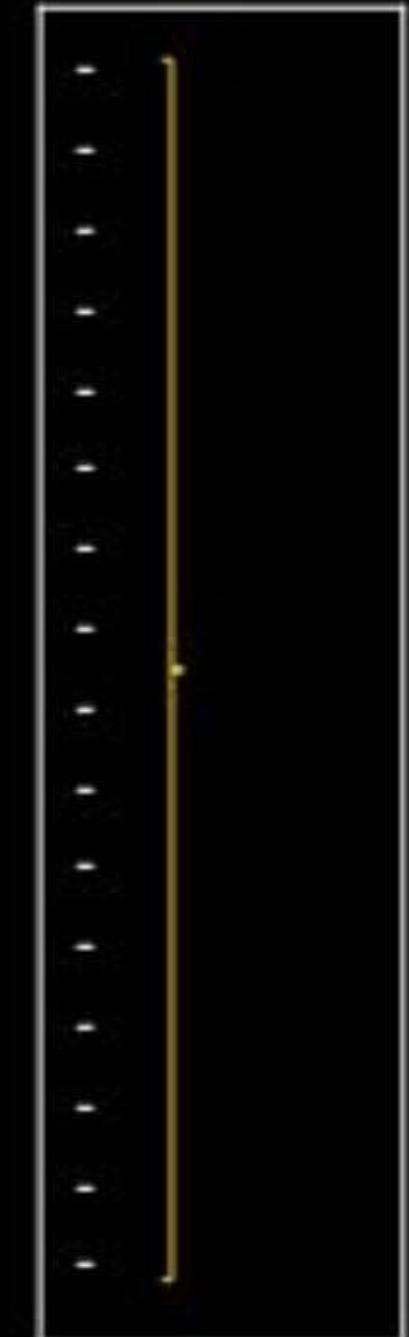
#LINE = 16 & 16 way set Associative

$$\# \text{ SET} = \frac{16}{16} \Rightarrow 1$$

$$S = 1$$

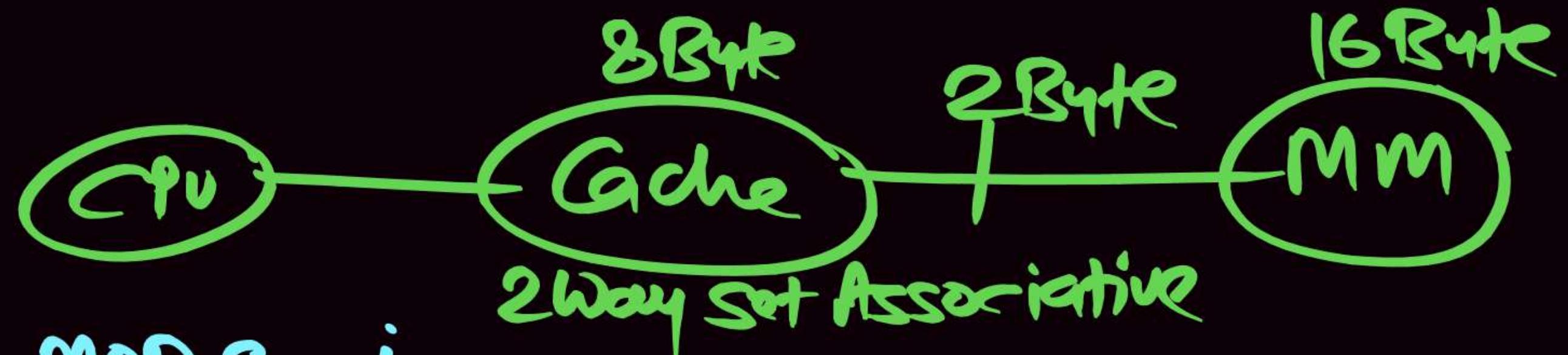
$$K \bmod 1 = i$$

Fully Associative



Fully
Associative
(Whole cache as a set)

2 Way Set Associative .



$$K \bmod S = i$$

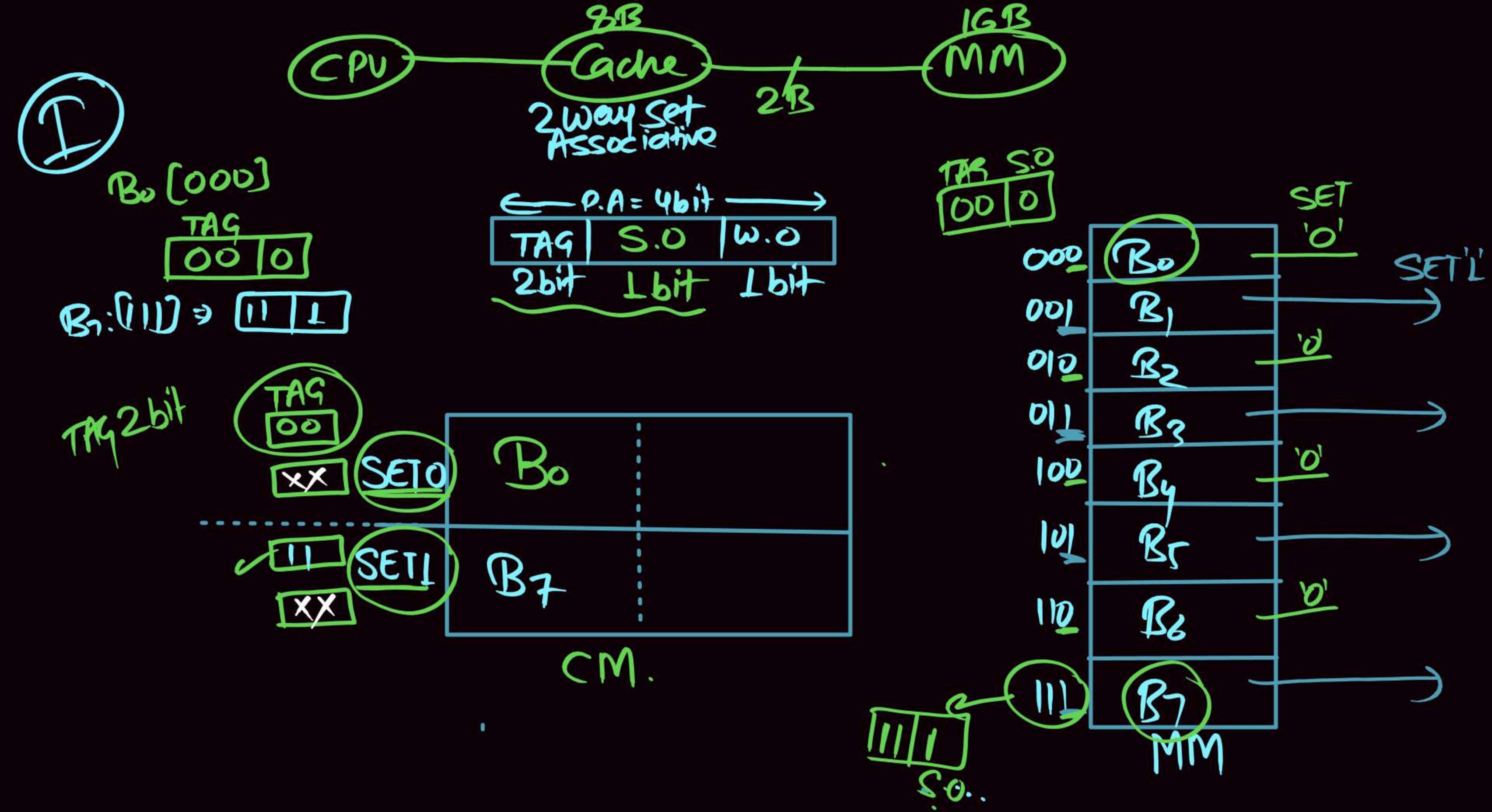
for this
DATA

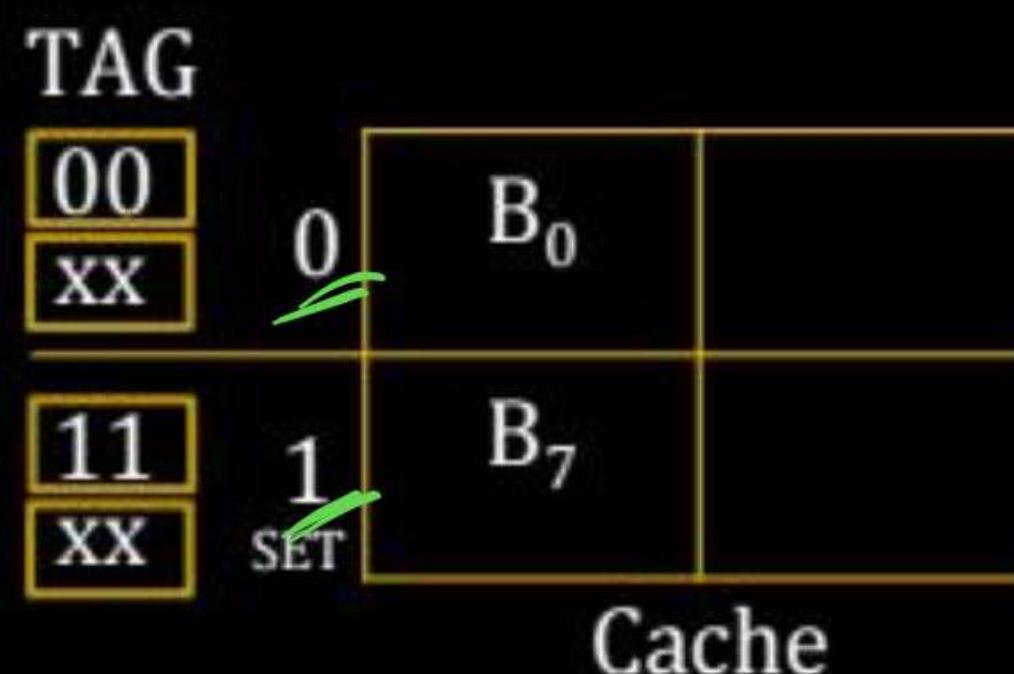
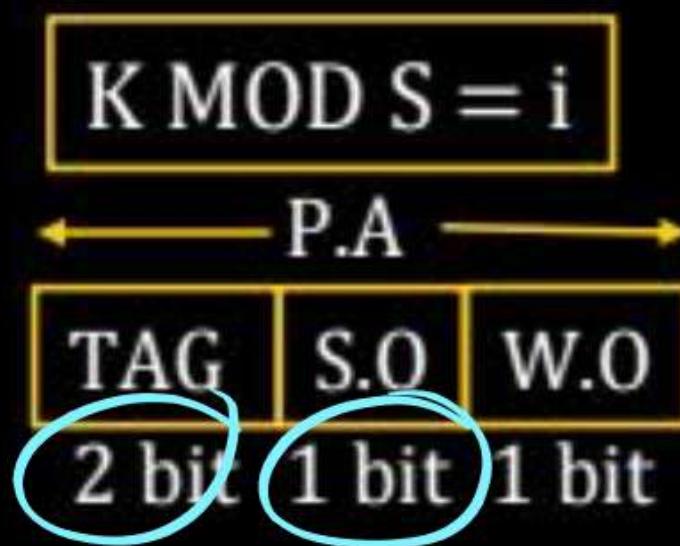
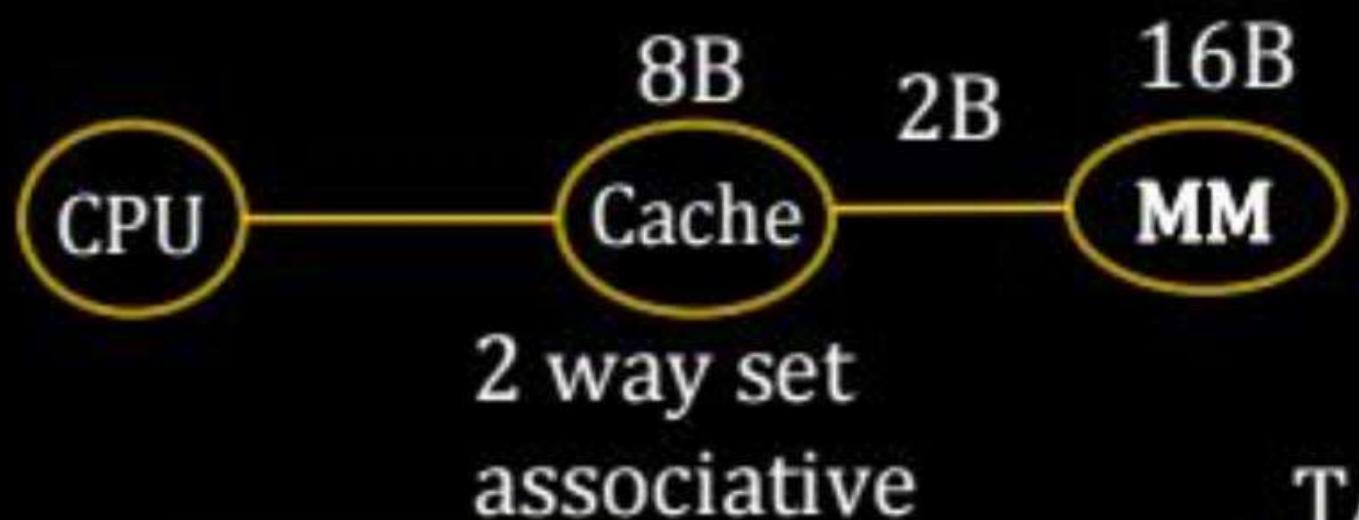
$$\boxed{K \bmod Q = i}$$

$\leftarrow P.A = 4\text{bit} \rightarrow$		
TAG	S.O	W.D

2bit 1bit 1bit

2Way Set
Associative





P.W

000	B ₀
001	B ₁
010	B ₂
011	B ₃
100	B ₄
101	B ₅
110	B ₆
111	B ₇

MM BLOCK Mapping Tech. CM SET

$$B_0[000] \xrightarrow{\begin{array}{l} K \bmod S = i \\ 0 \bmod 2 = '0' \end{array}} \underline{\text{SET '0'}}$$

TAG	SET
00	0

$$B_7[111] \xrightarrow{\begin{array}{l} K \bmod S = i \\ 7 \bmod 2 = '1' \end{array}} \underline{\text{SET '1'}}$$

TAG	SET
11	1

$$\frac{\text{Tag Memory Size}}{\text{#SETS} \times \frac{\text{\#LINES In a each set}}{\text{Tag bits}}}$$

Example: # SET = 2 $\frac{\text{\# LINE in Each set}}{= 2}$ TAG = 2 bit

$$\frac{\text{Tag Memory size}}{= 2 \times 2 \times 2 \\ = 8 \text{ bits}}$$

Working of 2 way set
Associative Cache.

Note

In a Direct Mapping Only 1 Comparator is
Used to check Cache Hit @ Cache Miss.

Note

In a Set Associative Mapping Along with 1 Comparator
Multiplex(MUX) is also used to check Cache Hit @ cache Miss.

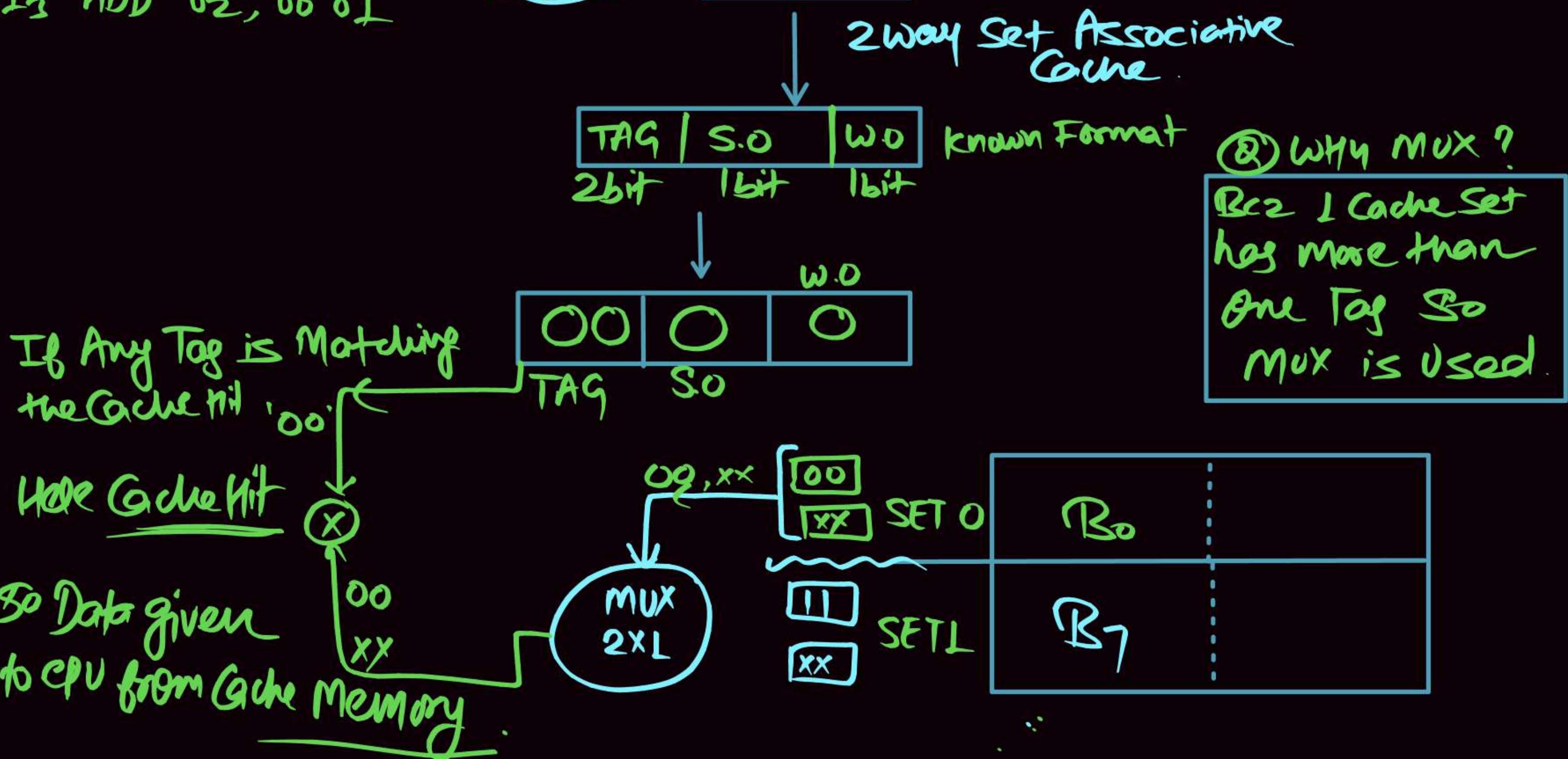
Note

In a fully Associative N-Comparators (N: Number of Cache
LINE) are used to check Cache Hit @ Cache Miss.

..

I_1 MOV δ_0 [0000]
 I_2 mov δ_1 [1000]
 I_3 ADD $\delta_2, \delta_0 \delta_1$

CPU generated Request go to the Cache.



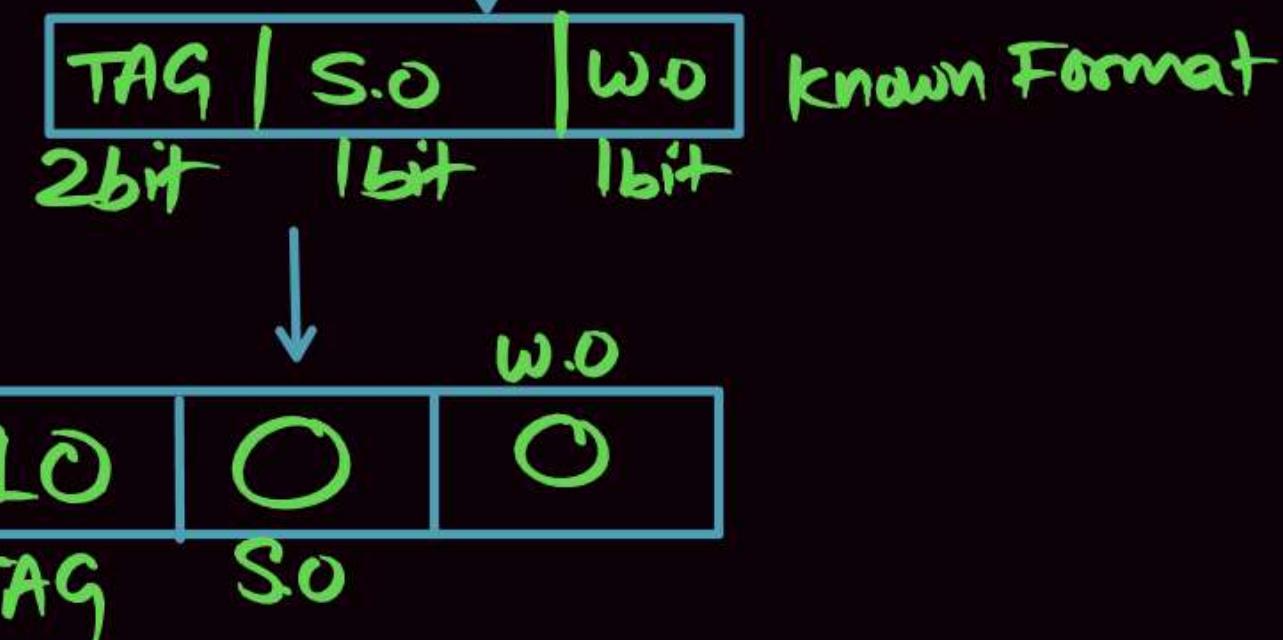
I_1 MOV δ_0 [0000]
 I_2 MOV δ_1 [1000]
 I_3 ADD $\delta_2, \delta_0 \delta_1$

CPU generated Request go to the Cache.

Request II

NO Tag is Matching
So Cache Miss

So Reference
forward to MM



Q) Why Mux?
Bcz 1 Cache Set has more than one tag so Mux is used.



B_4 : 

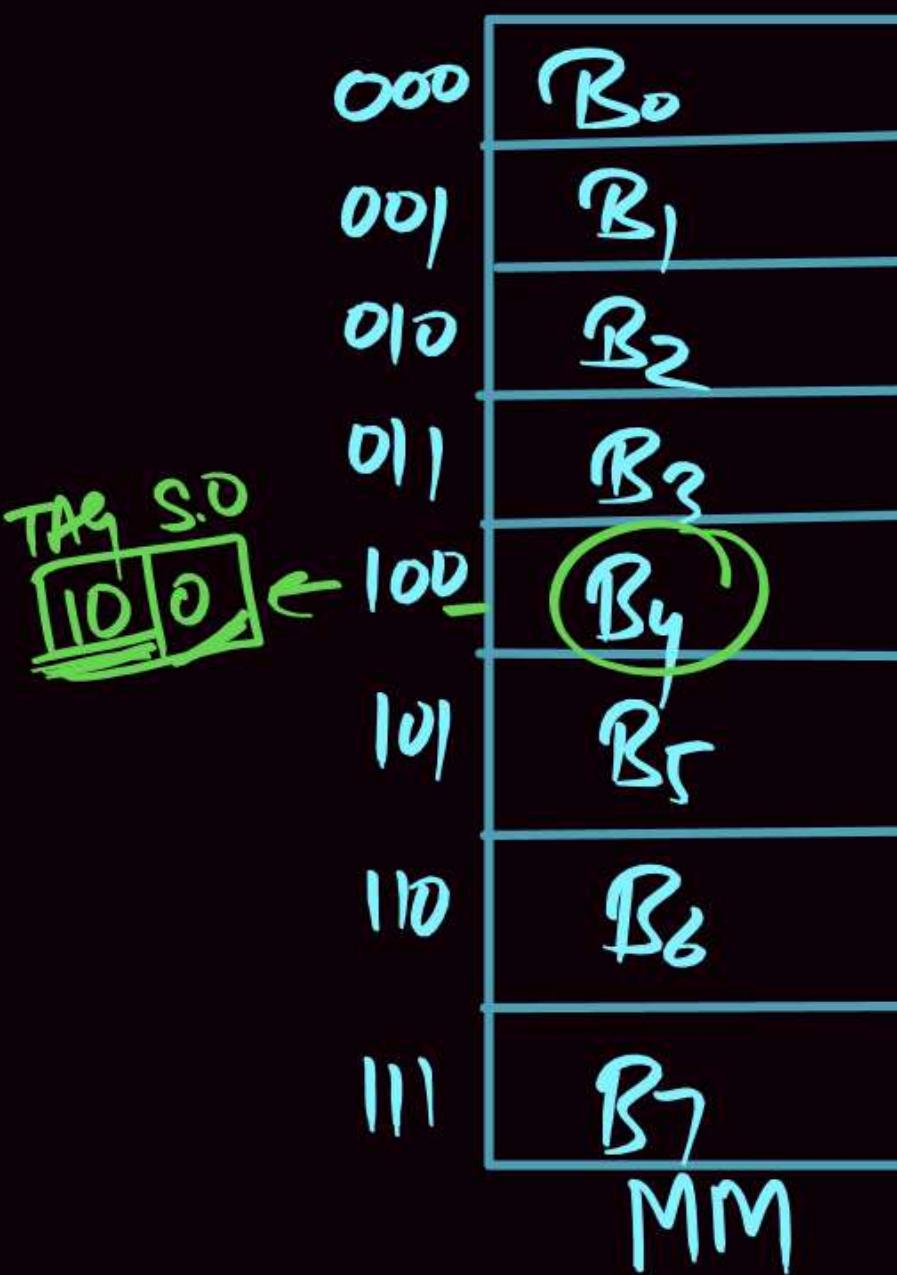
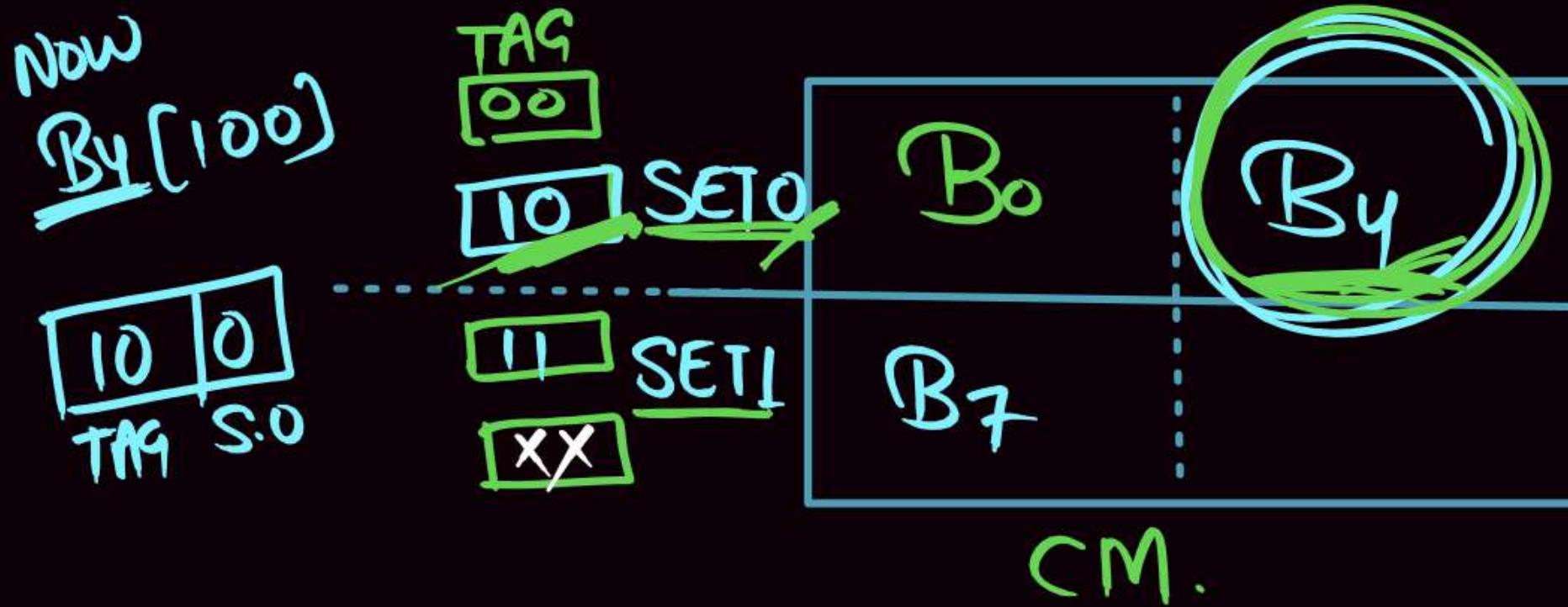
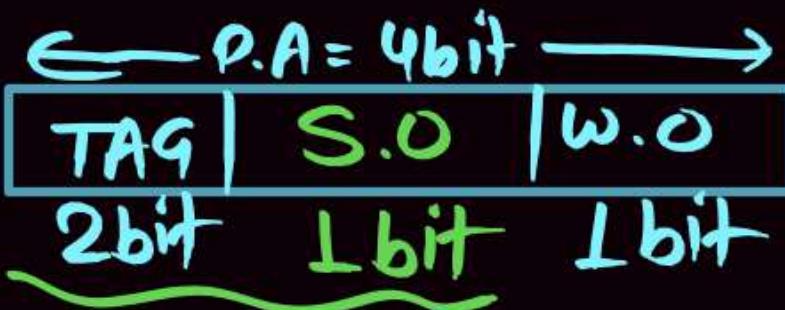
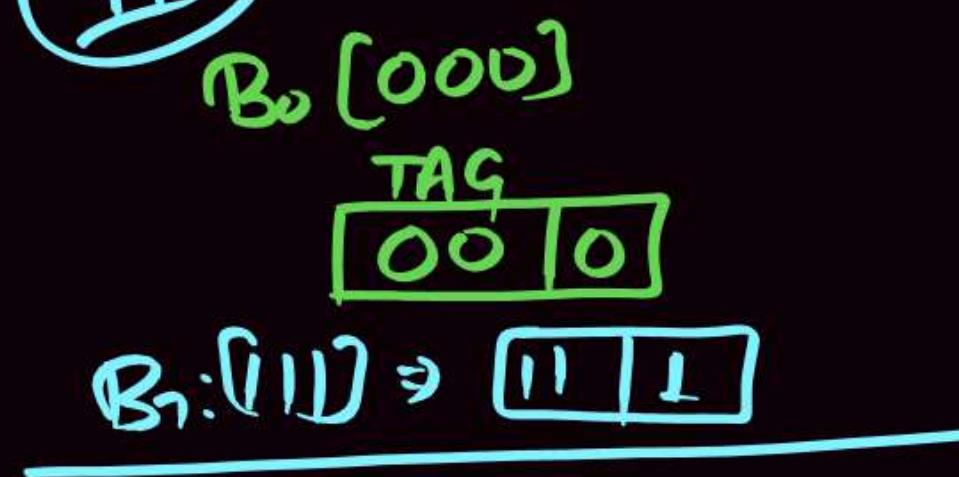
000	B_0
001	B_1
010	B_2
011	B_3
100	B_4
101	B_5
110	B_6
111	B_7

TAG	SET
10	0

2bit 1bit

$$\frac{k \bmod s = i}{4 \bmod 2 = } \underline{\text{SET '0'}}$$

MM



Advantage of Set Associative Mapping:

so conflict miss reduce.

D₁: B₀ ✓ HIT

T₂: B_y → MISS (then Load from MM to CM).

T₃: B₀ → HIT

T₄: B_y → HIT

T₅: B₀ → HIT

T₆: B_y → HIT



1 Miss (out of 6 Access)

1) In direct Mapping

Hit Latency = Latency of Tag comparator

2) In Set Associative Mapping

Hit Latency = Latency of Tag comparator + Latency of Multiplexer

Q.

P
W

[Common Data for this and next question]

A computer has a 256 Kbyte, 4-way set associative. Write back data cache with block size of 32 Bytes. The processor sends 32 bit address to the cache controller. Each cache tag directory entry contains, in addition to address tag, 2 valid bits, 1 modified bit and 2 replacement bit.

The number of bits in the tag field of am address is

- (a) 11
- (b) 14
- (c) 16
- (d) 27

[GATE - 2012: 2 Marks]

Q.

[Common Data from previous question]

A computer has a 256 Kbyte, 4-way set associative. Write back data cache with block size of 32 Bytes. The processor sends 32 bit address to the cache controller. Each cache tag directory entry contains, in addition to address tag, 2 valid bits, 1 modified bit and 2 replacement bit.

[GATE - 2012: 2 Marks]

The size of the cache tag directory is

- (a) 160 Kbits
- (b) 136 Kbits
- (c) 40 Kbits
- (d) 32 Kbits

P
W

Consider a machine with a byte addressable main memory of 2^{32} bytes divided into blocks of size 32 bytes. Assume that a direct mapped cache having 512 cache lines is used with this machine. The size of the tag field in bits is ____.

[GATE-2017(Set-1)-CS: 2M]

A computer system with a word length of 32 bits has a 16 MB byte-addressable main memory and a 64 KB, 4-way set associative cache memory with a block size of 256 bytes. Consider the following four physical addresses represented in hexadecimal notation.

$A_1 = 0 \times 42C8A4, A_2 = 0 \times 546888, A_3 = 0 \times 6A289C, A_4 = 0 \times 5E4880$

Which one of the following is TRUE?

[GATE-2020-CS: 2M]

- A A1 and A3 are mapped to the same cache set.
- B A2 and A3 are mapped to the same cache set.
- C A3 and A4 are mapped to the same cache set.
- D A1 and A4 are mapped to different cache sets.

NAT

The width of the physical address on a machine is 40 bits. The width of the tag field In a 512 KB 8-way set associative cache is _____bits.

[GATE-2016(Set-2)-CS: 2M]

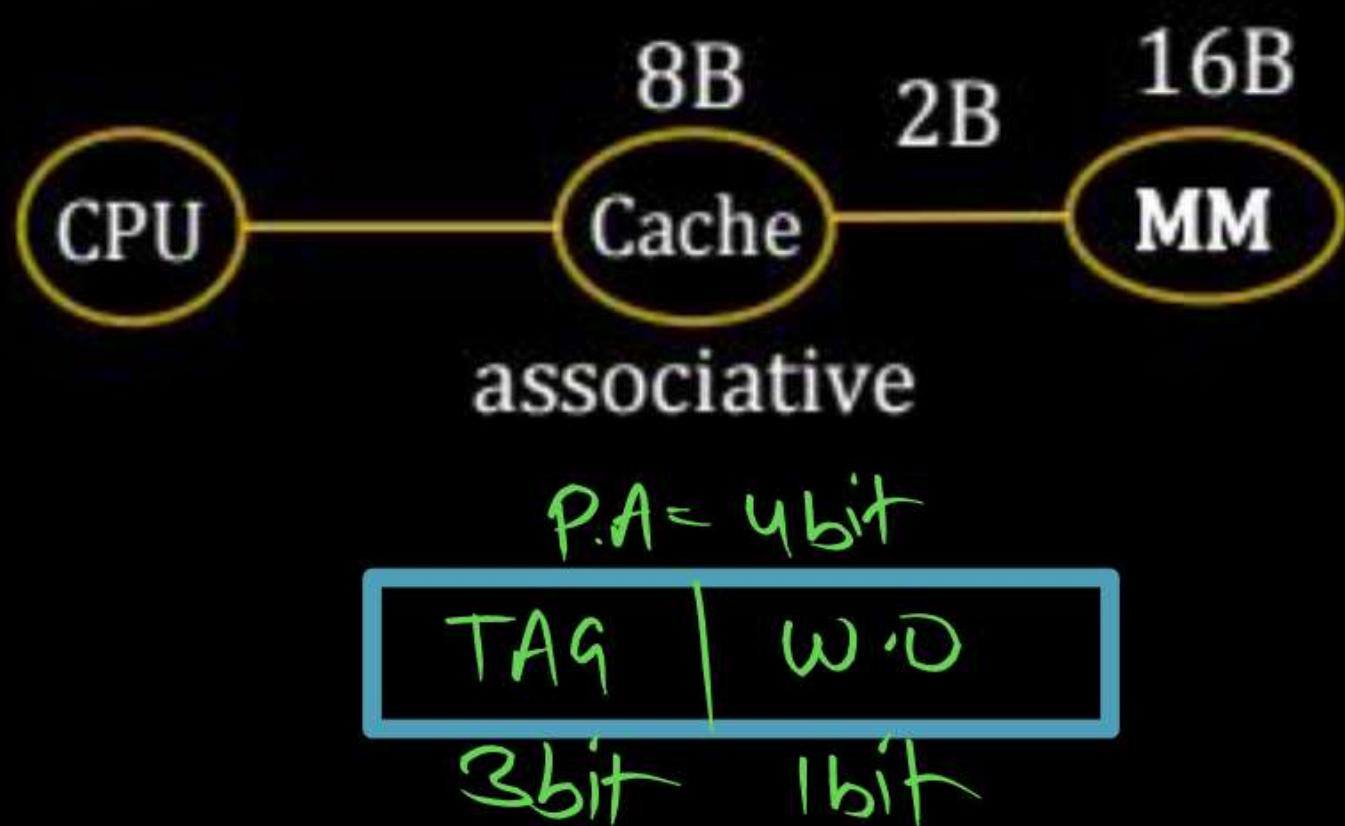
NAT

A 4-way set-associative cache memory unit with a capacity of 16 KB is built using a block size of 8 words. The word length is 32 bits. The size of the physical address space is 4 GB. The number of bits for the TAG field is ____.

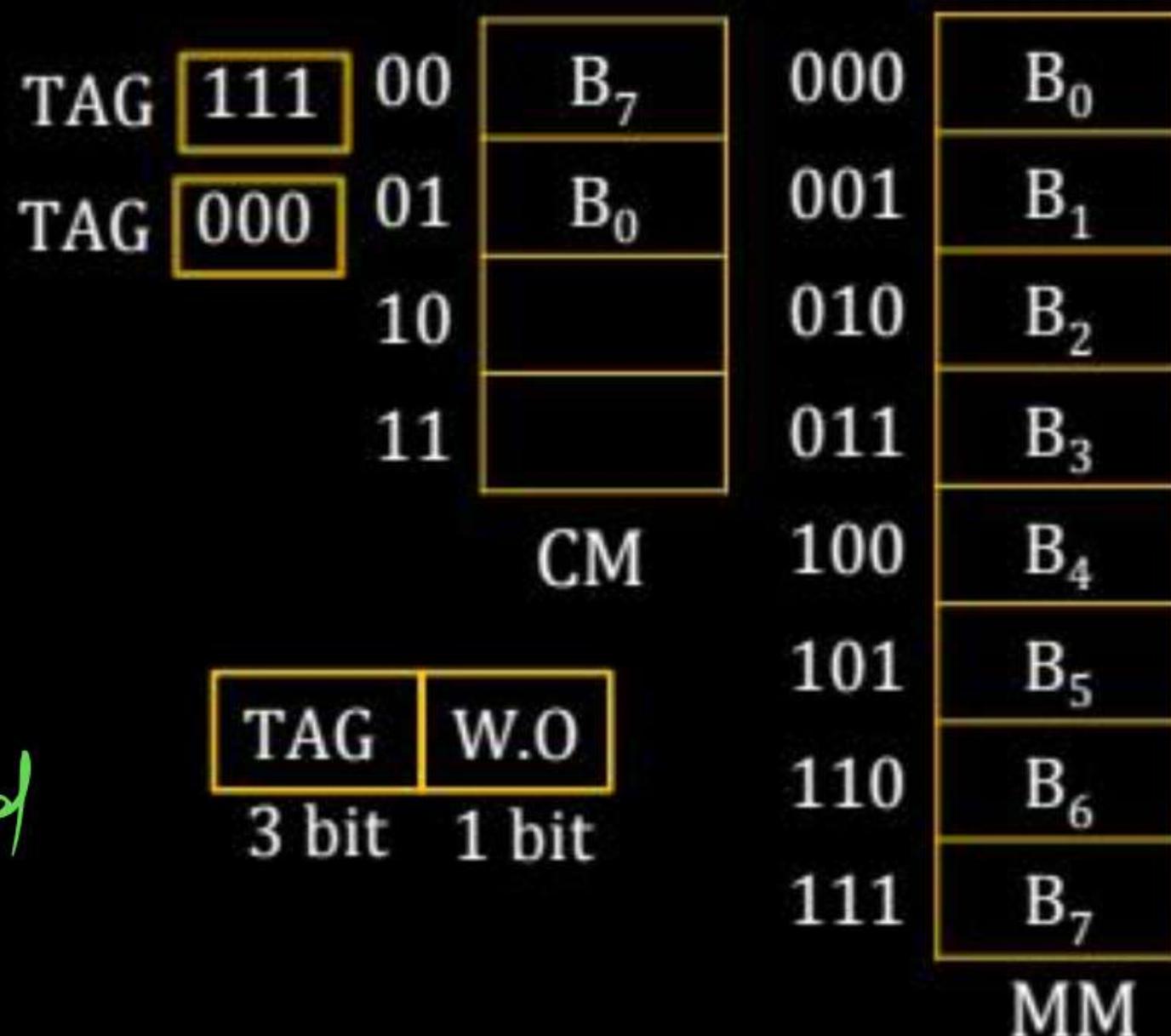
[GATE-2014(Set-2)-CS: 1M]

3) Associative Mapping

- In this no mapping function is used to transfer the data from MM to CM.
- No Conflict Miss



for each CM Line Combinator is Required
to Check Cache Hit or Miss.



MM Block	Associative Mapping	CM Line
----------	---------------------	---------

B _{7[111]}	<u>No mapping</u> Function	Any line
B _{0[000]}	<u>No mapping</u> Function	Any line

$$\frac{\text{Tag Memory}}{\text{Size}} = \# \text{ LINES} \times \text{Tag bits}$$

Example: #LINE = 4 & Tag bits = 3

$$\frac{\text{Tag Memory}}{\text{Size}} = 4 \times 3 = 12 \text{ bits}$$

Q.

Consider fully associative cache consists 8 Block & MM contains
128 Block & Request made by the CPU:
119, 84, 37, 0, 16, 0, 84, 120, 121, 93, 37, 0, 43, 39, 47, 48.
Calculate # of compulsory & Capacity miss?

P
W



0	119 43
1	84 39
2	37 47
3	0 48
4	16
5	120
6	121
7	93

Hit: 0, 84, 37, 0

#Hits: 4

Compulsory Miss = 8

Capacity Miss = 4

Total Miss = 12

Replacement Algorithm

When Cache is Full, then replacement algorithm are required to replace the exist cache block with new block.

In the CM design 3 type of replacement algorithm is used.

- 1) Random Algorithm
- 2) FIFO Replacement
- 3) LRU Replacement

In the random algorithm, any cache block can be replaced based on the random selection.

Q.

Consider 4 block cache memory (initially empty) with the following MM block references.

7, 8, 10, 15, 7, 8, 16, 7, 8, 10

Identify the Hit Ratio using

- | | |
|---------------------------|---------------------------------------|
| (i) FIFO | (ii) LRU |
| (iii) Direct Mapped cache | (iv) 2 - way Set Associative with LRU |

P
W

Q.

Consider a small two-way set-associative cache memory, consisting of 4 blocks. For choosing the block to be replaced, use the least recently used (LRU) scheme. The number of cache misses for the following sequence of block addresses is 8, 12, 0, 12, 8

- (a) 2
- (b) 3
- (c) 4
- (d) 5

[GATE - 2004]

P
W

Q.

Consider a 2-way set associative cache with 256 blocks and uses LRU replacement. Initially the cache is empty conflict misses are those misses which occur due to contention of multiple blocks for the same cache set. Compulsory misses occur due to first time access to the block. The following sequence of accesses to memory blocks (0, 128, 256, 128, 0, 128, 256, 128, 1, 129, 257, 129, 1, 129, 257, 129) is repeated 10 times. The number of conflict misses experienced by the cache is _____.

P
W

[GATE - 2017]

Q.

[Common Data from previous question]

P
W

Consider two cache organization. The first one is 32 KB 2-way set associative with 32-byte block size. The second one is of the same size but direct mapped. The size of an address is 32 bits in both cases. A 2-to-1 multiplexer has latency of 0.6 ns while a k-bit comparator has a latency of $k/10$ ns. The hit latency of the set associative organization is h_1 while that of the direct mapped one is h_2 . The value of h_2 is

- (a) 2.4 ns
- (b) 2.3 ns
- (c) 1.8 ns
- (d) 1.7 ns

[GATE - 2006: 2 Marks]

Q.

P
W

A computer system has a level - 1 instruction cache (1-cache), a level-1 data cache(D-cache) and a level-2 cache(L2-cache) with the following specifications.

	Capacity	Mapping method	Block size
1-cache	4K words	Direct mapping	4 words
D-cache	4K words	2-way set associative mapping	4 words
L2-cache	64K words	4-way set associative mapping	16 words

Capacity mapping method block size 1-cache 4K words direct mapping 4 words D-cache 4 k words 2 way set associative mapping 4 words L2-cache 64K words 4-way set associative mapping 16 words. The length of the physical address of a word in the main memory is 30 bits. The capacity of the tag memory in the 1-cache, D-cache & L2-cache is. Respectively,

- (a) $1K \times 18\text{-bit}$, $1K \times 19\text{-bit}$, $4K \times 16\text{-bit}$
- (b) $1K \times 16\text{-bit}$, $1K \times 19\text{-bit}$, $4K \times 18\text{-bit}$
- (c) $1K \times 16\text{-bit}$, $512 \times 18\text{-bit}$, $1K \times 16\text{-bit}$
- (d) $1K \times 16\text{-bit}$, $512 \times 18\text{-bit}$, $1K \times 18\text{-bit}$

[GATE - 2006: 2 Marks]

Types of Misses

In the CM design 3 types of misses are present.

- 1) Compulsory miss - (Cold start miss / first reference miss)

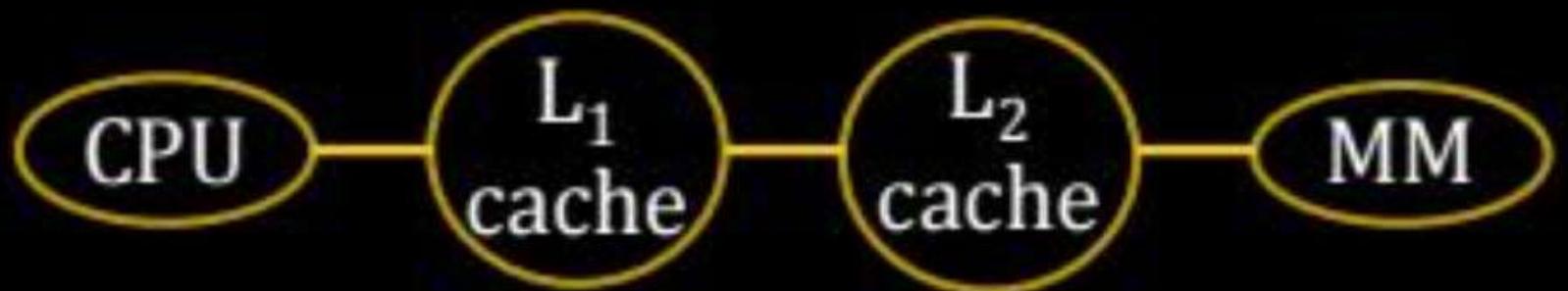
This miss will occur when the very first reference to the cache itself a miss.

- 2) Capacity Miss - This miss will occur when cache is full.
- 3) Conflict Miss (Collision miss / reference miss)

This miss will occur when the too many blocks are placed into same cache line or same cache SET.

Multi level cache

- ❑ To reduce the miss penalty multi-level caches are used in the system design.
- ❑ The number of cycles required to transfer the data from higher levels to L_1 due to miss operation is called as miss penalty



Local Miss Rate [LMR]

Global Miss Rate [GMR]

$$\text{Local Miss Rate} = \frac{\text{\#misses in the cache}}{\text{\# accesses to that cache}}$$

$$\text{Global Miss Rate} = \frac{\text{\#misses in the cache}}{\text{Total \#CPU generated reference}}$$

Average access time of the memory is always calculated in terms of Hit time, miss rate and miss penalty is as follows:

$$T_{avg} = \text{Hit time } L_1 + (\text{Miss Rate } L_1 * \text{Miss penalty } L_1)$$

$$\text{Miss penalty } L_1 = \text{Hit time } L_2 + (\text{Miss rate } L_2 * \text{Miss penalty } L_2)$$

$$\text{Miss penalty } L_2 = \text{MM Access Time}$$

**THANK
YOU!**

