

COMPUTER SCIENCE

Computer Organization and Architecture

Cache Memory

Lecture_04



Vijay Agarwal sir



A graphic of a construction barrier with orange and white diagonal stripes and two yellow bollards at the top.

TOPICS TO BE COVERED

- o1 Cache Memory**
- o2 Mapping Techniques**

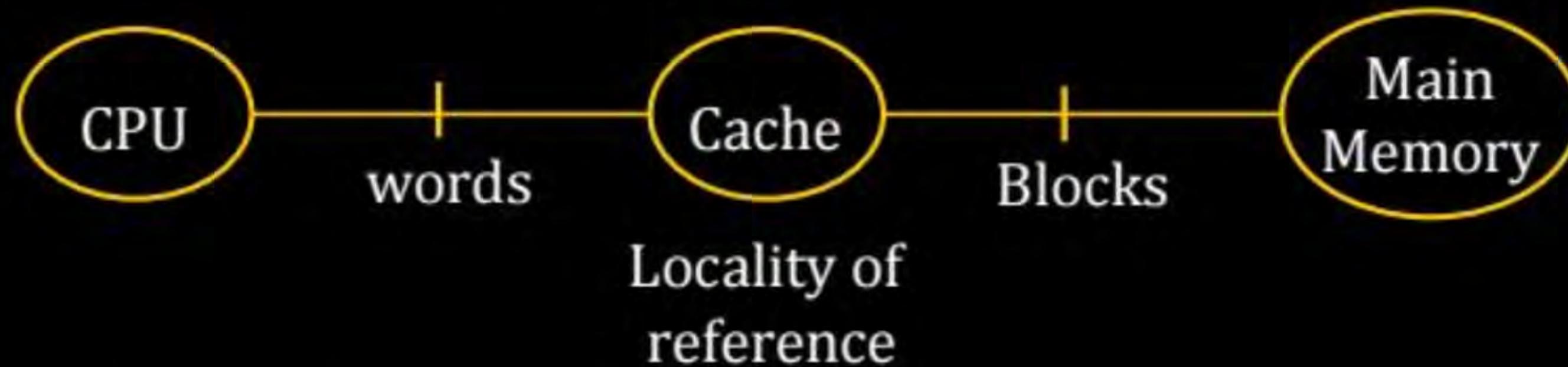
Mapping

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

Memory Organization

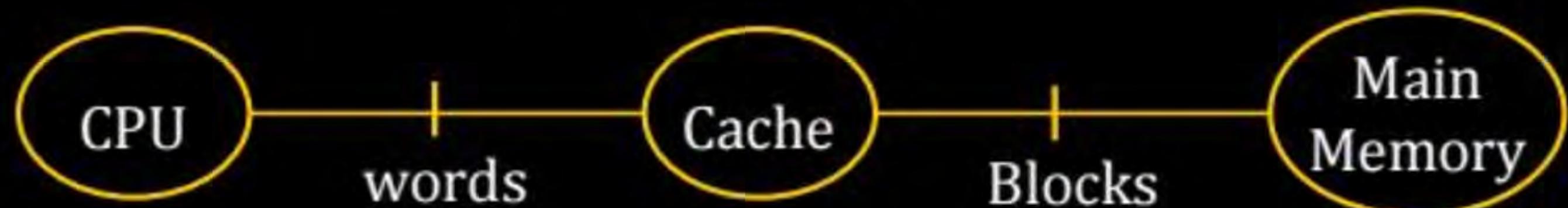


Cache Memory

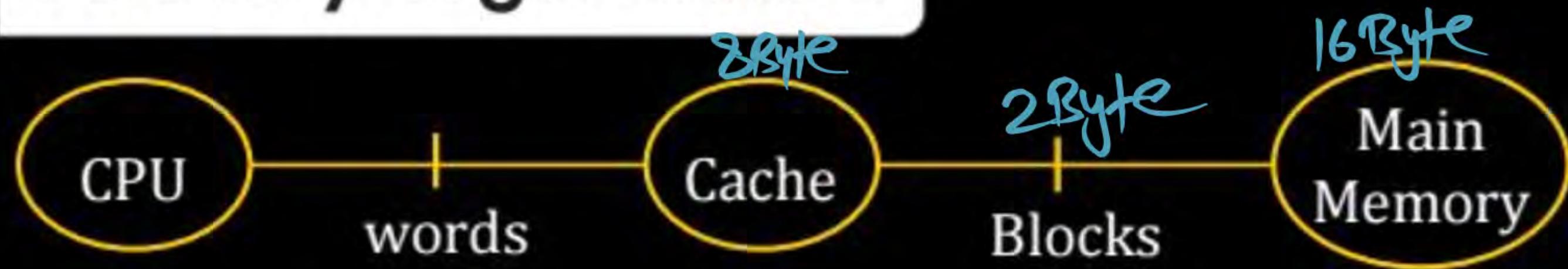


- ✓ 1) Memory Organization.
- ✓ 2) Mapping Technique.
- ✓ 3) Replacement Algorithm.
- ✓ 4) Updating Technique.
- ✓ 5) Multi level cache.

Memory Organization



Memory Organization



Cache Memory

Line offset	Word Block OFFSET
-------------	----------------------

↓

Number of CM
Blocks (CM Lines)

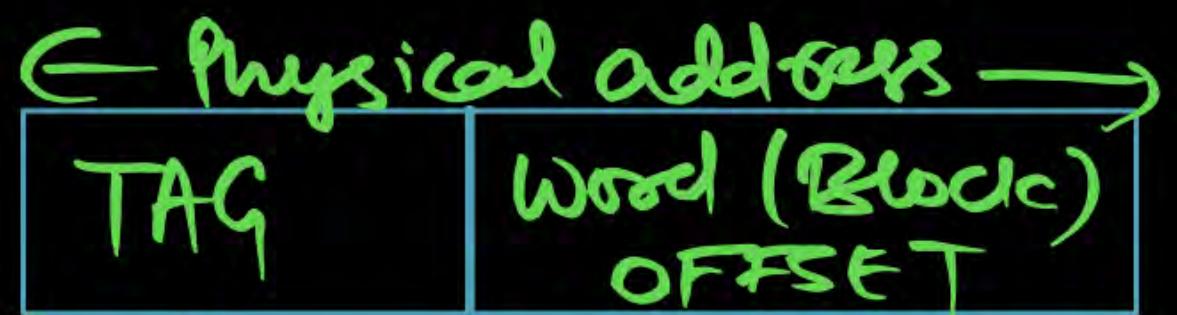
↓

#Word Address in
a Block

↓
 $\log_2 \#CM\text{Lines}$

↓
 $\log_2 \text{BlockSize}$

Main Memory



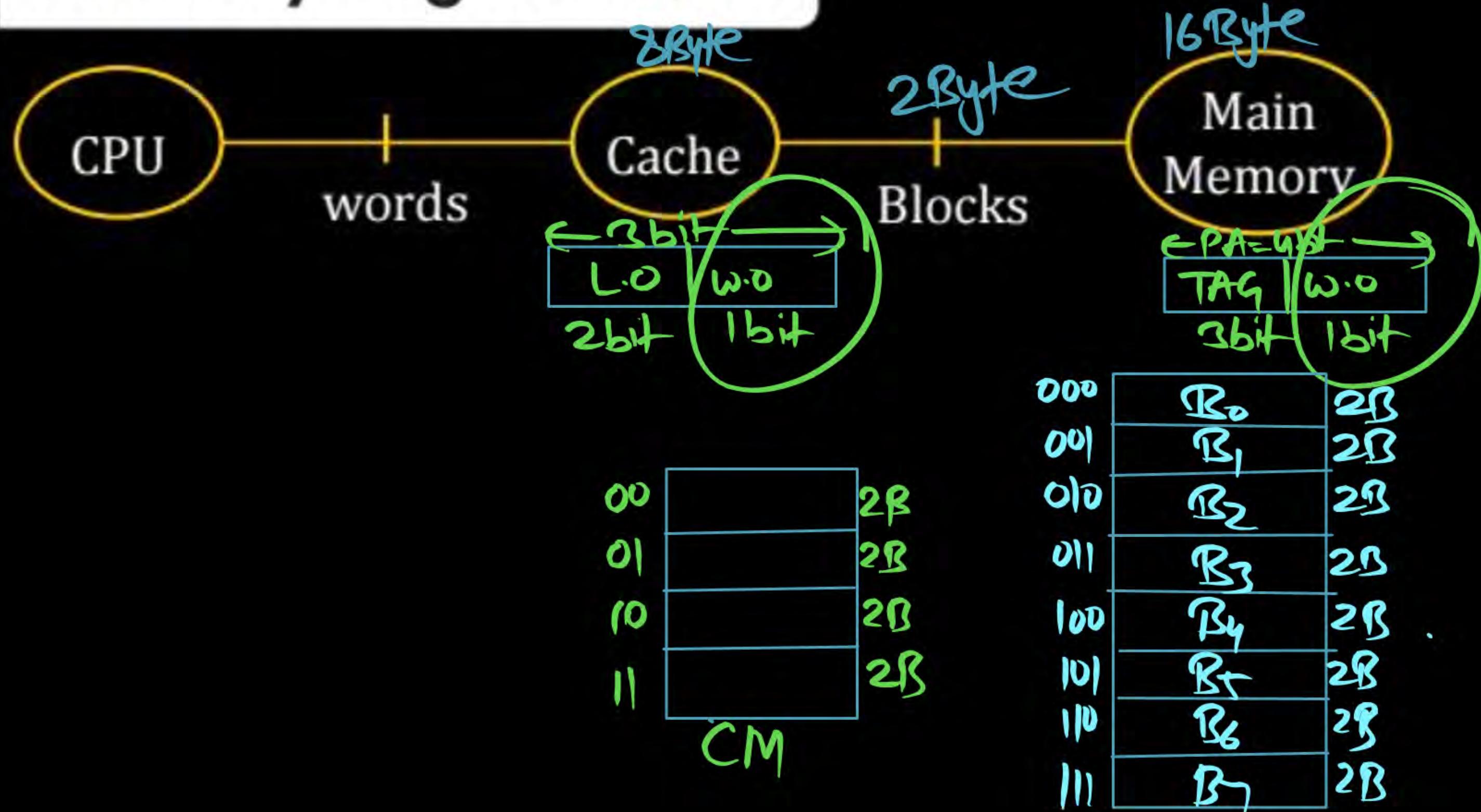
↓
Number of
Main Memory
Blocks

$$\lceil \log_2 \# \text{MM Blocks} \rceil$$

↓
Word Address in
a Block

$$\lceil \log_2 \text{Block Size} \rceil$$

Memory Organization



Q.

Consider a Cache memory which is Indexed with 16 bit address,
organized into 32 byte block size & physical address is 32 bits.

P
W

- (i) # CM Block
- (ii) # MM Blocks
- (iii) CM Address format
- (iv) MM Address format

Q.

Consider a system MM = 256MB, Cache = 128KB & Block size = 512 Byte then

P
W

- 1) # LINES?
- 2) #MM Blocks
- 3) L.O & W.O Format
- 4) TAG & W.O Format

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:

①

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:



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

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

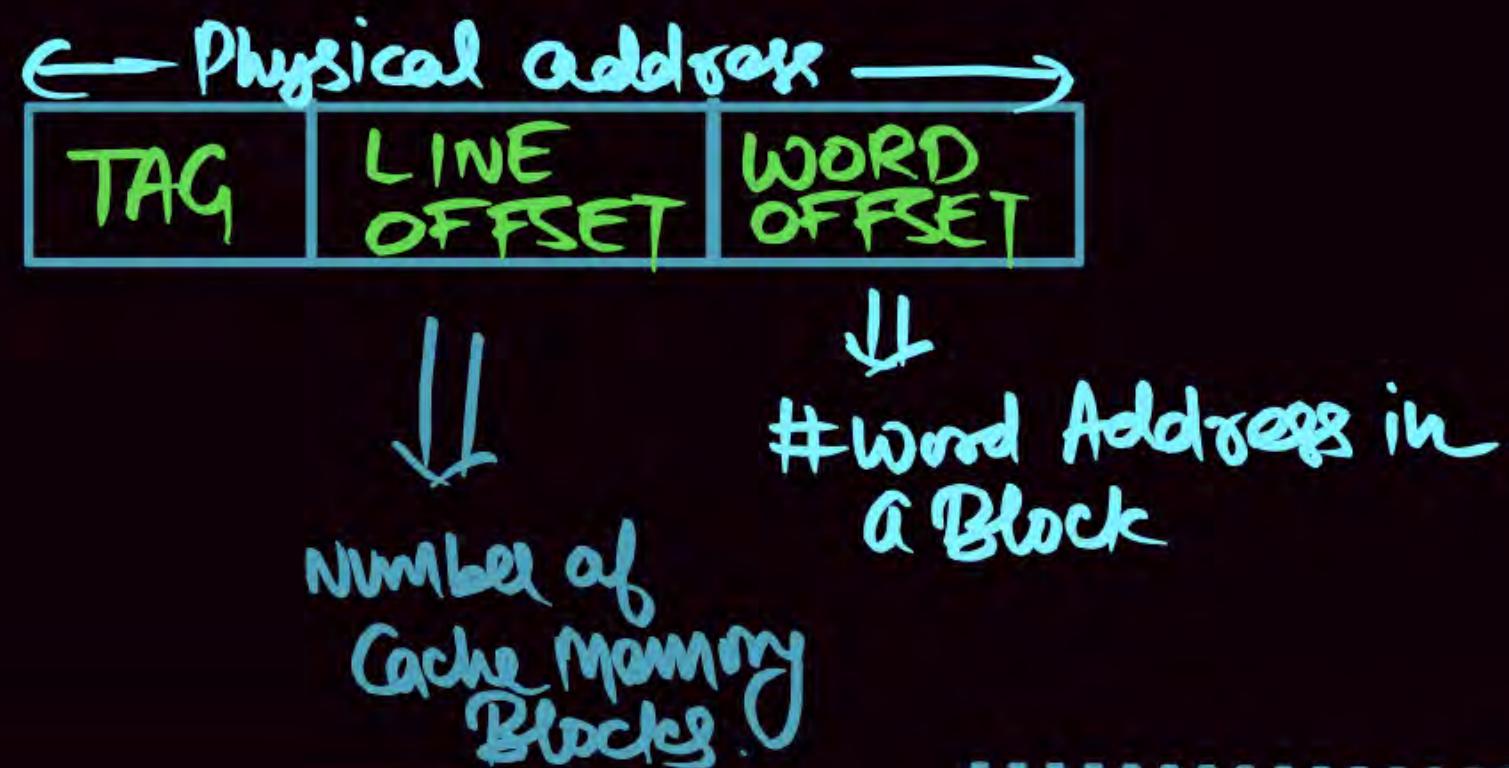
$$\text{Word Offset} = \log_2(\text{Block Size})$$

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

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

Direct Mapping

In Direct Mapped Cache Physical address is interpreted as :



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

$$\text{Word offset} = \lceil \log_2 \text{Block size} \rceil$$

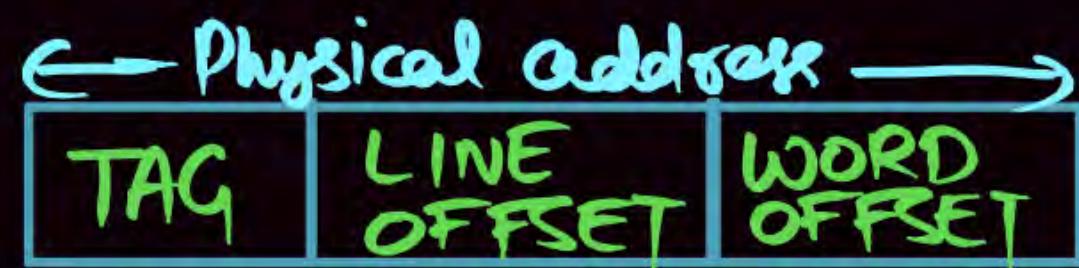
$$\# \text{Cache Lines} = \frac{\text{CM size}}{\text{Block size}}$$

$$L.O = \log_2 \# \text{LINES.}$$

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

Direct Mapping

In Direct Mapped Cache Physical address is interpreted as :



Q.) Consider a Direct Mapping if the size of Cache memory is 512KB & Main Memory 512MB & Cache line size (Block) is 64KB then calculate the number of bit required for

- (i) P.A
- (ii) TAG
- (iii) L.O
- (iv) W.O
- (v) #LINES
- (vi) TAG Memory Size.

$$\#LINES = \frac{Cache\ Size}{Block\ Size}$$

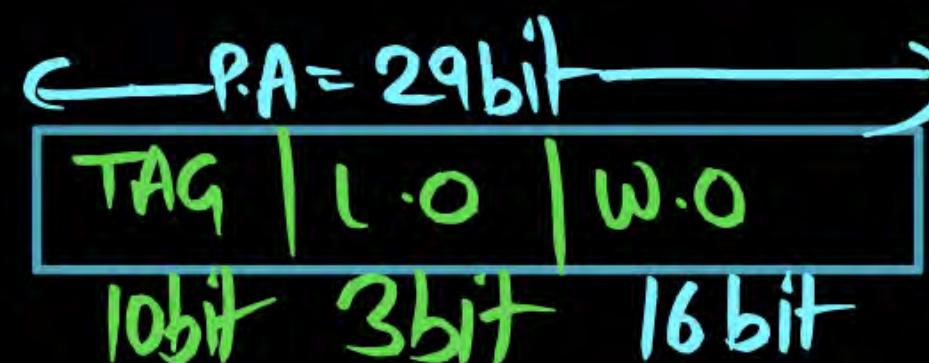
$$\#LINES = \frac{512KB}{64KB}$$

$$= 2^9 B$$

$$= \frac{2^9 B}{2^{16} B}$$

$$MM\ Size = 512MB \Rightarrow 2^{29} \text{ Byte} \Rightarrow P.A = 29 \text{ bit}$$

$$Block\ Size = 64KB \Rightarrow 2^{16} \text{ Byte} \Rightarrow W.O = 16 \text{ bit}$$



$$TAG = 29 - (3 + 16) = 10bit$$

$$Tag\ Memory\ Size = \#LINES \times Tag\ bits$$

$$\Rightarrow 8 \times 10$$

$$= 80 \text{ bits Ans}$$

#LINE = 2³

8 lines

L.O = 3 bit

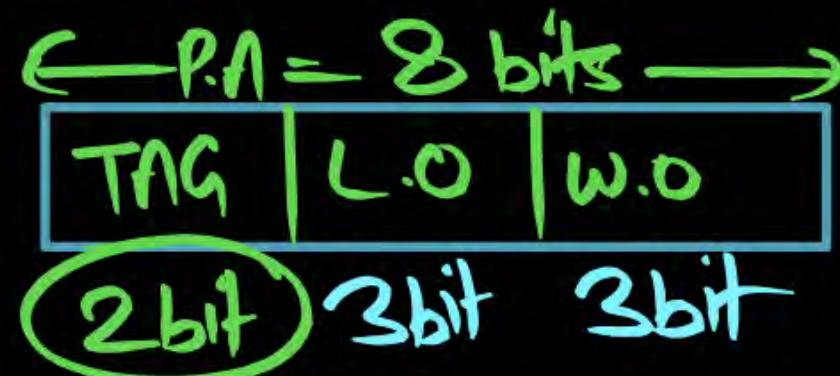
Q.

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

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

memory size? $\hookrightarrow 8\text{bit}$

$$\# \text{LINES} = \frac{\text{Cache Size}}{\text{Block Size}} \Rightarrow \frac{64B}{8B} = \frac{2^6B}{2^3B} = 2^3 = 8 \text{ Lines}$$



$$\begin{aligned} \text{TAG} &= 8 - (3+3) \\ &= 2 \text{ bit} \end{aligned}$$

$\# \text{LINES} \times \text{Tag bit}$

$$\text{Tag memory size} = 8 \times 2 \text{ bit}$$

$$= 16 \text{ bits} \quad \underline{\text{Ans}}$$

L.O = 3 bit

P
W

3bit

Q.3

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

2^7 \bowtie
 2^6 B
P
W

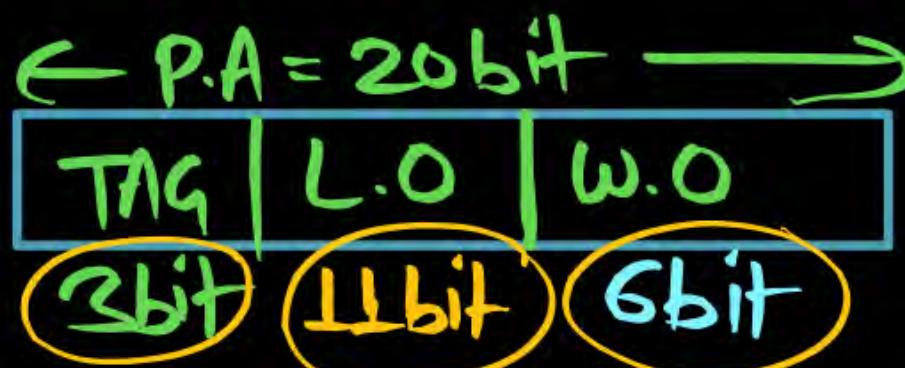
Byte. Main Memory is 1 MB then what is the line number of

$$A: 10 \Rightarrow 10^{10}$$

$$B: 11 : 10^{11}$$

$$C: 12 : 11^{10}$$

physical address $(ABCDE)_{16}$?



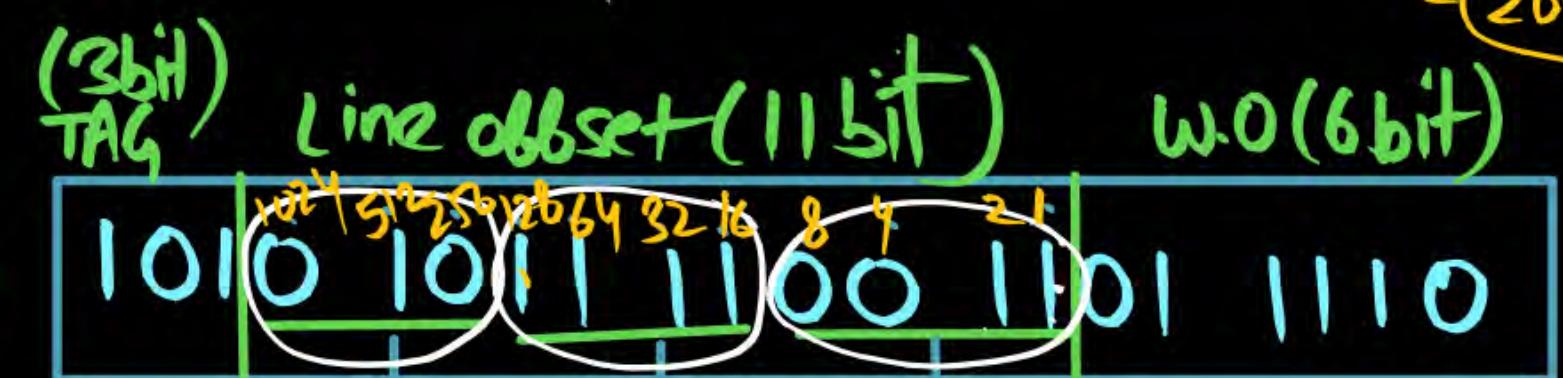
$$\begin{aligned} \text{TAG} &= 20 - (11 + 6) \\ &= 3 \text{ bit} \end{aligned}$$

$$\begin{aligned} &\therefore \\ &\left(\frac{512}{243} \right)_{10} \\ &\quad + 55 \end{aligned}$$

$ABCDE \Rightarrow$

$$\begin{aligned} \text{MM} &= 1\text{ MB} = 2^{20}\text{ B} \Rightarrow \text{P.A.} = 20 \text{ bit} \\ \text{LineSize} &= 64\text{ B} \Rightarrow 2^6\text{ B} \Rightarrow \text{WD} = 6 \text{ bit} \end{aligned}$$

$$\# \text{LINES} = \frac{\text{CacheSize}}{\text{Block Size}} \Rightarrow \frac{128\text{ kB}}{64\text{ B}} \Rightarrow \frac{2^7\text{ B}}{2^6\text{ B}} = \frac{2^1}{2^0} = 2^1 \text{ Lines} = 2048 \text{ Lines}$$



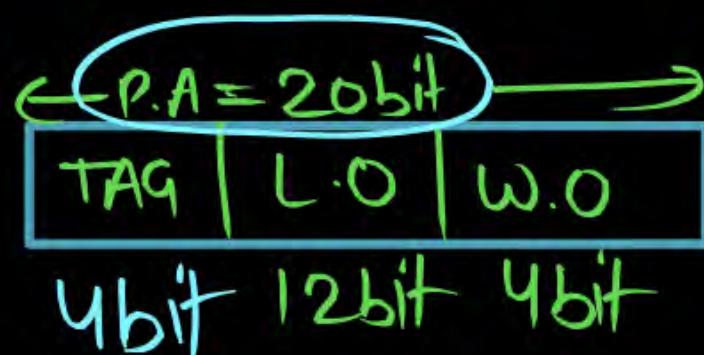
Q.4

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 $(E201F)_{16}$? P W B

#t_{INT} = 2
L.O = 12bit

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

[GATE-2015]



$(E201F)_{16}$

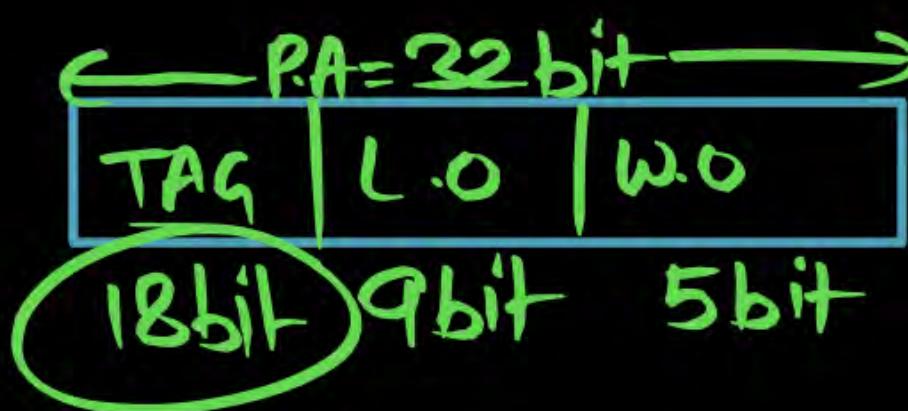
TAG(4bit)	L.O(12bit)	W.O(4bit)
E	201	F

TAG	L.O(12bit)	W.O(4bit)
1110	0010 0000 0001	1111

Q5.

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 $\log_2 512 = 9$ bit

P
W



Ans (18)

$$\begin{aligned} \text{TAG} &= 32 - (9 + 5) \\ &= 18 \text{ bit} \end{aligned}$$

NAT

Q.6

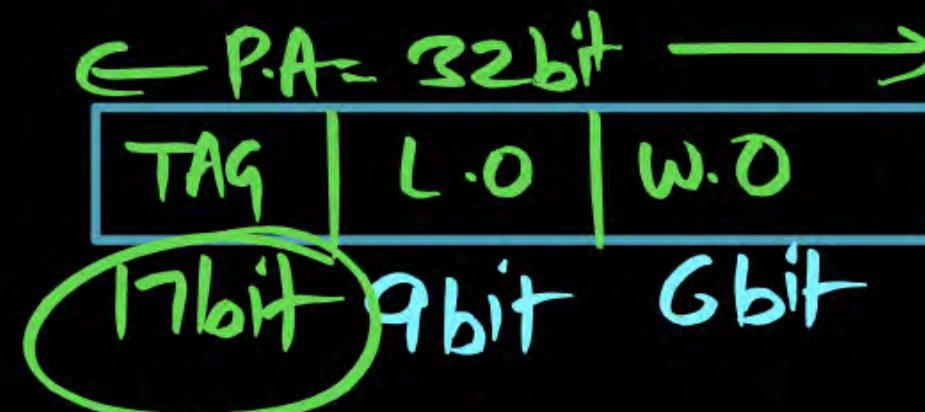
P
W

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 KB = 2^{10} bytes), and each cache block is of size 64 bytes.

R

$\hookrightarrow \text{W.O} = 6 \text{ bit}$

The size of the tag field is 17 bits. Ans



$$\begin{aligned}\text{TAG} &= 32 - (9 + 6) \\ &= 17 \text{ bit}\end{aligned}$$

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

$$\# \text{LINES} = \frac{\text{CmSize}}{\text{Block Size}} = \frac{32 \text{ kB}}{64 \text{ B}} \cdot \frac{2^{15} \text{ B}}{2^6 \text{ B}} = 2^9 \text{ lines}$$

Ans (17)

$\hookrightarrow \text{L.O} = 9 \text{ bit}$

Q.F

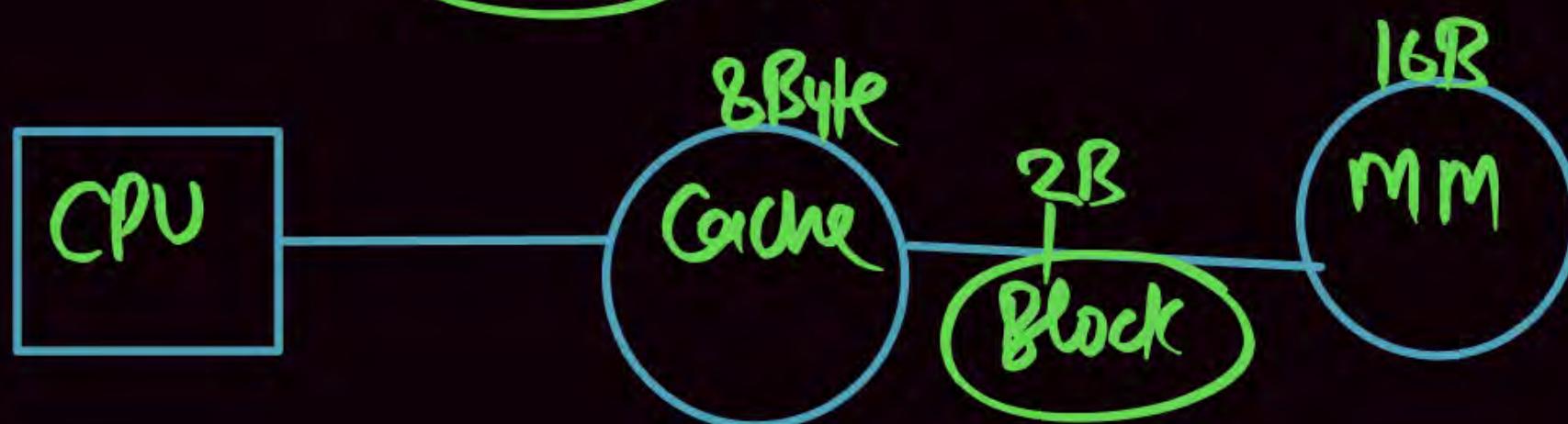
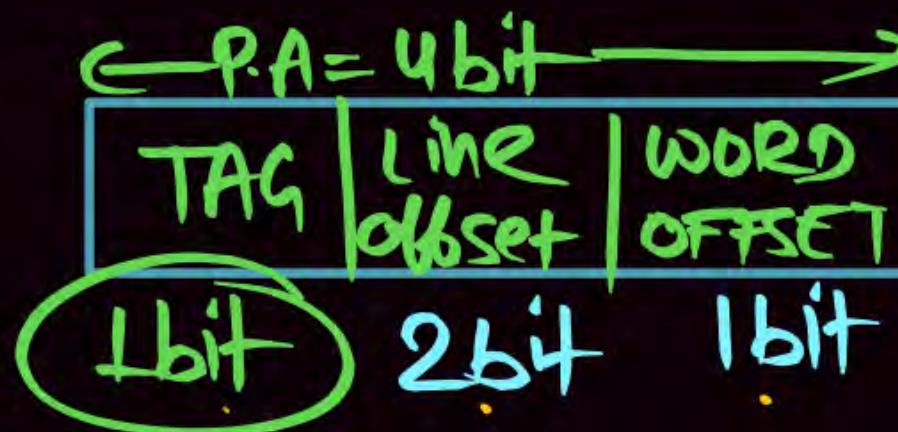
MM Size = 16Byte, CM Size = 8Byte, Block Size = 2Byte
2Byte

#bits TAG, LO, W.O

② Tag Memory Size ?

↓
wordbit

Soln



$$\# \text{Lines} = \frac{\text{CM Size}}{\text{Block Size}} = \frac{8}{2} = 4$$

Tag memory size = #lines × Tag bits

$$\Rightarrow 4 \times 1 \\ = 4 \text{ bits Ans}$$

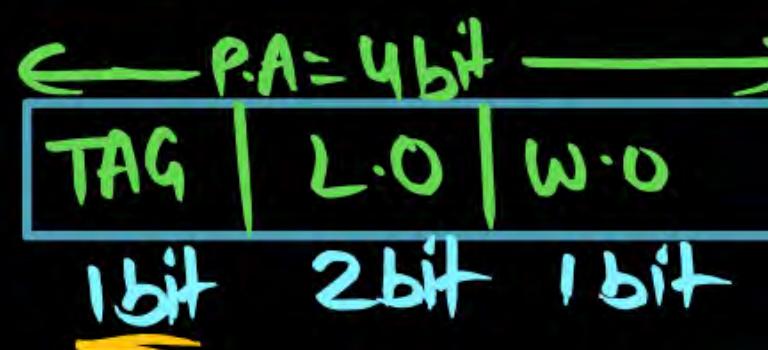


Memory Organization

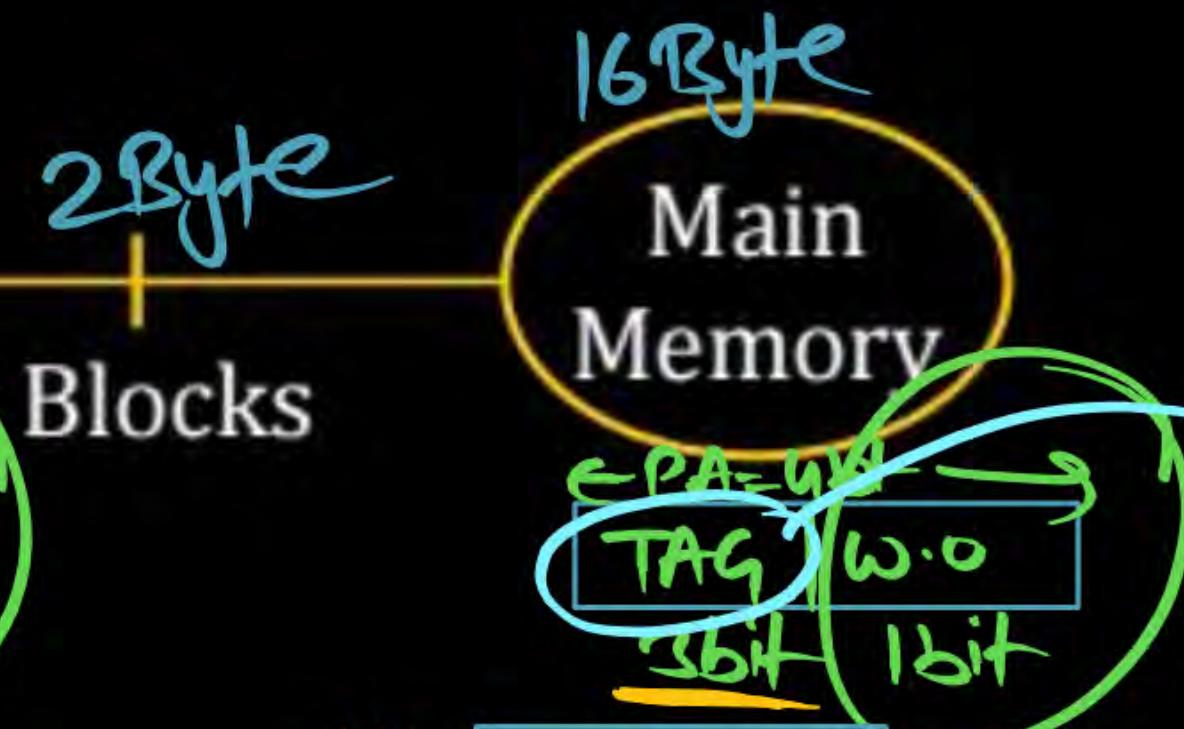
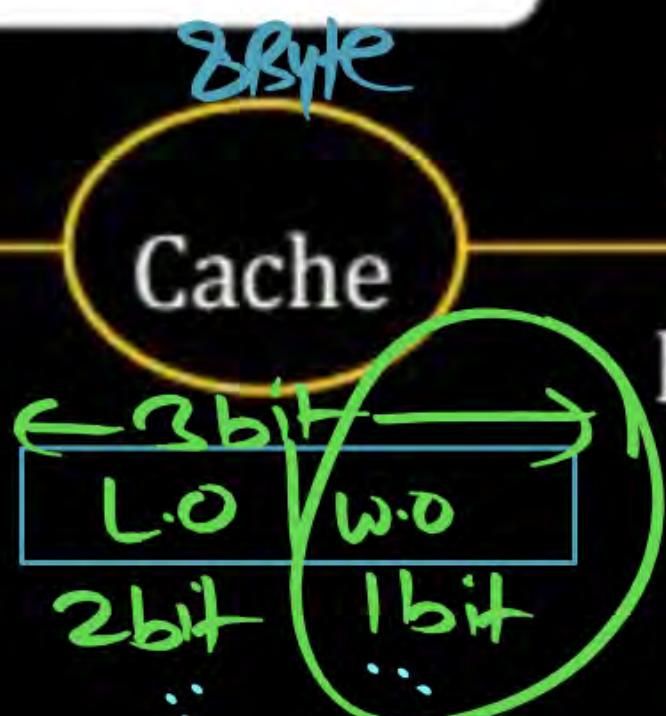
Q.7



Cache Controller

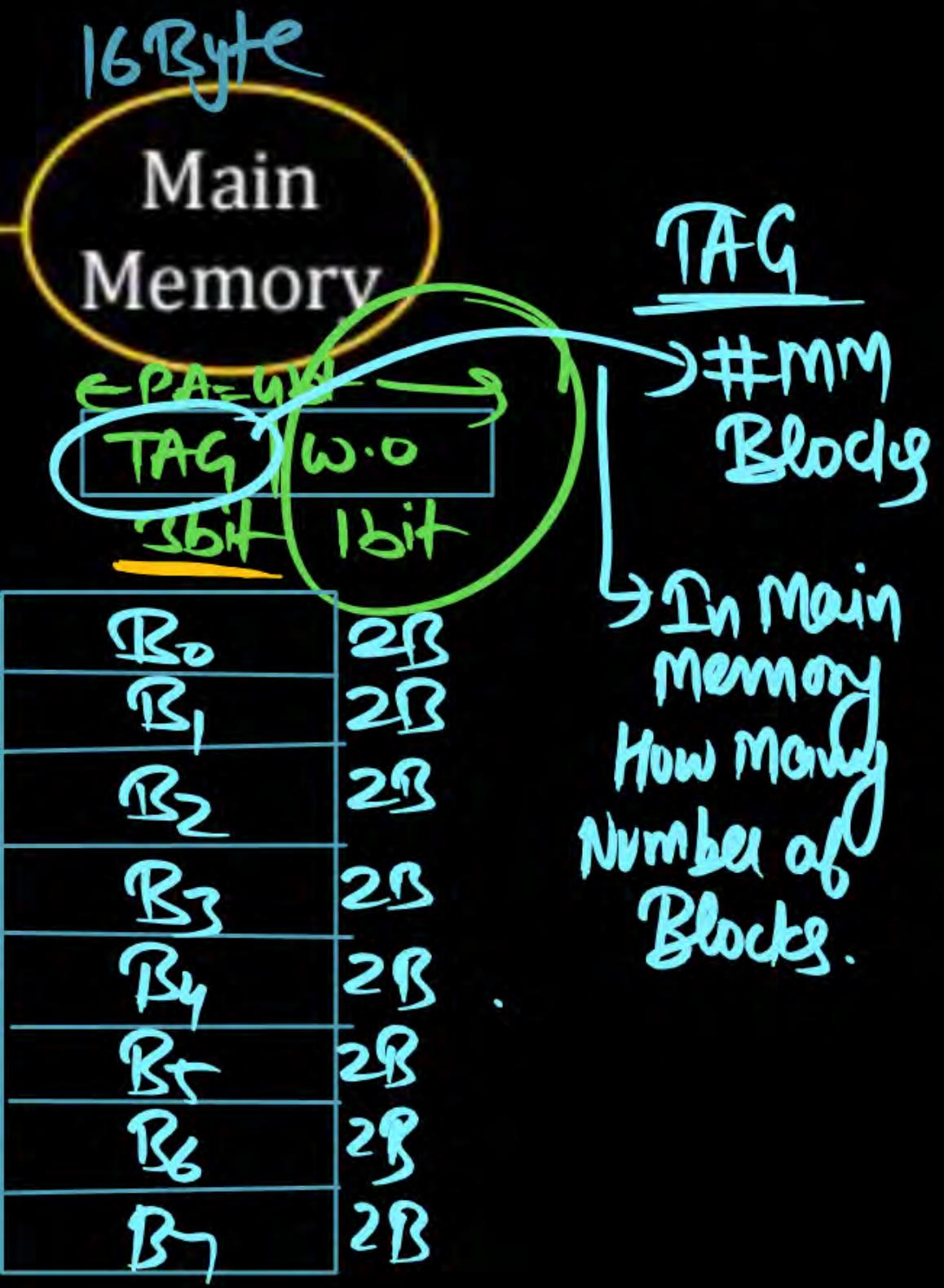
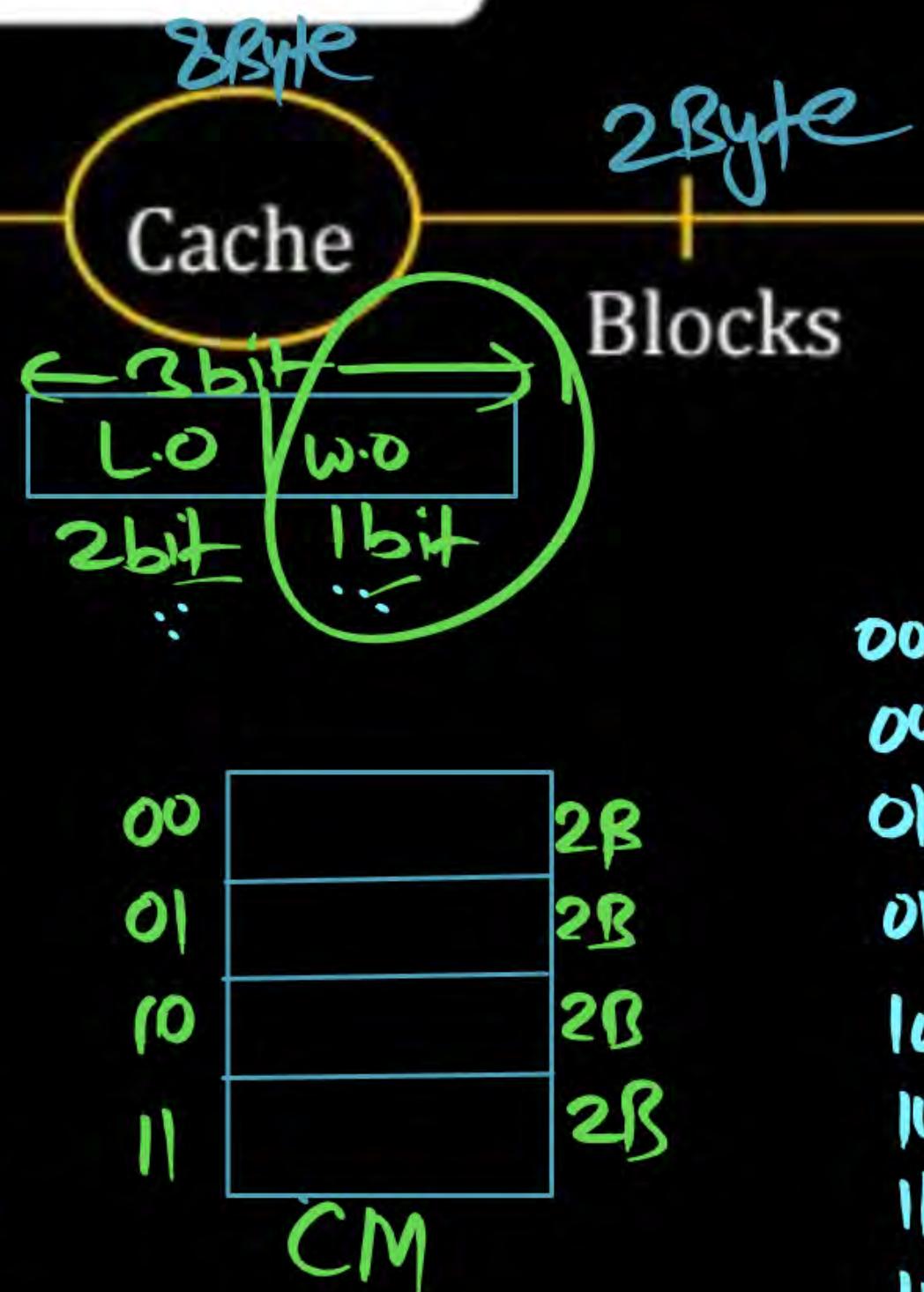
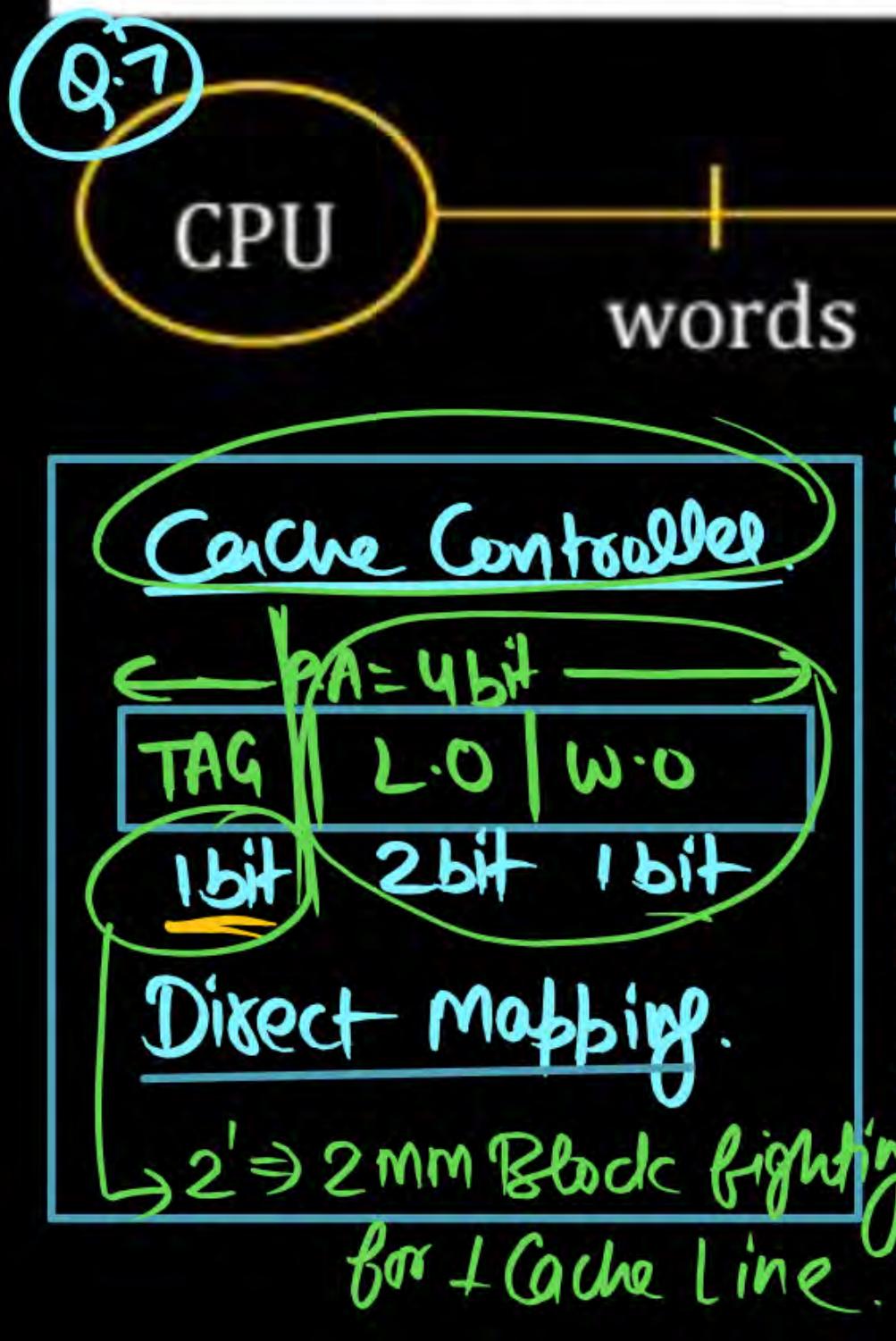


Direct Mapping



In Main
Memory
How Many
Number of
Blocks.

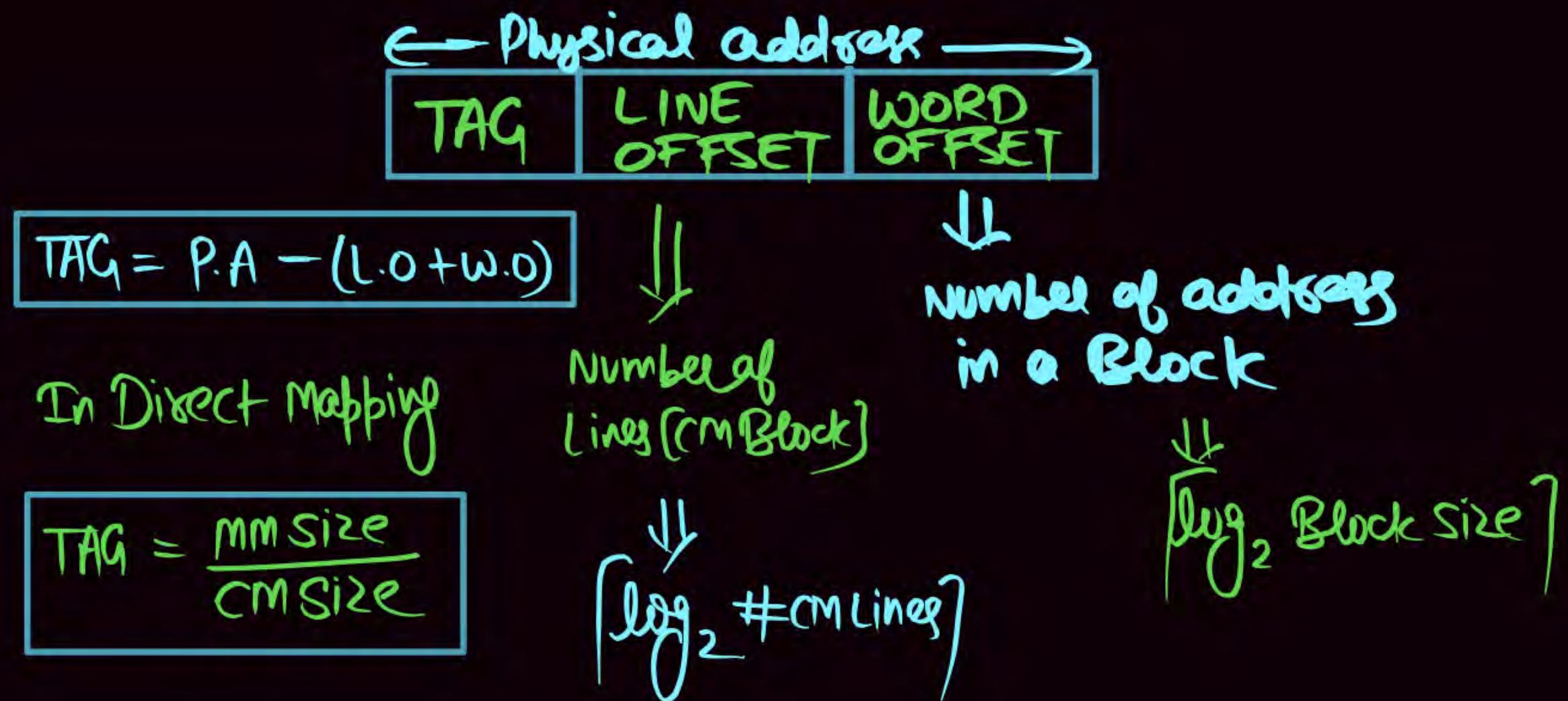
Memory Organization



Direct-Mapping

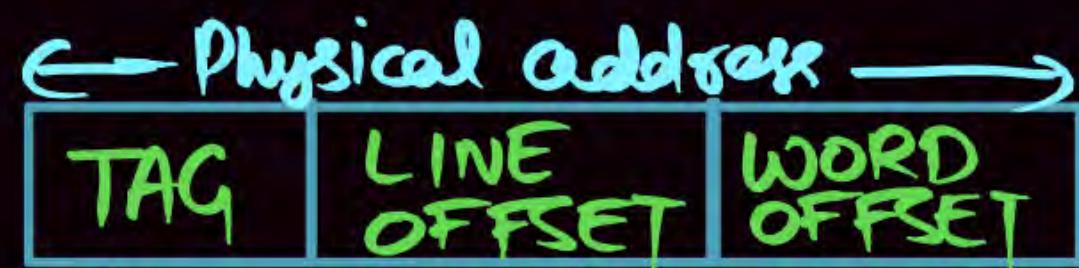
Direct Mapping

In Direct Mapped Cache Physical address is interpreted as :



Direct Mapping

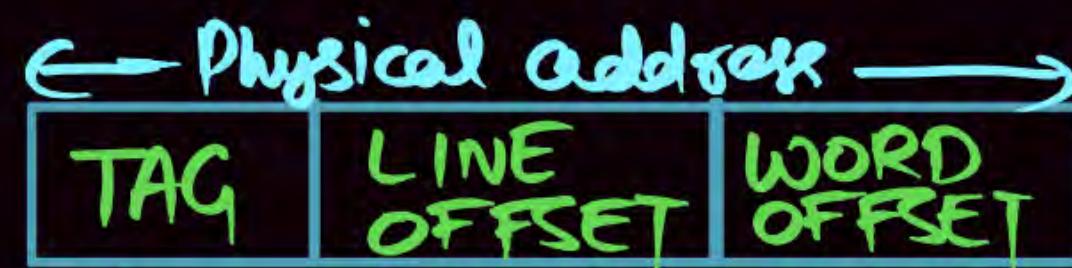
In Direct Mapped Cache Physical address is interpreted as :



Number of MM
Blocks are possible
in One Cache Line.
But Not at same
time.

Direct Mapping

In Direct Mapped Cache Physical address is interpreted as :



Number of MM
Blocks are possible
in One Cache Line.

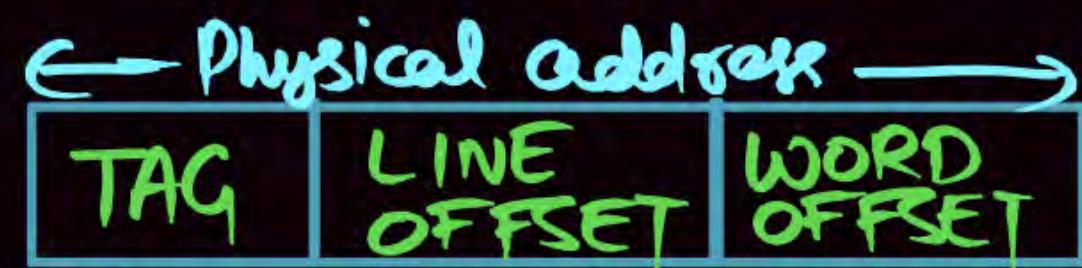
But Not at same
time.

③ $\Rightarrow \boxed{\text{TAG} = 2 \text{ bit}} \Rightarrow 2^2 \Rightarrow 4 \text{ Main Memory Block}$
are Mapped to One Cache Line.

4 MM Block are Competing for
One Cache Memory Line

Direct Mapping

In Direct Mapped Cache Physical address is interpreted as :



Number of MM
Blocks are possible
in One Cache Line.

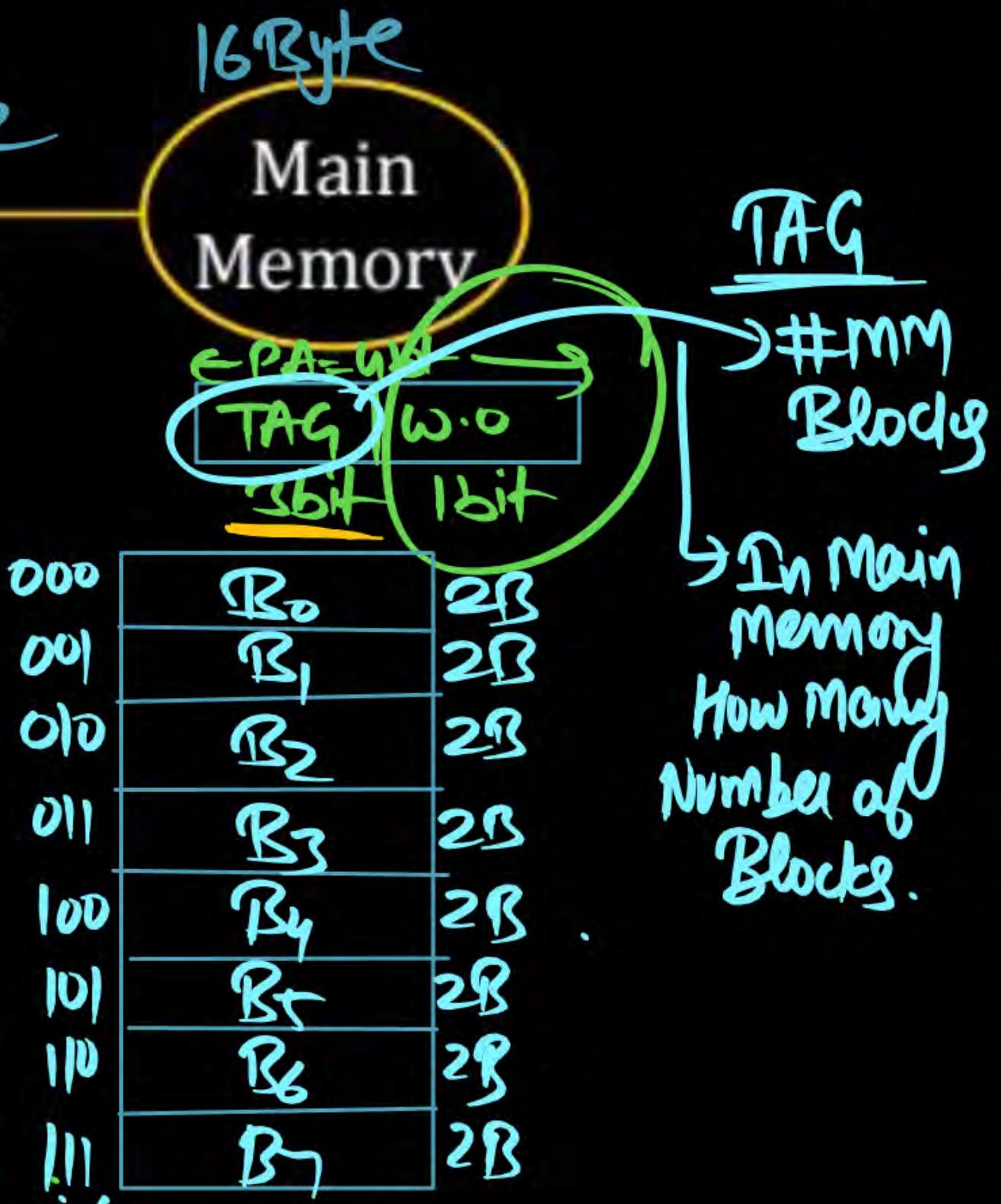
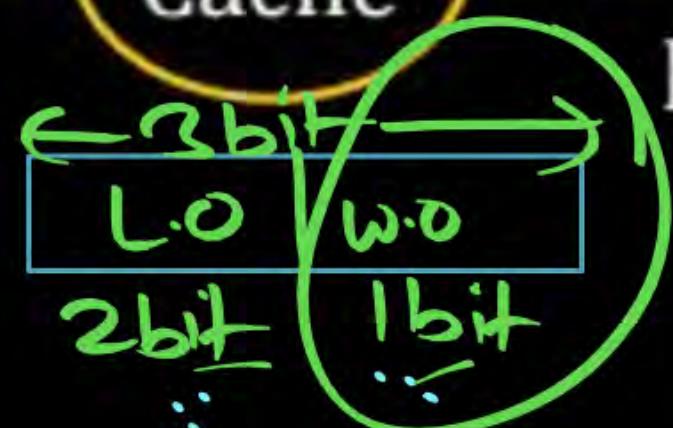
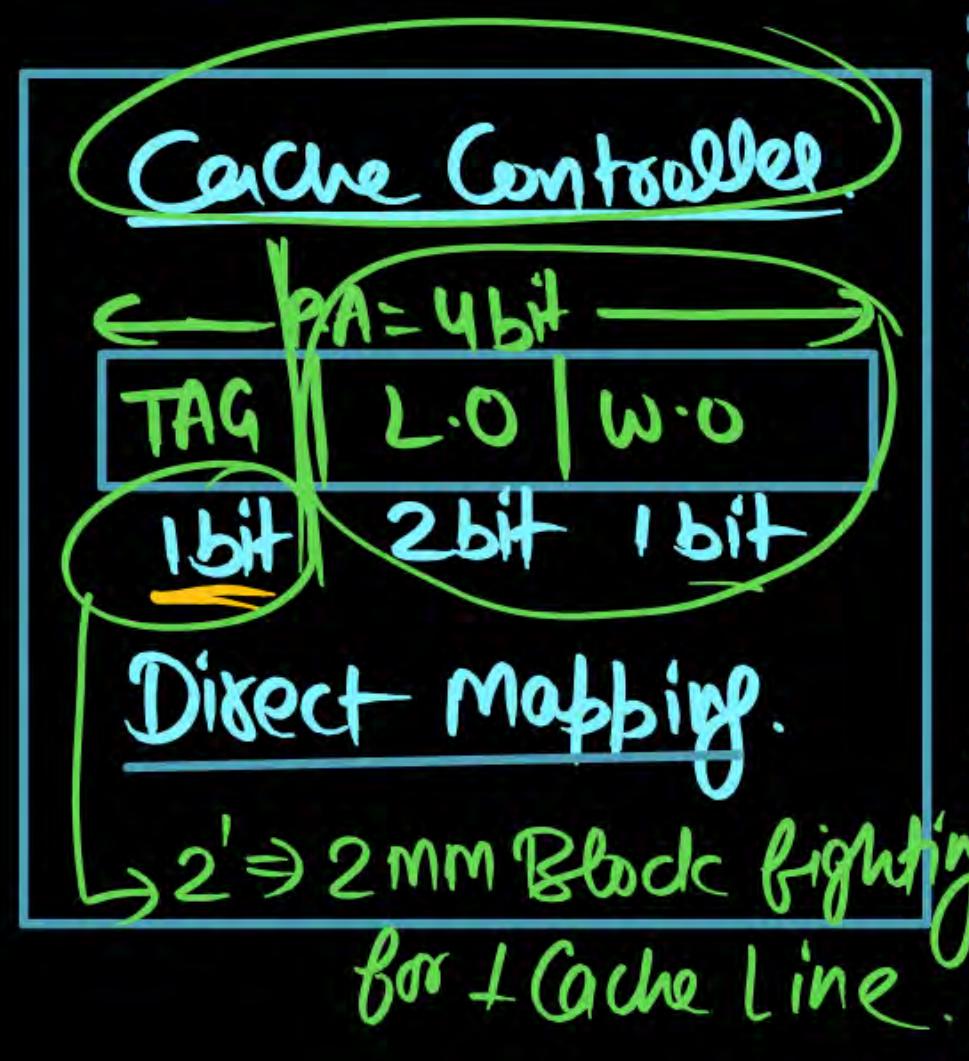
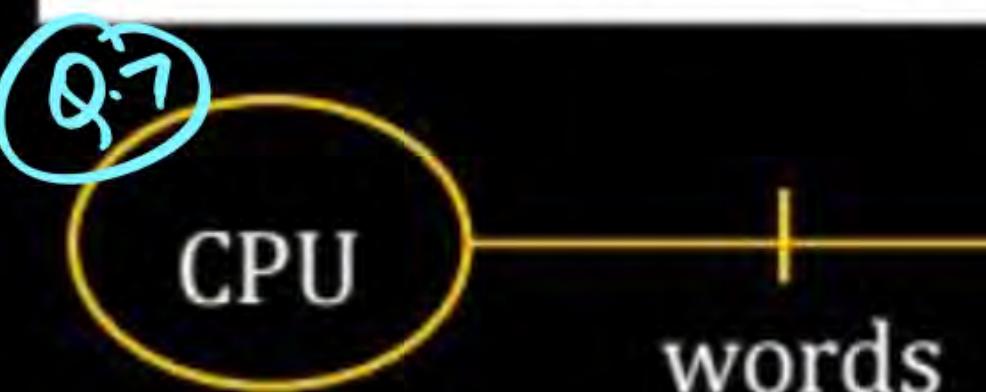
But Not at same
time.

③ $\Rightarrow \boxed{\text{TAG} = L \text{ bit}} \Rightarrow 2^L \Rightarrow Q$ Main Memory Block
are Mapped to One Cache Line.

Q MM Block are Competing for
One Cache Memory Line

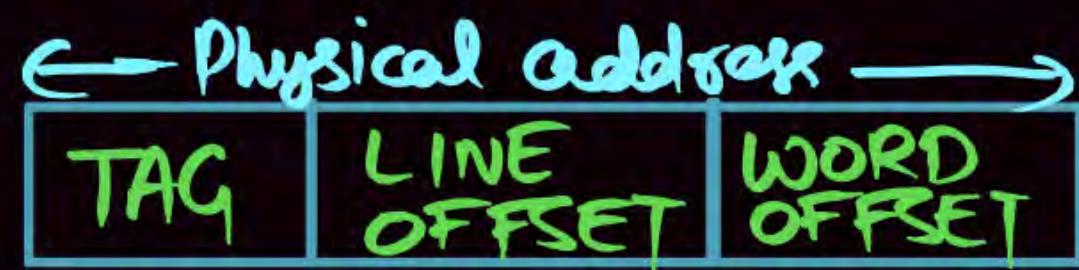
Memory Organization

Q7



Direct Mapping

In Direct Mapped Cache Physical address is interpreted as :



Direct Mapping:

In Direct Mapping an Mapping function is Used to transfer(Copying) the Data from Main memory to Cache memory

Mapping function : $K \bmod N = i$ or
K: MM Block No | MM Request
N: Number of CM Lines
i: Cache Line Number
[0 to N-1]

Cache Address = MM Block Number (MM Request) $\bmod \#CM$ Lines.

Direct Mapping:

$$k \bmod N = j$$

Wrong 'K' Main Memory Block are transferred
into $k \bmod N$ Cache Memory Line Number.

Ex Assume we have 4 Cache Lines in CM. & MM Block No LL
(B_{LL})

(i) Then MM Block No LL (B_{LL}) are Mapped
(transferred) into $\frac{k \bmod 'N'}{4} \rightarrow$ CM Line No '3'

(ii) If MM Block = 7 (B_7) then
MM Block Number 7 (B_7) are placed into $7 \bmod 4 =$ CM Line Number '3'

i.e. Here Block No B_{LL} & Block No B_7 are Mapped (Placed) into Cache Line Number 3.
But any one block present in Cache Line No 3.

$$K \bmod N = i$$

$$K \bmod 4 = j$$



$$\# CM Lines [N] = 4$$

$$B_{11} \bmod(1\cdot)4 = \frac{CM Line No}{3}$$

$$B_7 \bmod(1\cdot)4 = 3$$

1) Direct Mapping

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

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

(Or)

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

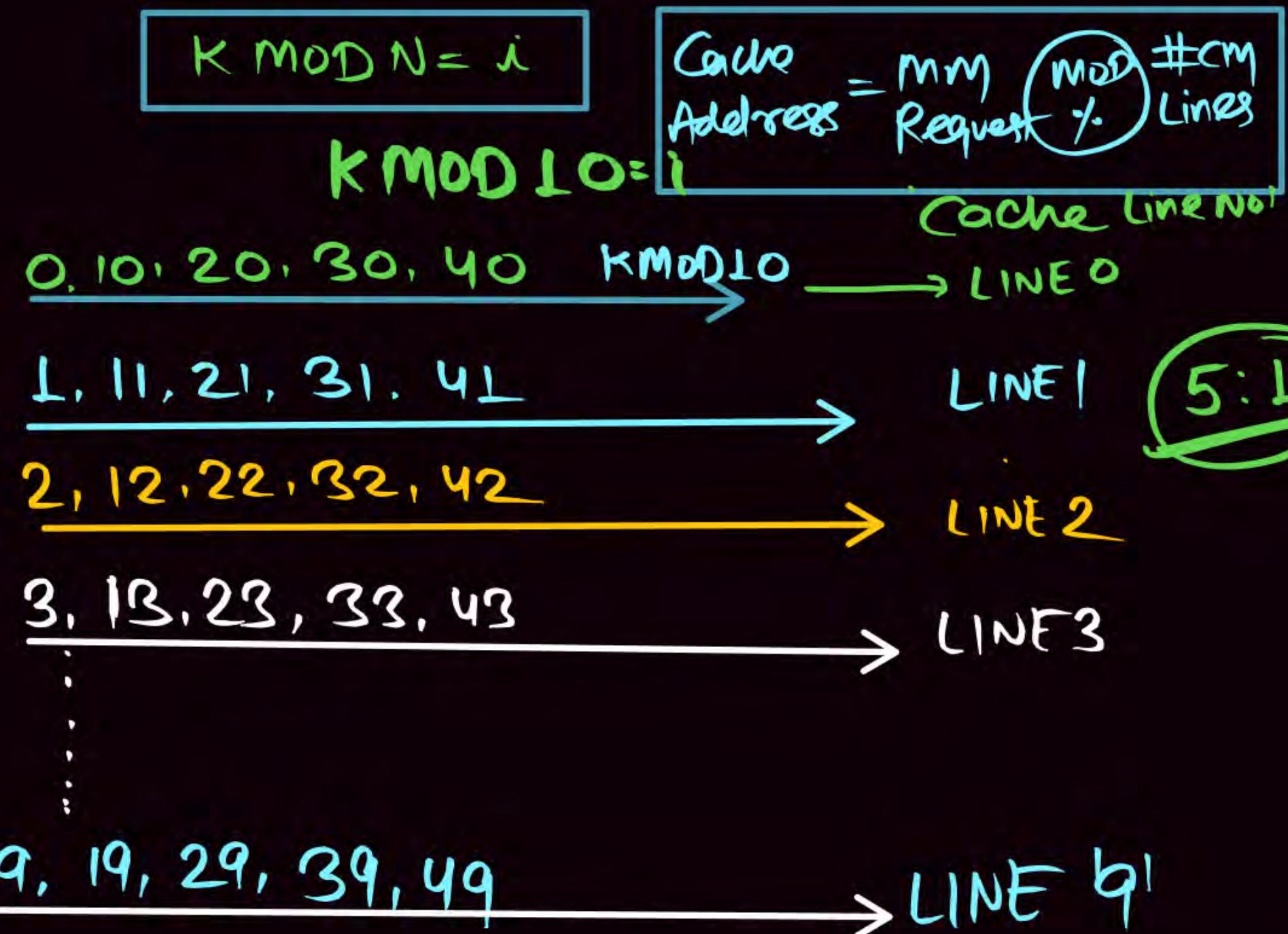
K: MM Block No.

N: # of Cache Line

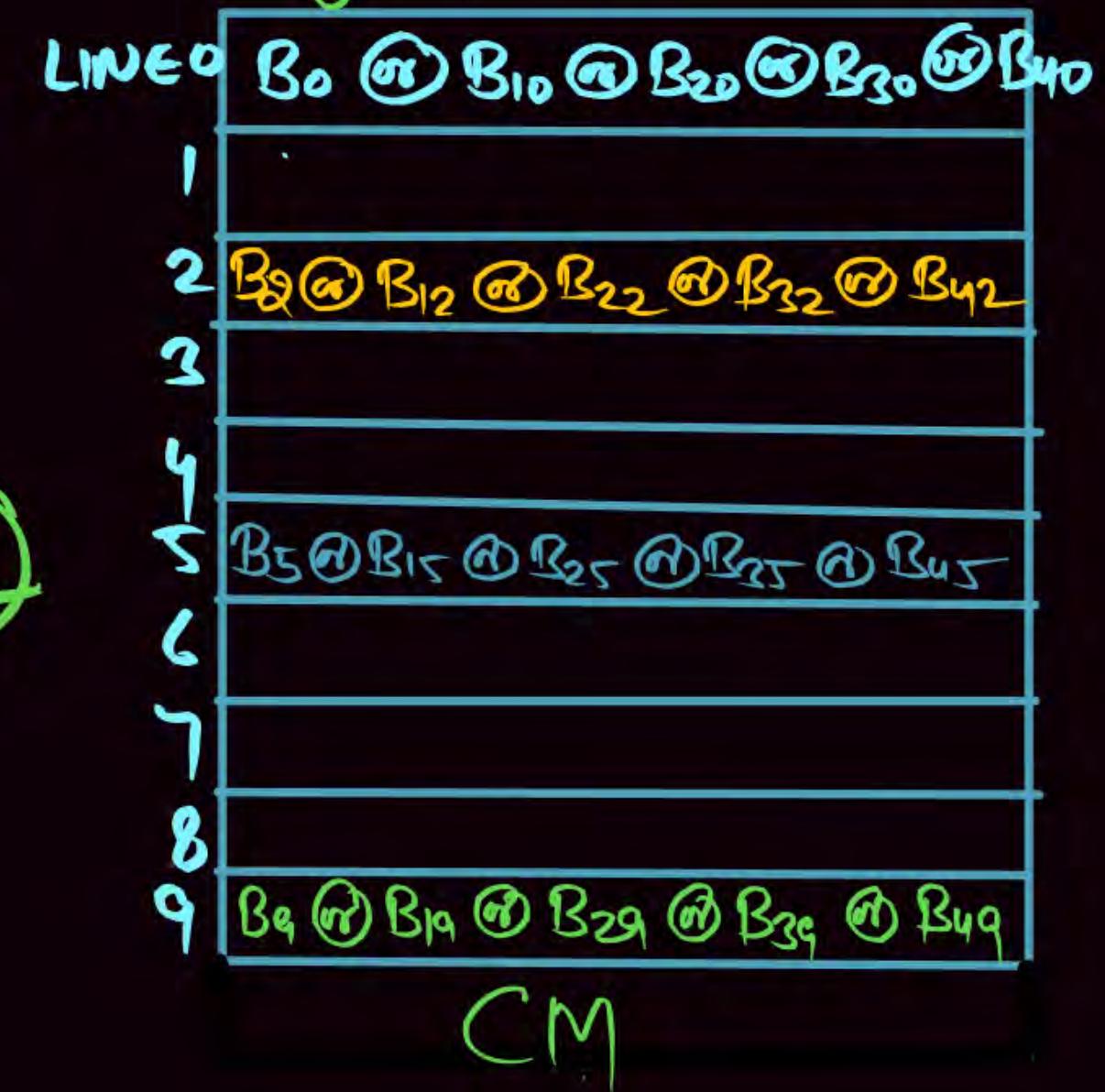
i: CM Line Number

Direct Mapping:

Q MM = 50 Blocks, CM = 10 Lines.



5 MM Block are fitting
For the One CM Line But
Only one at a time.



Direct Mapping:

Ex MM = 50 Blocks, CM = 10 Lines.

$$TAG = \frac{MM}{CM} = \frac{50}{10}$$

$$TAG = 5$$

5 : L

5 MM Blocks are fighting (competing)
for the One Cache Line
(mapped)

But Out of 5 Only ^{Any} One MM Block
Present in the Cache at a time.

Direct Mapping:

$N = 4 \times (2^3)$

$K \bmod N = i$

$K \bmod 4 = l$

Ex 2 # MM Blocks = 16, # CM Lines = 4

(B_0)	$0 \bmod 4 = 0$	$B_8 \bmod 4 = 0$
(B_1)	$1 \bmod 4 = 1$	$B_9 \bmod 4 = 1$
(B_2)	$2 \bmod 4 = 2$	$B_{10} \bmod 4 = 2$
(B_3)	$3 \bmod 4 = 3$	$B_{11} \bmod 4 = 3$
B_4	$4 \bmod 4 = 0$	$B_{12} \bmod 4 = 0$
B_5	$5 \bmod 4 = 1$	$B_{13} \bmod 4 = 1$
B_6	$6 \bmod 4 = 2$	$B_{14} \bmod 4 = 2$
B_7	$7 \bmod 4 = 3$	$B_{15} \bmod 4 = 3$

LINE 0	$B_0 \oplus B_4 \oplus B_8 \oplus B_{12}$
LINE 1	$B_1 \oplus B_5 \oplus B_9 \oplus B_{13}$
LINE 2	$B_2 \oplus B_6 \oplus B_{10} \oplus B_{14}$
LINE 3	$B_3 \oplus B_7 \oplus B_{11} \oplus B_{15}$

Direct Mapping:

Q2) # MM Blocks = 16, # CM Lines = 4

$$TAG = \frac{16}{4}$$

4:L

$$TAG = 4$$

4 MM Block are Mapped
into One Cache Line But
Only One MM Block can present
in Cache at a time.

Direct Mapping:

(ex)

$$MM = 16 \text{ Blocks} \cdot CM = 4 \text{ Lines.}$$

Block No (B_0, B_4, B_8, B_{12}) Mapped into Cache Line No '0'

Block No (B_1, B_5, B_9, B_{13}) Mapped into Cache Line '1'

Block No $(B_2, B_6, B_{10}, B_{14})$ Mapped into Cache Line '2'

Block No $(B_3, B_7, B_{11}, B_{15})$ Mapped into Cache Line No '3'.

Direct Mapping:

$$N = 4 \times (\# \text{Lines})$$

$$i = (0 \text{ to } 3)$$

(Q3) MM = 8 Blocks. CM = 4 Lines

2:L

$$K \bmod 4 = i$$

CM Line Number

B_0	0	$\bmod 4 = 0$
B_1	1	$\bmod 4 = 1$
B_2	2	$\bmod 4 = 2$
B_3	3	$\bmod 4 = 3$
B_4	4	$\bmod 4 = 0$
B_5	5	$\bmod 4 = 1$
B_6	6	$\bmod 4 = 2$
B_7	7	$\bmod 4 = 3$

LINE 0	$B_0 \oplus B_4$
LINE 1	$B_1 \oplus B_5$
LINE 2	$B_2 \oplus B_6$
LINE 3	$B_3 \oplus B_7$

CM

Direct Mapping:

Q3) MM = 8 Blocks. CM = 4 Lines.

$$TAG = \frac{B}{4}$$

$$\boxed{TAG = 2}$$

TAG = 2 means 2 MM Blocks
are fighting for 1 Cache
Line But Only One at a time.
Can Present in Cache.

Direct Mapping:

B_0 & B_4 are Mapped into Cache Number 0

B_1 & B_5 are Mapped into Cache Number 1

B_2 & B_6 are Mapped into Cache Number 2

B_3 & B_7 are Mapped into Cache Line Number 3



But Only One MM Block Present in Any Cache Line
by Using Mapping function $[k \bmod N]$.

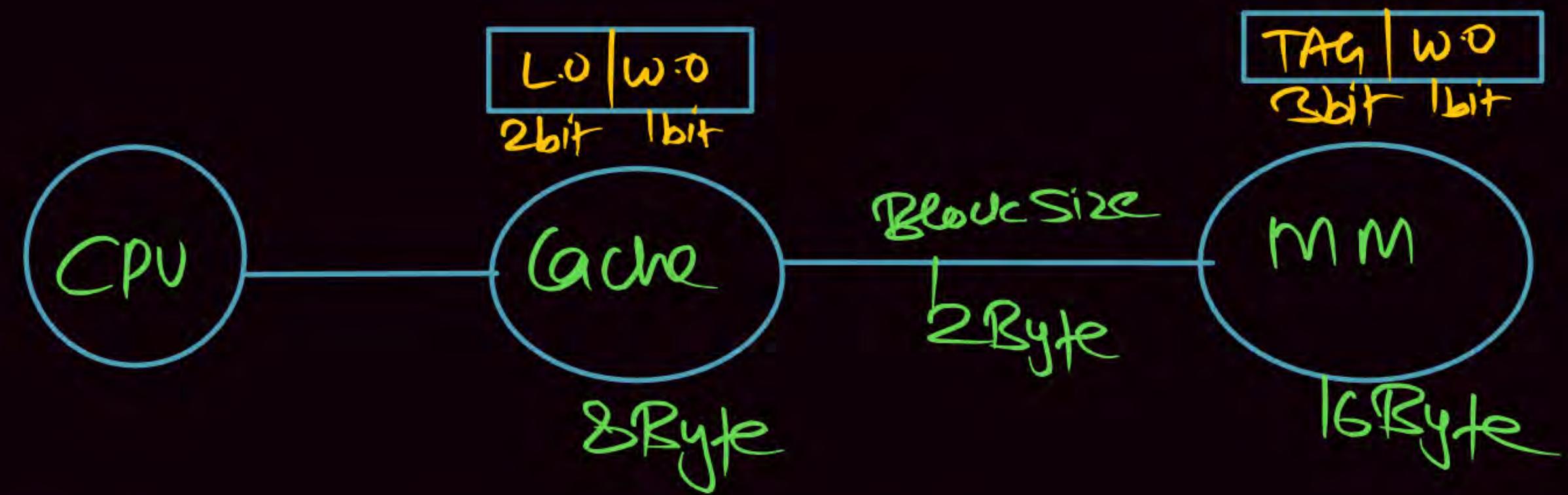
Q) Why Before Mapping we organize the Both Memory.
(MM & CM ?)

Ex) MM Size = 16B, Cache Memory = 8Byte, Block Size = 2Byte

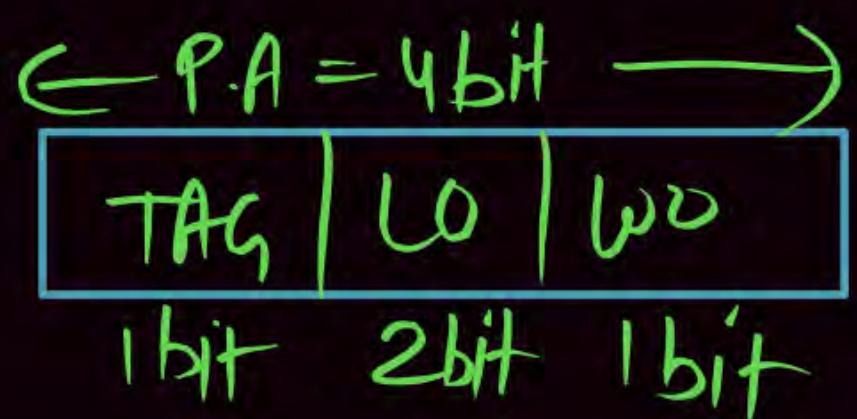
Q. (i) Before Mapping Show the address format of MM & CM ?

Q. (ii) After Mapping Show the Physical address format ?

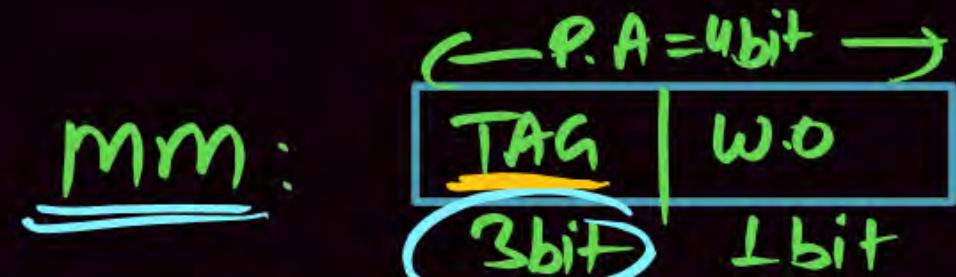
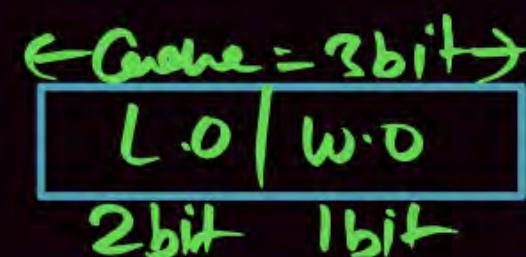
Q



Direct
Mapping:



Before Mapping. Cache



mm:

Here Tag bit Tells the Number of Main Memory Blocks.

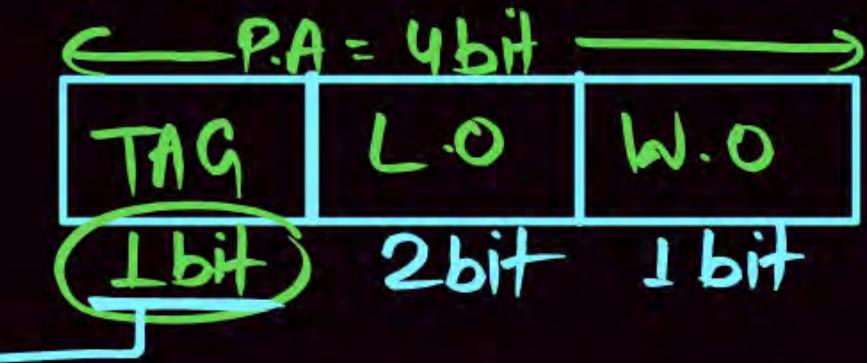
① Before Mapping

② AFTER Mapping

Direct Mapping:

(Role) Work of TAG in Mapping

(i) Here TAG = 1 bit \Rightarrow 2 MM Blocks are Mapped (fighting) for One Cache Line But Only One MM Block can Present in Any Cache Line.



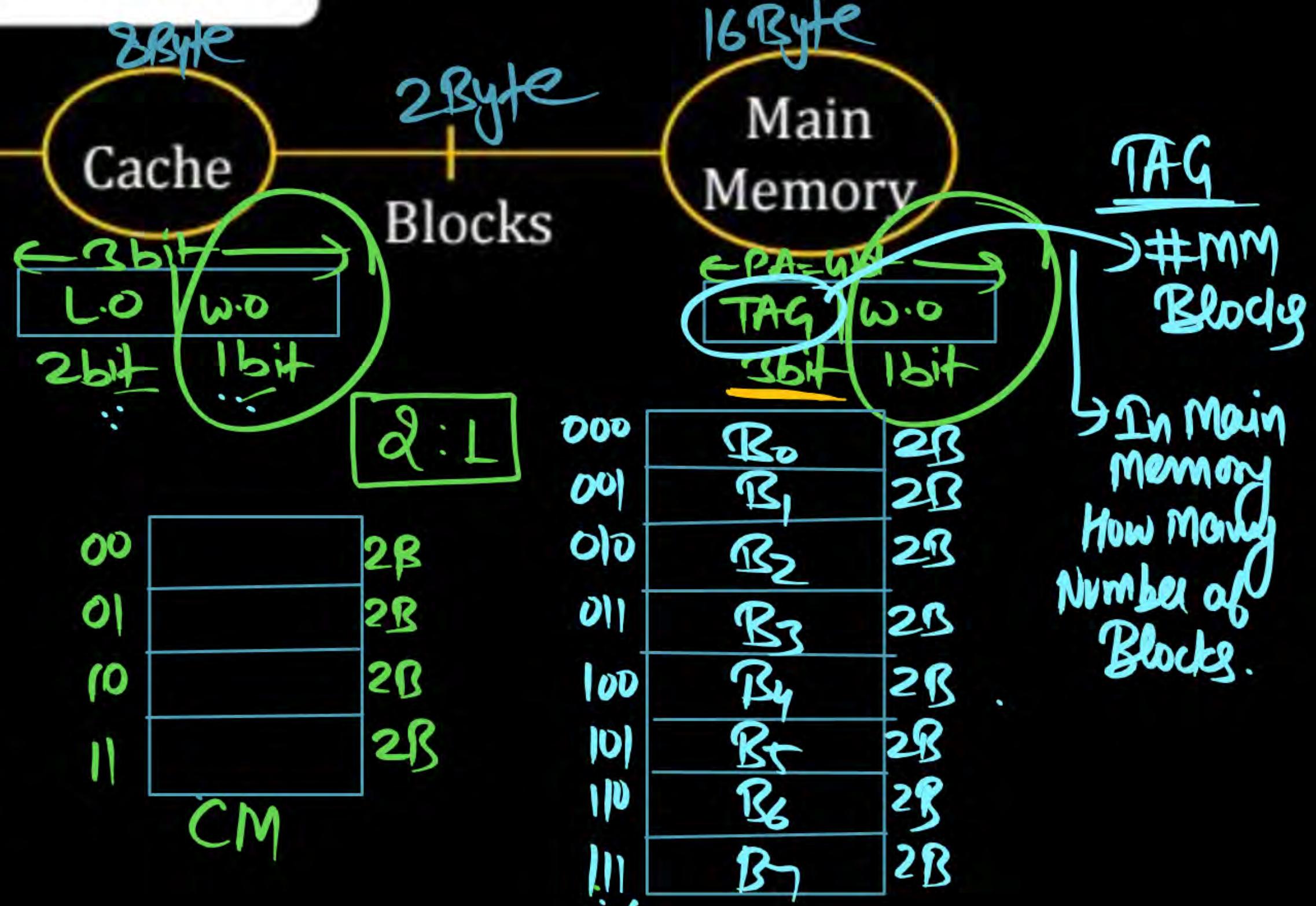
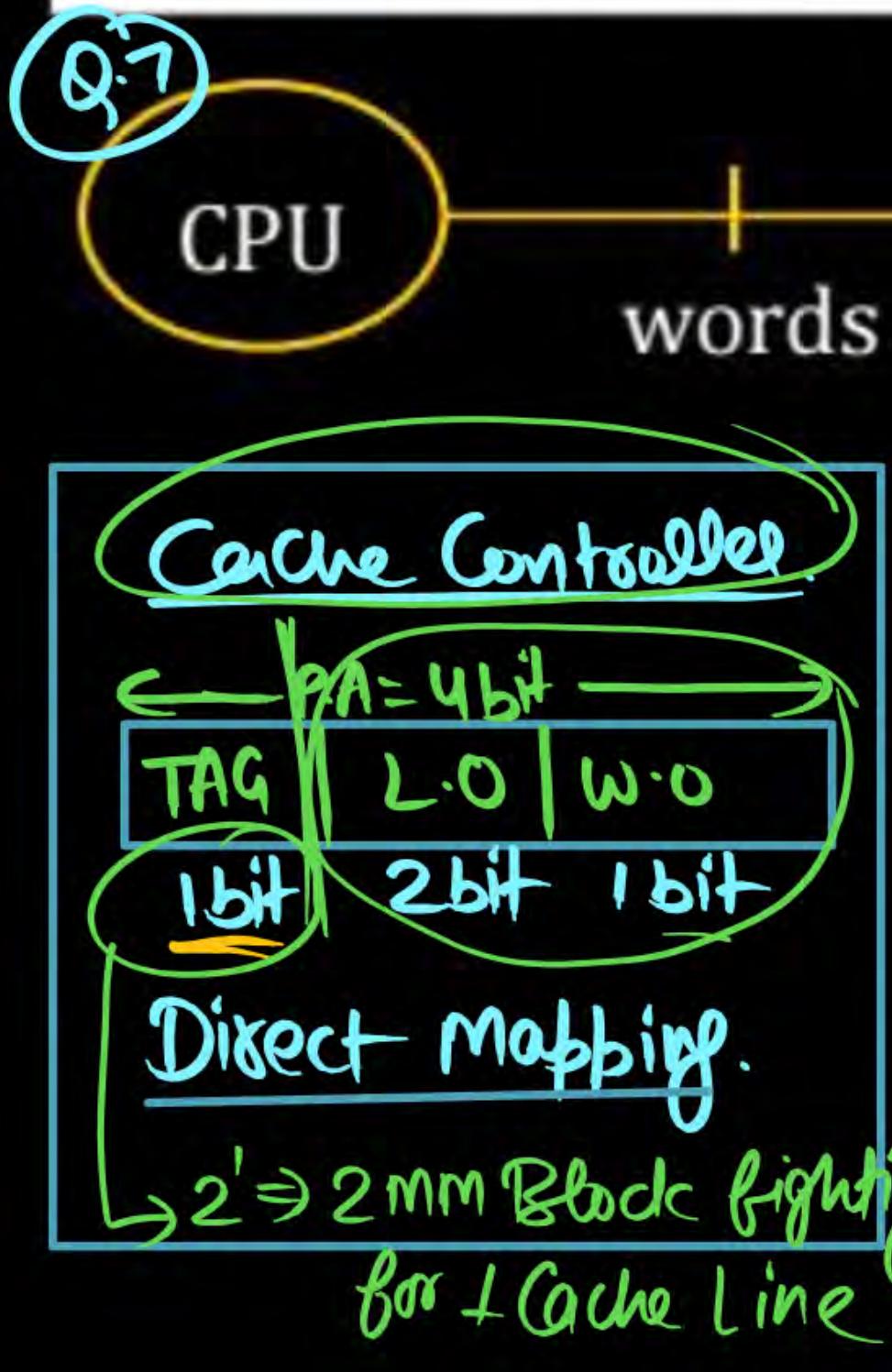
$$\text{TAG} = \frac{\text{MM Size}}{\text{CM Size}} = \frac{16B}{8B} = 2 \quad \boxed{\text{TAG} = 2}$$

OR Tag bit = 1 bit

$$\text{TAG} = \text{P.A} - (\text{L.O} + \text{W.O}) \Rightarrow 4 - (2+1) \quad \boxed{\text{TAG} = 1 \text{ bit}}$$

(ii) This tag bit is used to Check (Identify) Whether there is Cache Hit or Cache Miss.

Memory Organization



Q If the Number of Tag bit for a Cache block [Line No] 'x' is 4 bit, what is the Meaning of this?

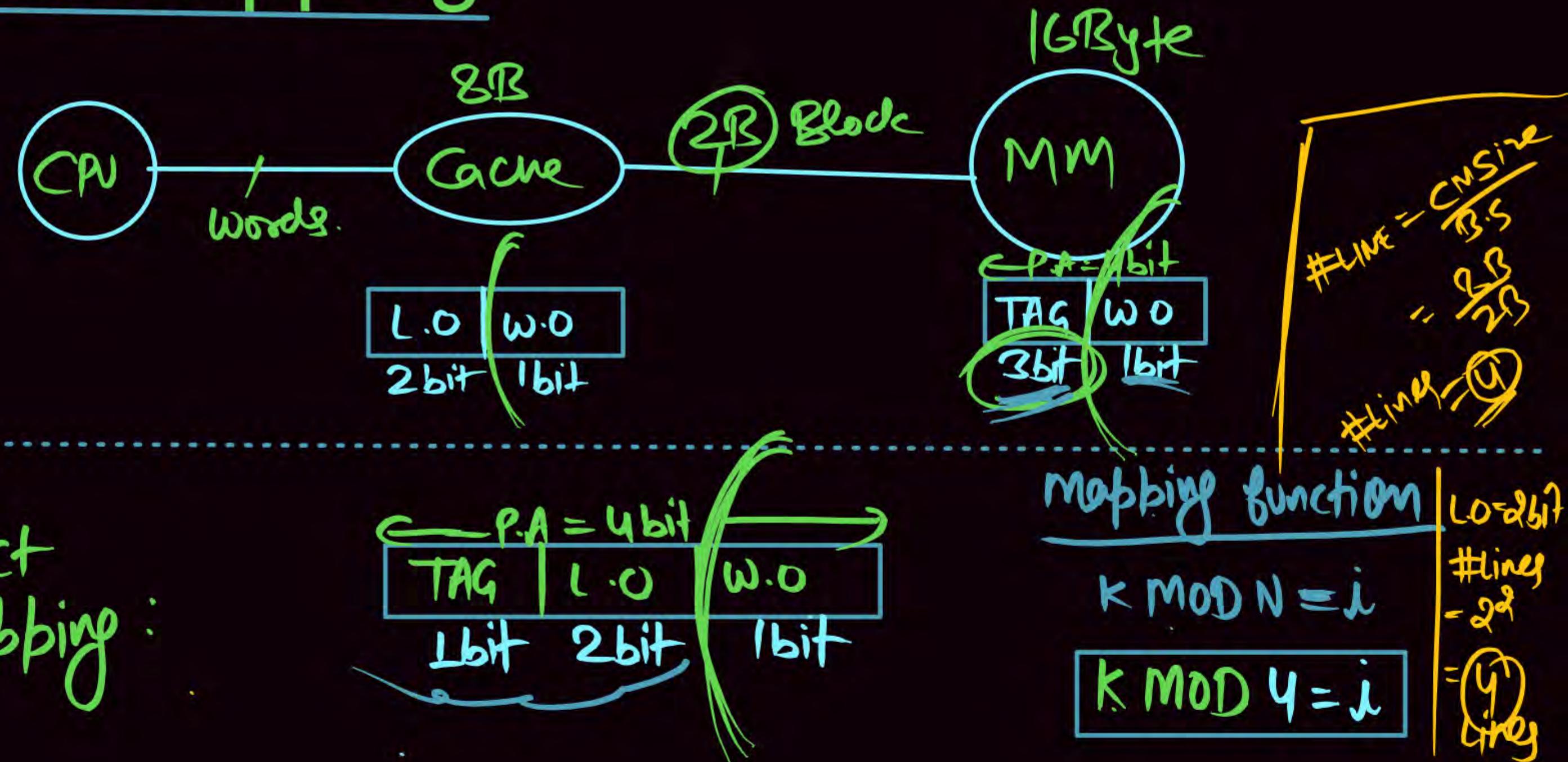
Soln

TAG = 4 bit

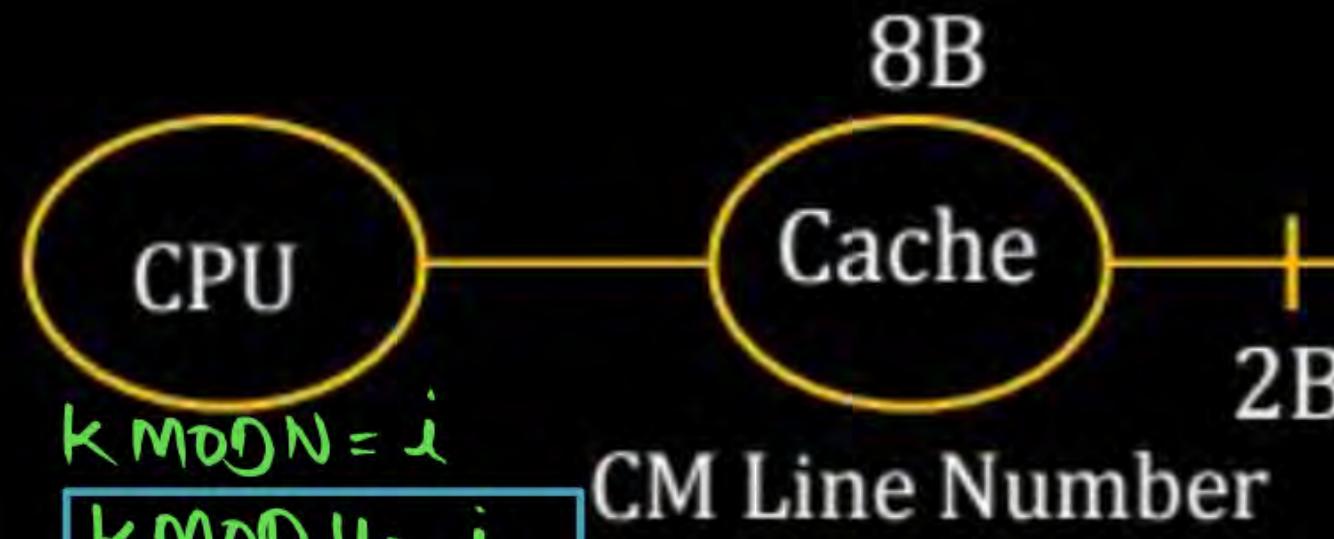
that means 2^4 [16] MM Block are Competing (fighting) for Cache Line Number 'x'.

16 : 1

Direct Mapping:



P
W



$$k \bmod N = i$$

$$k \bmod 4 = j$$

$$0 \bmod 4 = 0$$

$$1 \bmod 4 = 1$$

$$2 \bmod 4 = 2$$

$$3 \bmod 4 = 3$$

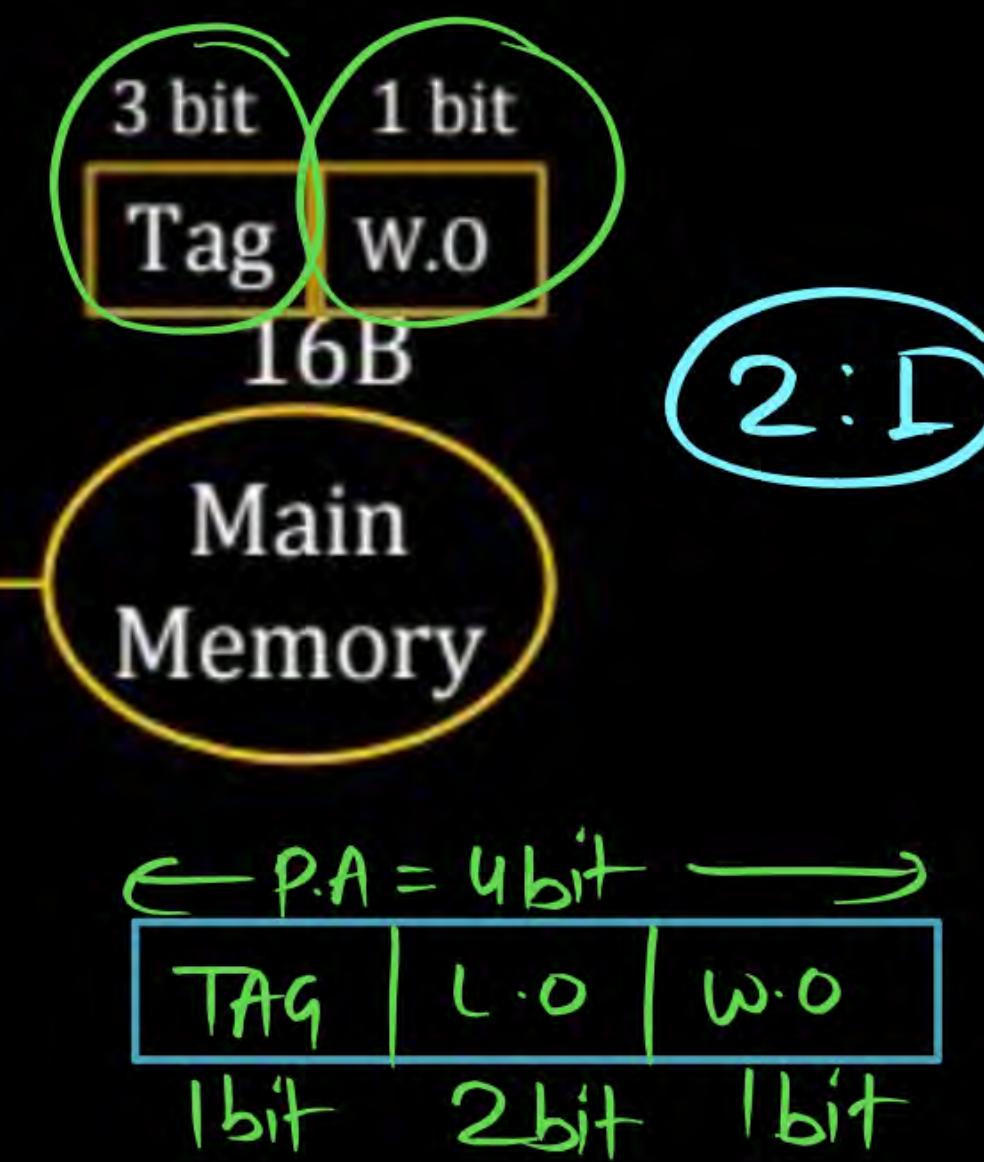
$$4 \bmod 4 = 0$$

$$5 \bmod 4 = 1$$

$$6 \bmod 4 = 2$$

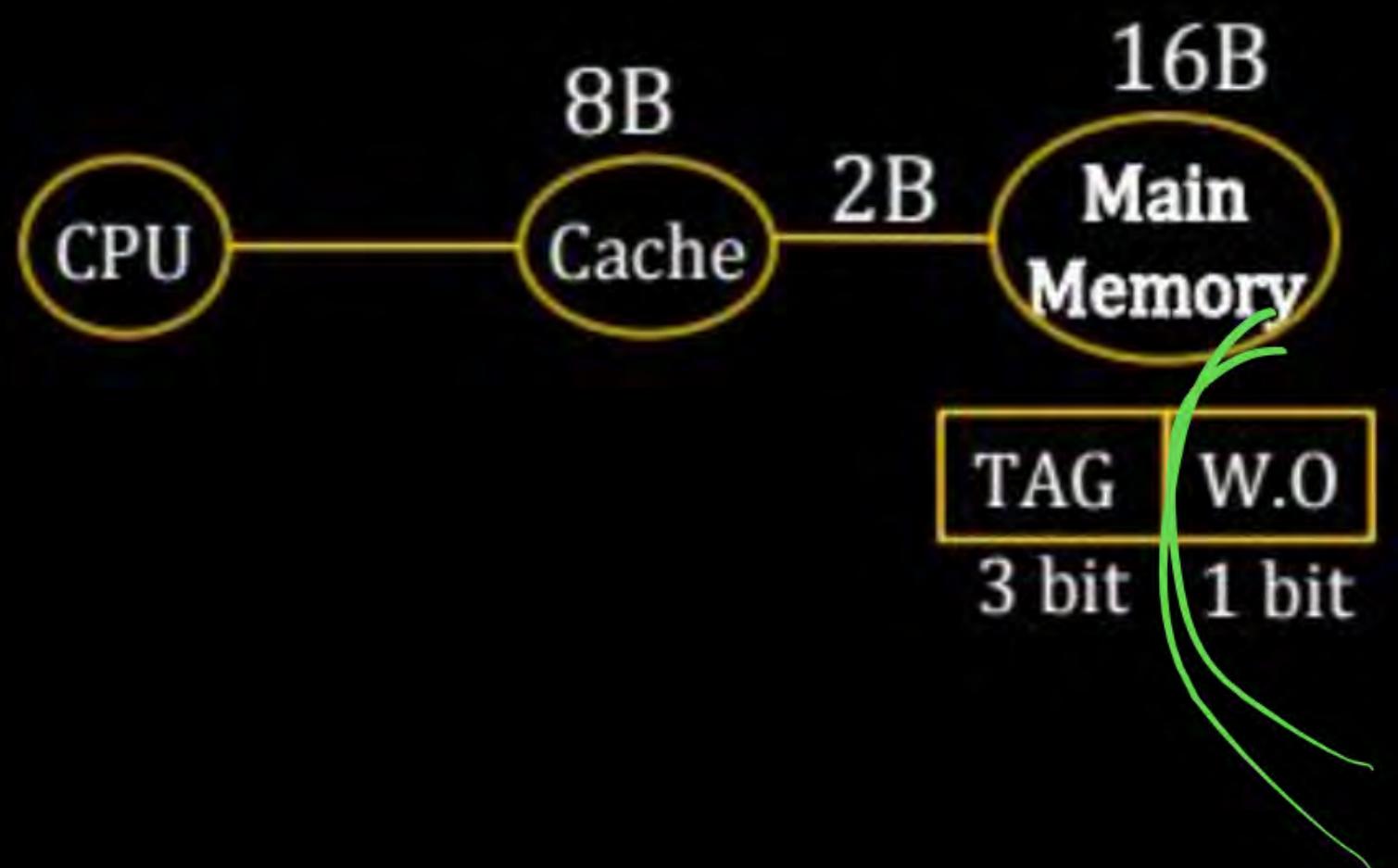
$$7 \bmod 4 = 3$$

- 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

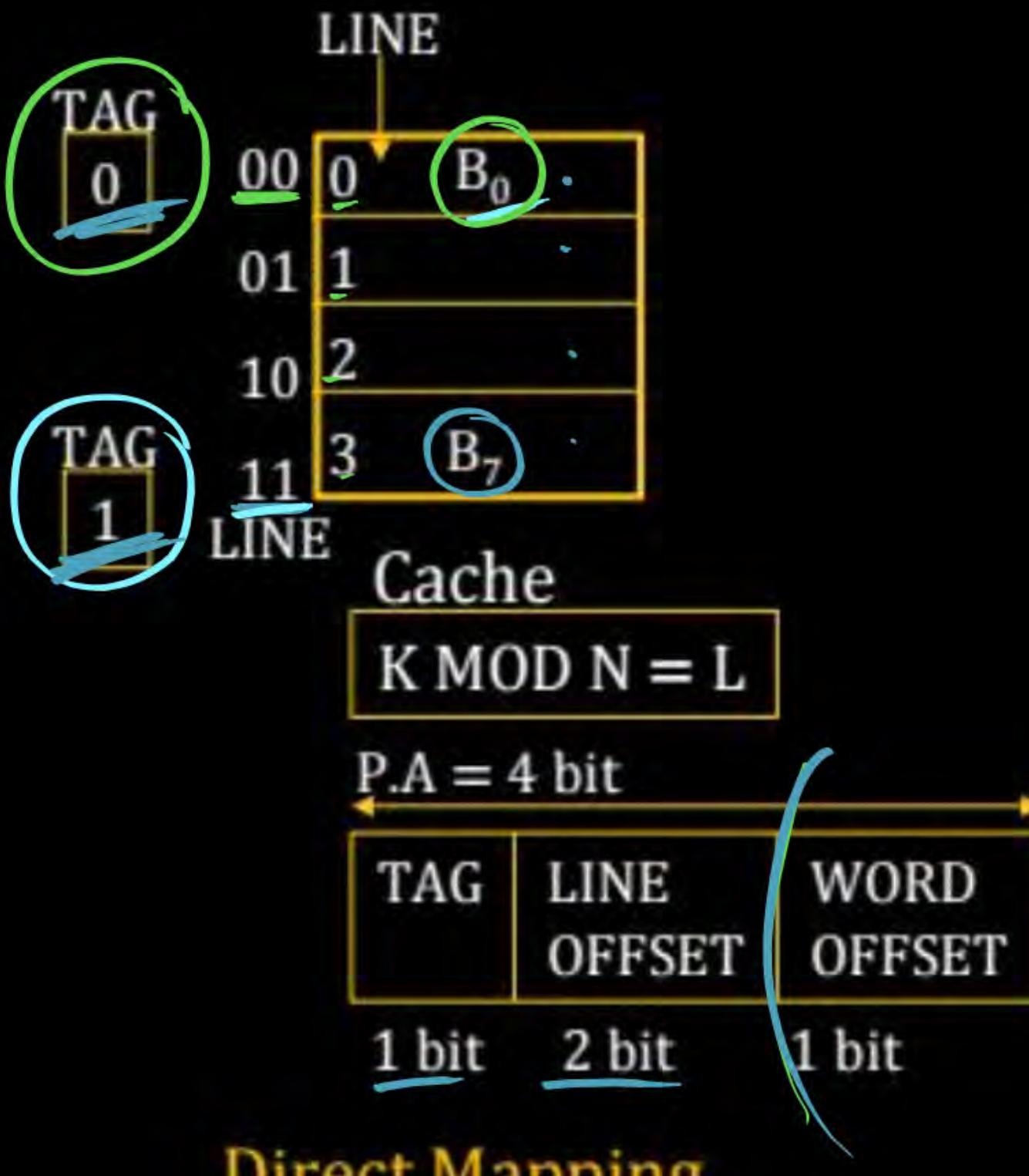


$$k \bmod N = i$$

$$k \bmod 4 = j$$

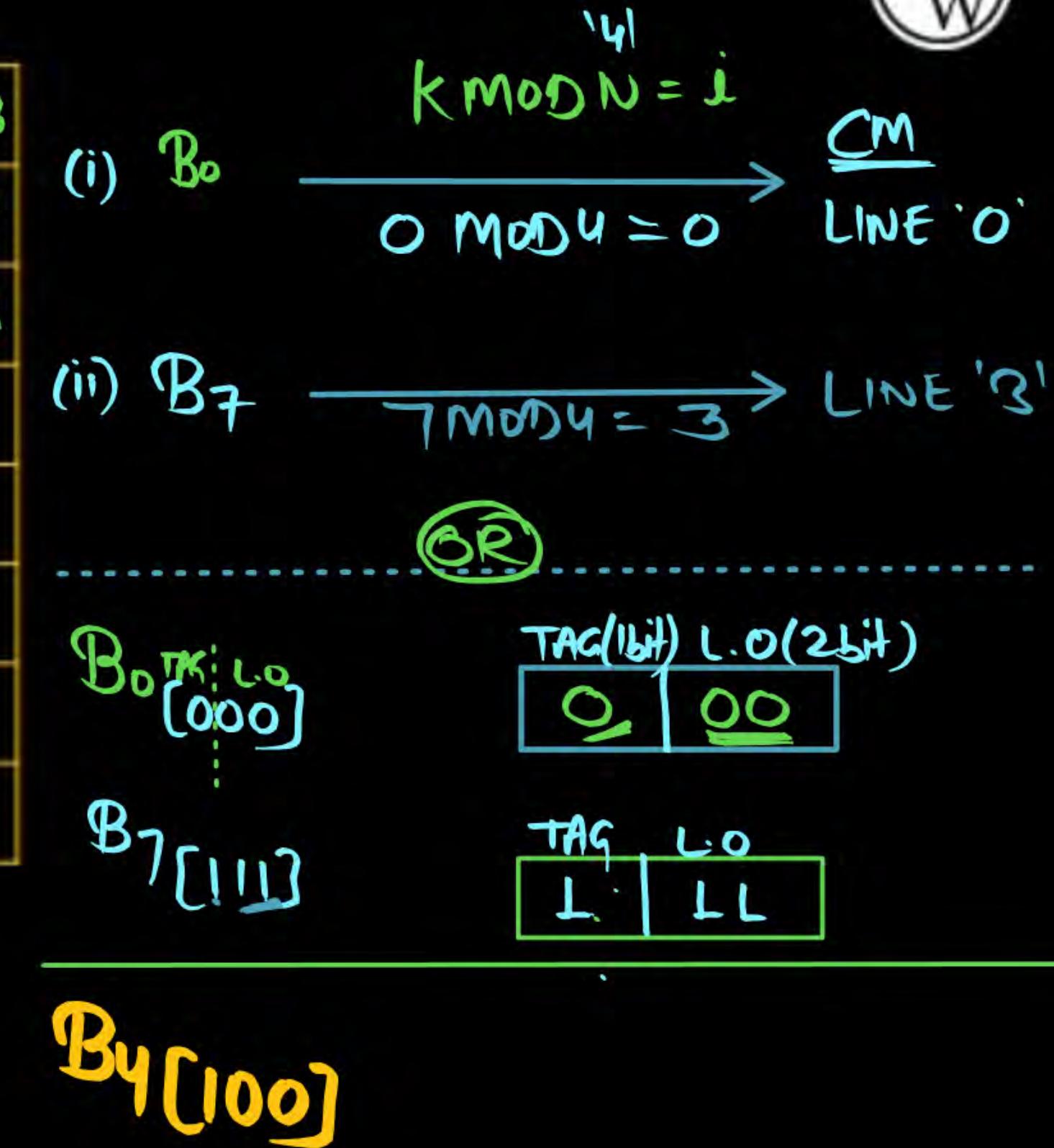


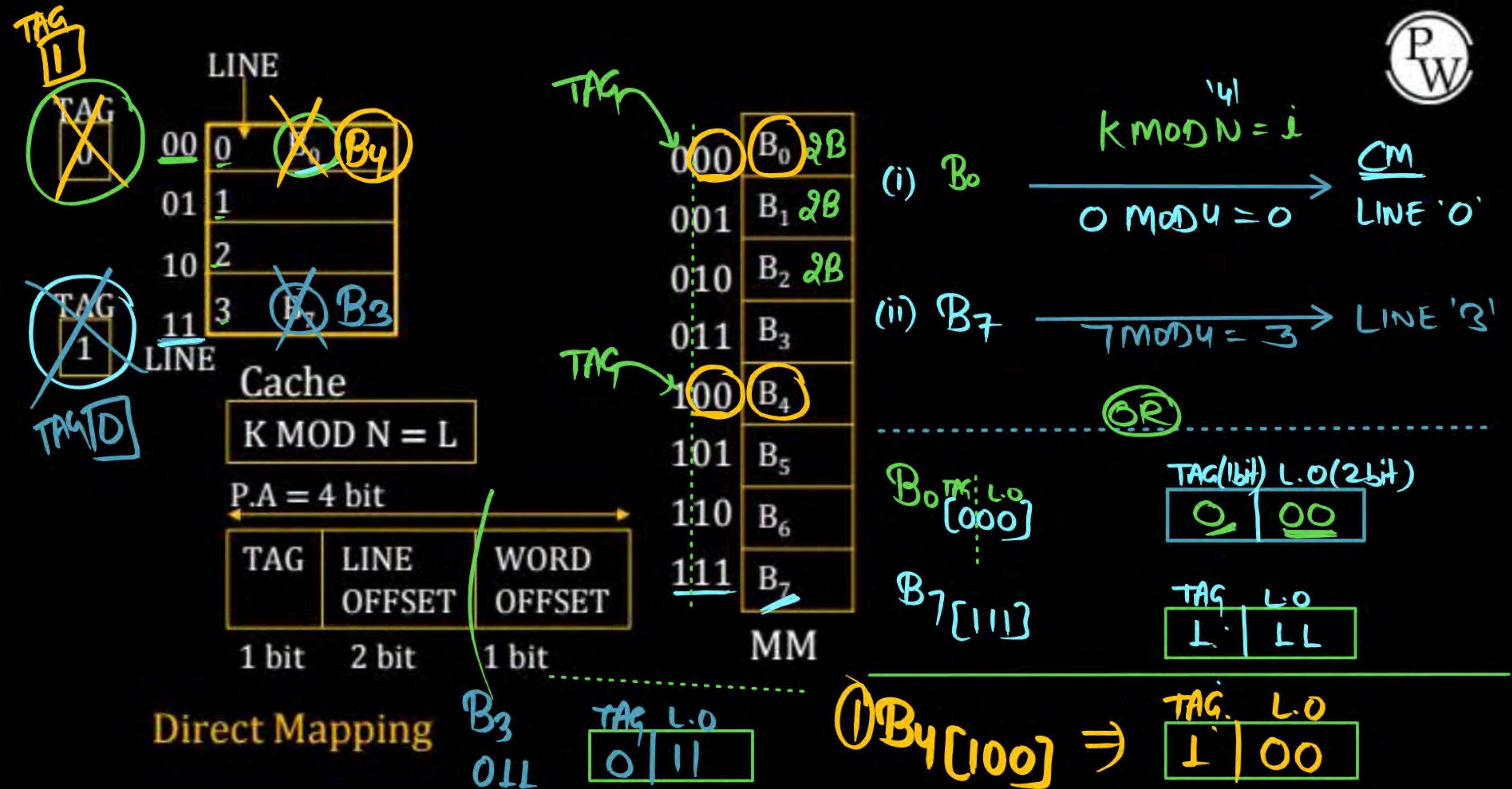
P
W

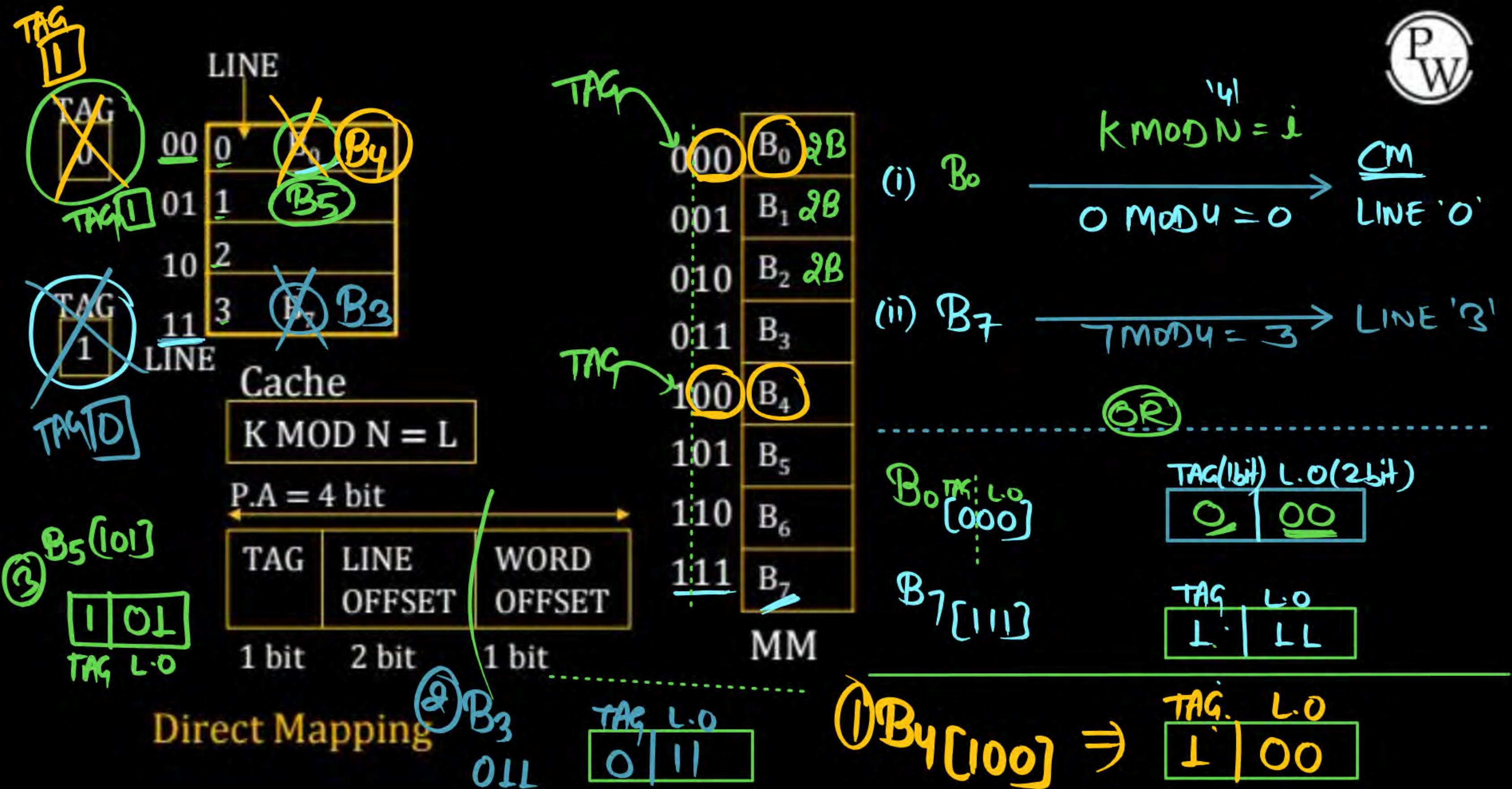


000	B ₀	2B
001	B ₁	2B
010	B ₂	2B
011	B ₃	
100	B ₄	
101	B ₅	
110	B ₆	
111	B ₇	

MM







MM Block

TAG LINE

0	00
---	----

 $B_0[000]$ Direct Mapping

$$K \bmod 4 = i$$

$$\underline{K \bmod N = i} \rightarrow$$

$$0 \bmod 4 = '0'$$

CM LINE

LINE '0'

TAG LINE

1	11
---	----

 $B_7[111]$

$$K \bmod N = i \rightarrow$$

$$7 \bmod 4 = '3'$$

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)

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

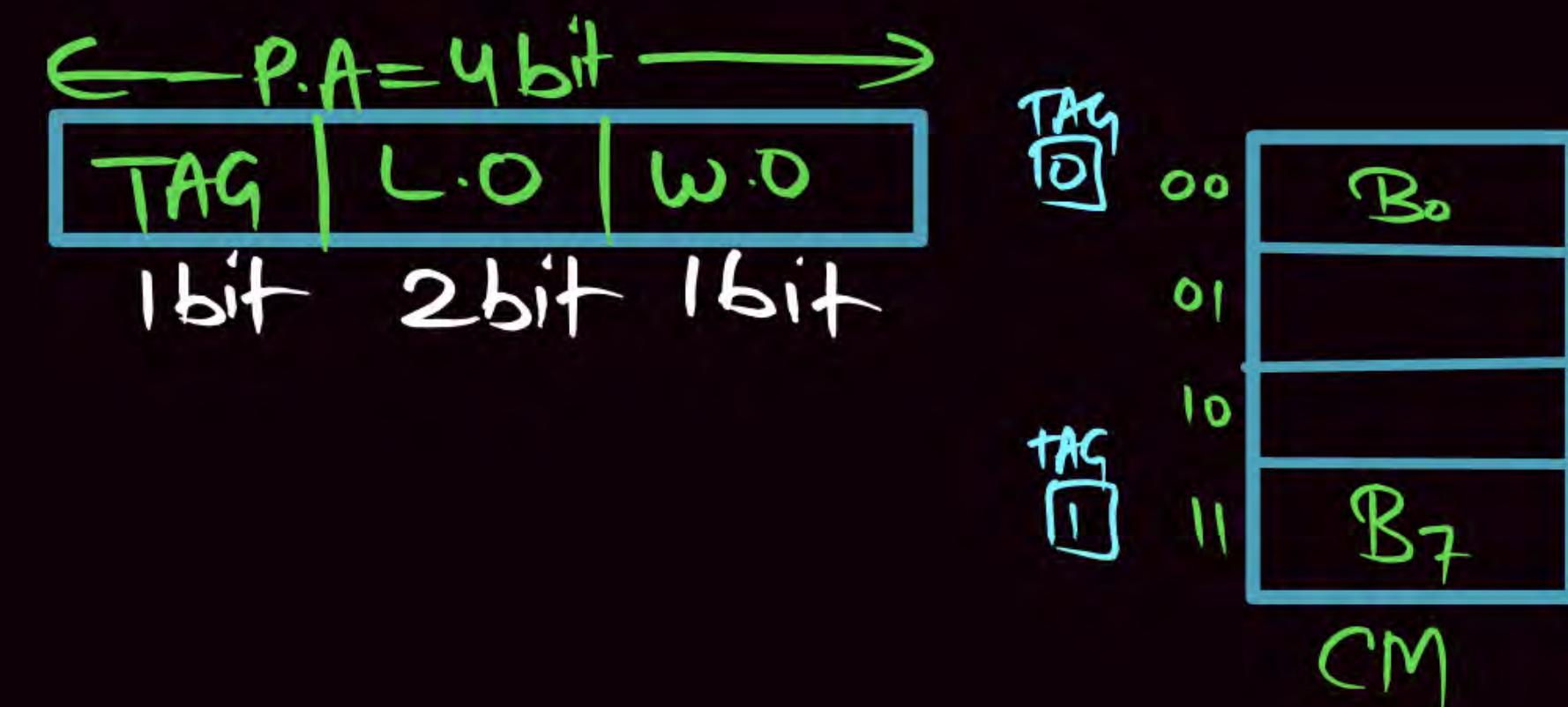
Q How to check Cache Hit @ Miss Practically ?

P.A = 4 bit

MM = 16B

Cache = 8B

Block Size = 2B



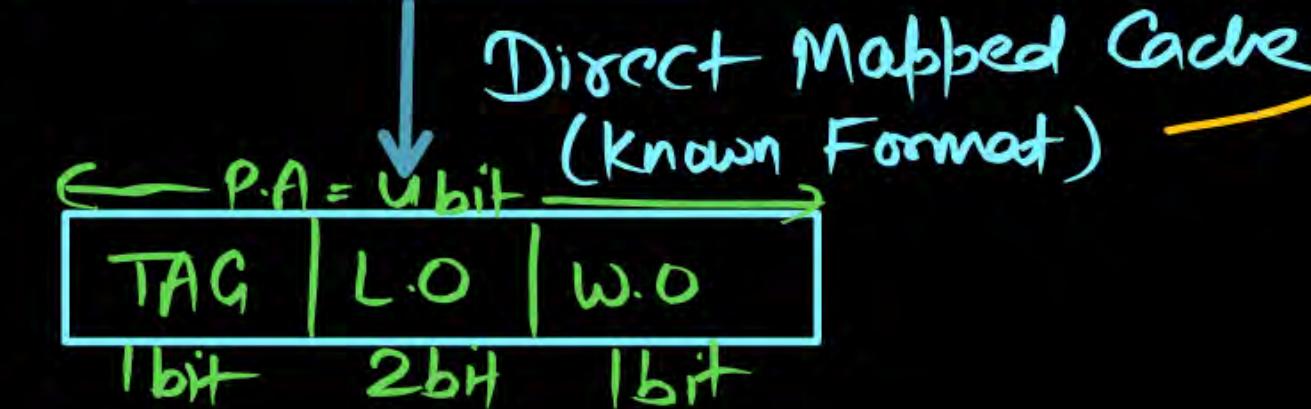
Consider the following program CPU generated Request Refer to Cache.

I₁: MOV r₀ [0000]
^{B₀}

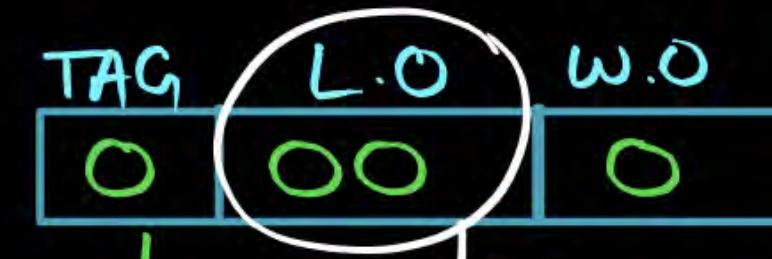
I₂: MOV r₁ [1000]
^{B₄}

I₃: ADD r₀r₁

T₁: 0000



format
According to
Previous example



Consider the following program

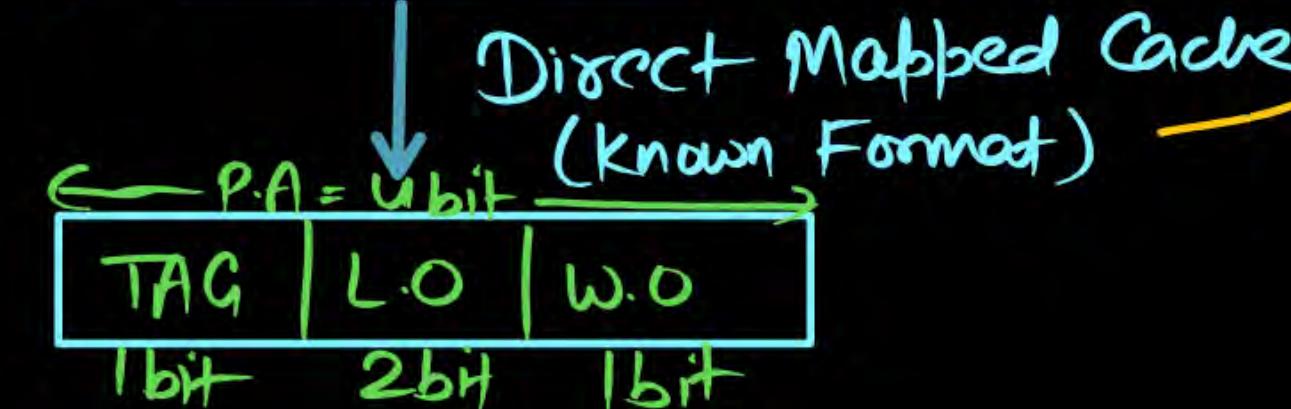
I₁: MOV r₀ [0000]^{B₀}

I₂: MOV r₁ [1000]^{B₄}

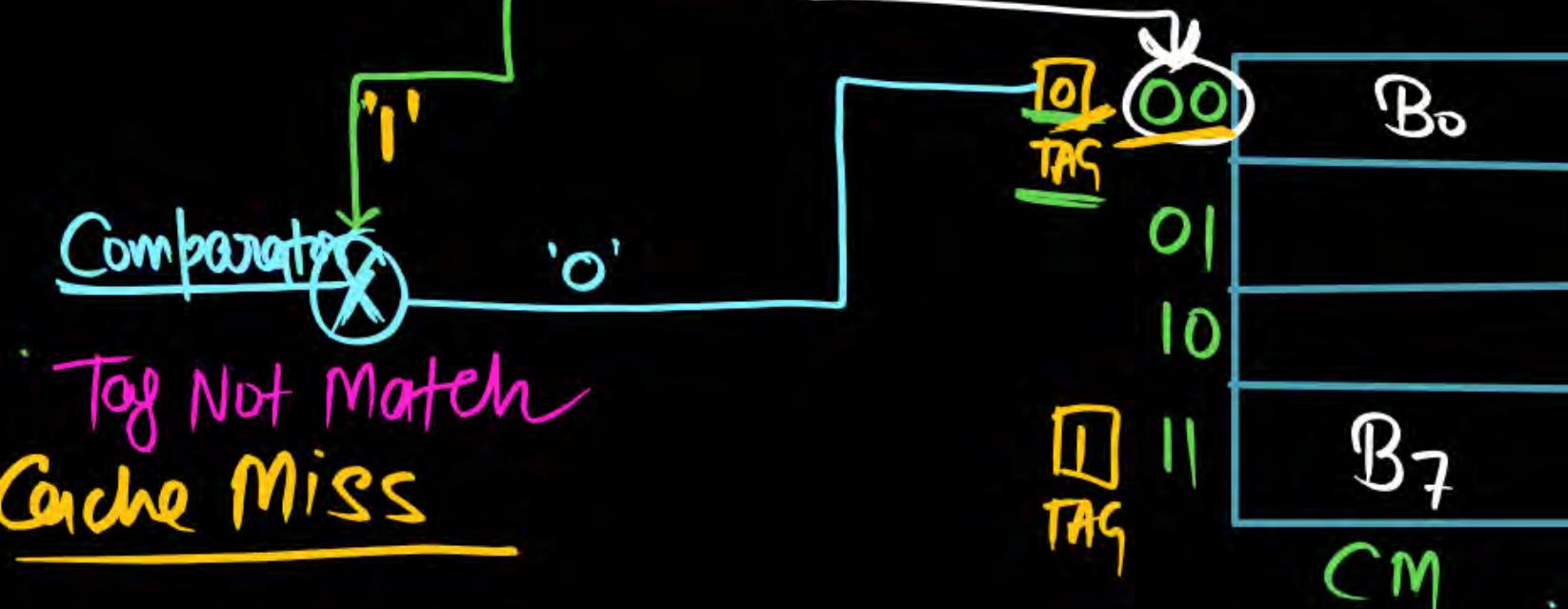
I₃: ADD r₀r₁

CPU generated Request Refer to Cache.

I₂: 1 000

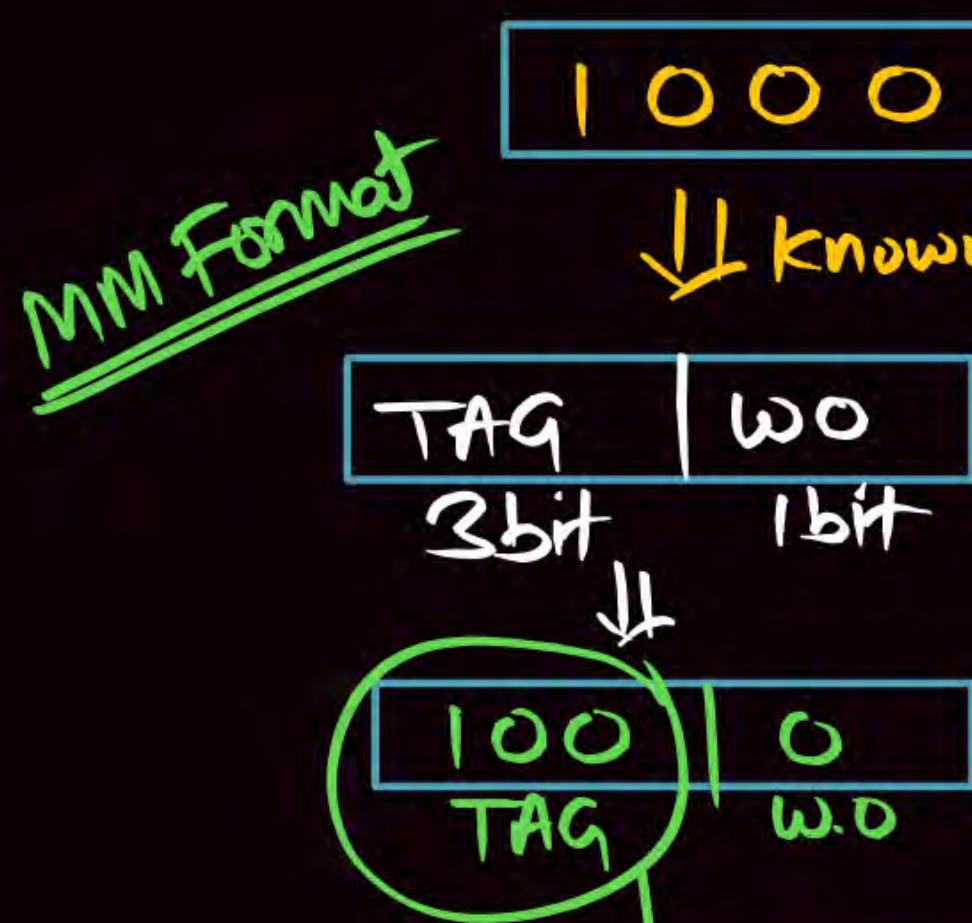


format
According to
Previous example



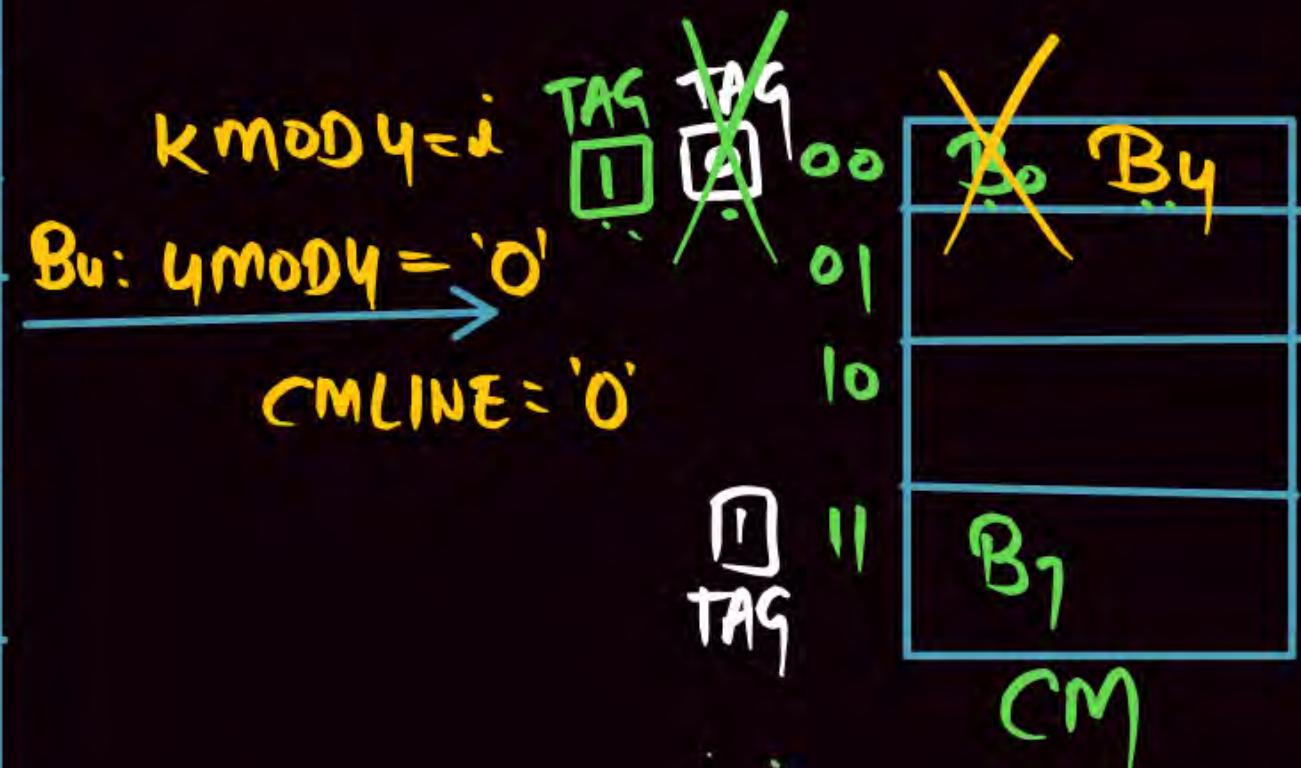
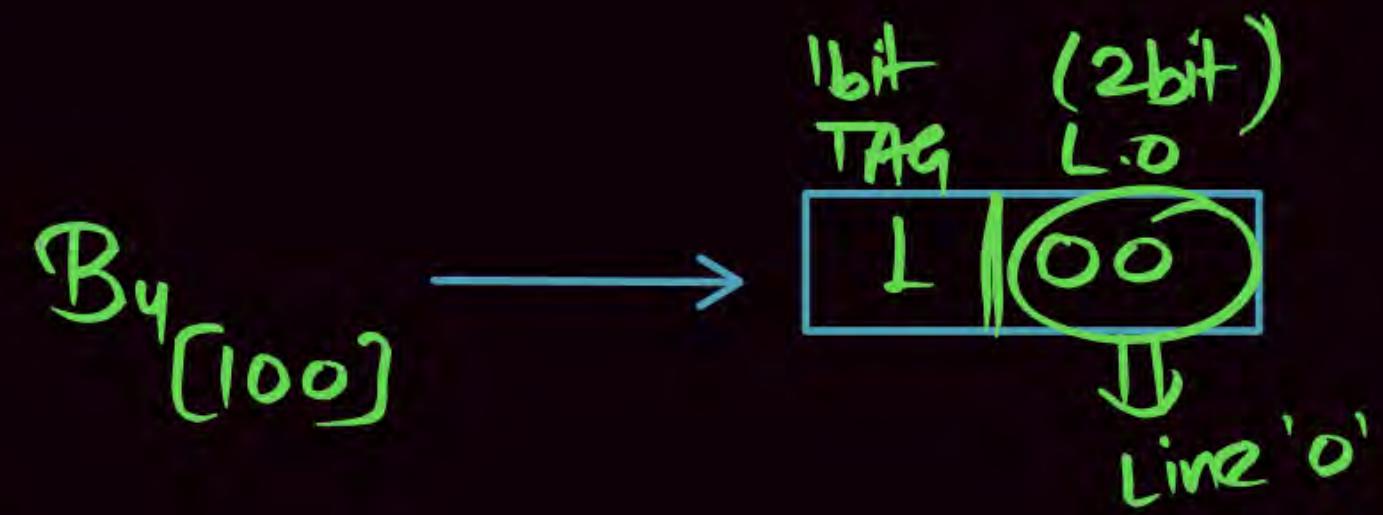
here Cache Miss then
Reference forward
to Main Memory

Reference Forward to Main Memory:

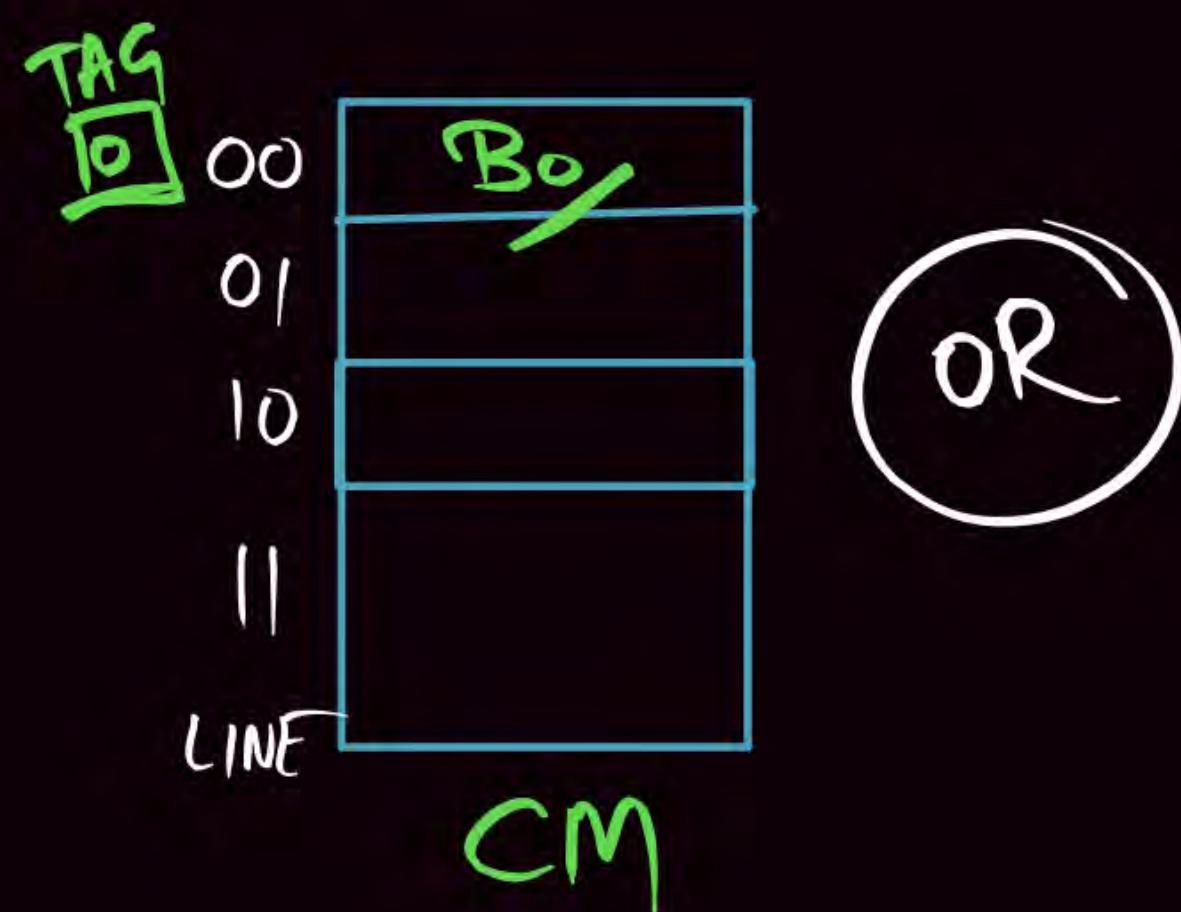
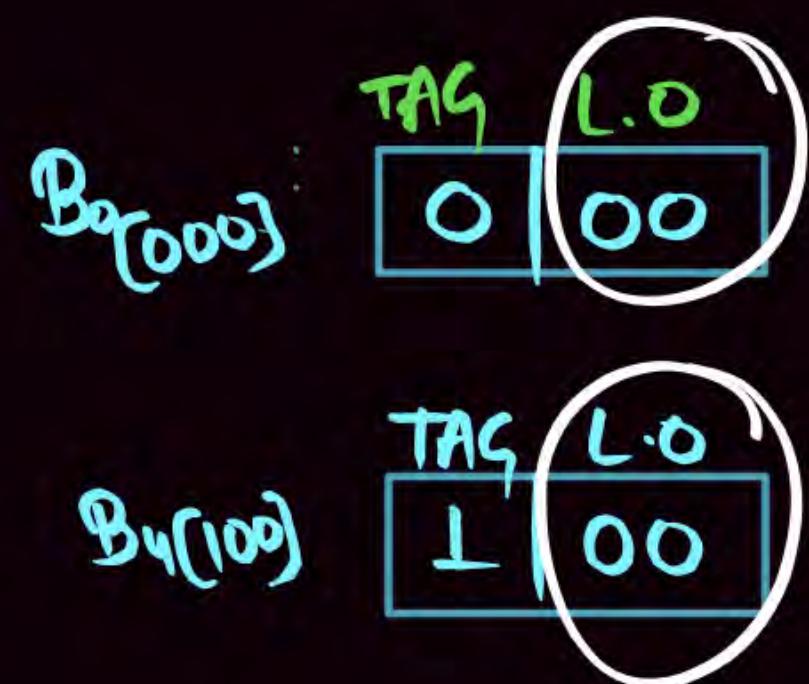


Main Memory

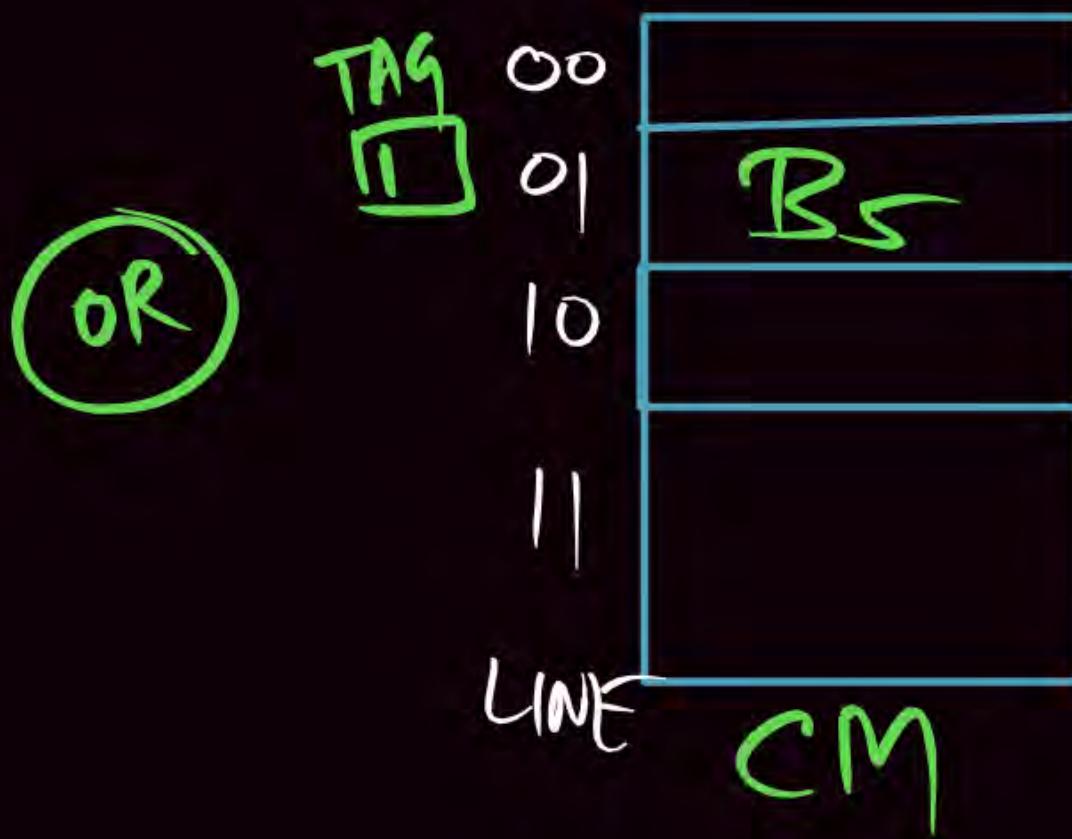
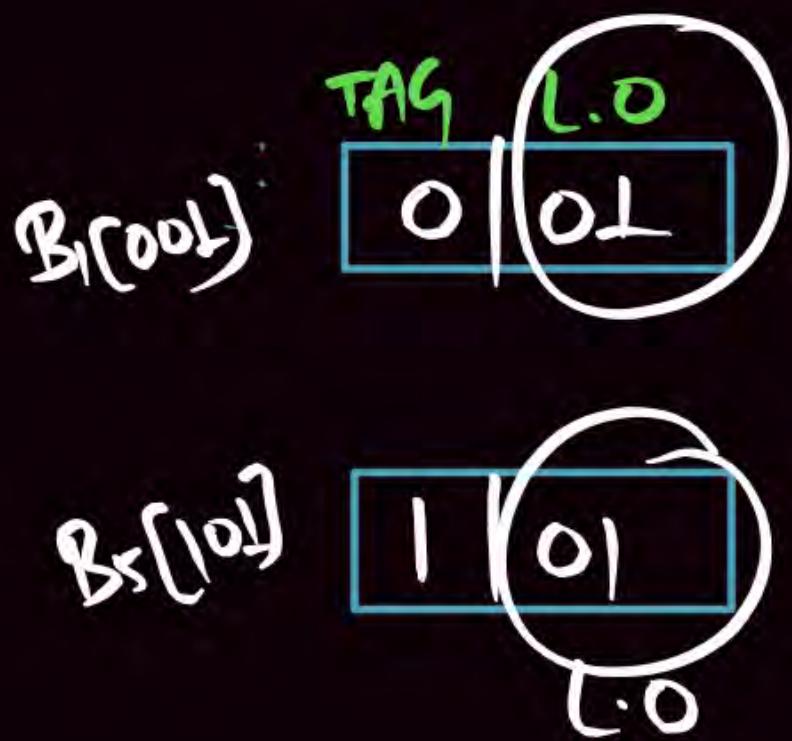
000	B ₀
001	B ₁
010	B ₂
011	B ₃
100	B ₄
101	B ₅
110	B ₆
111	B ₇



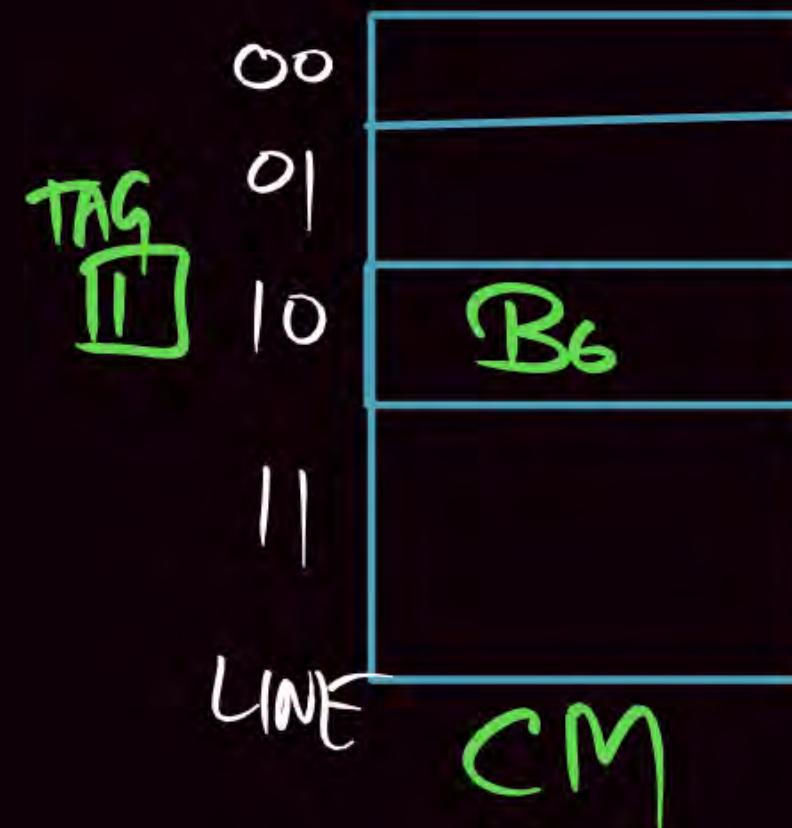
(i) B_0 & B_4 mapped to CM Line '0'



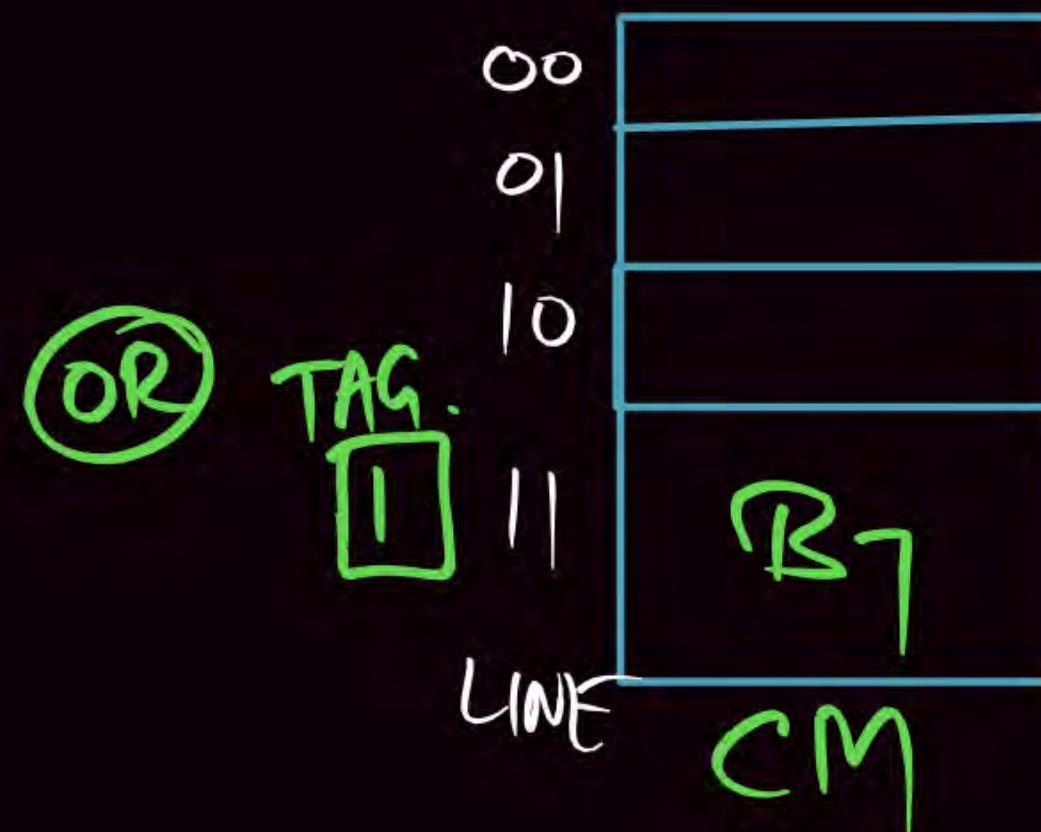
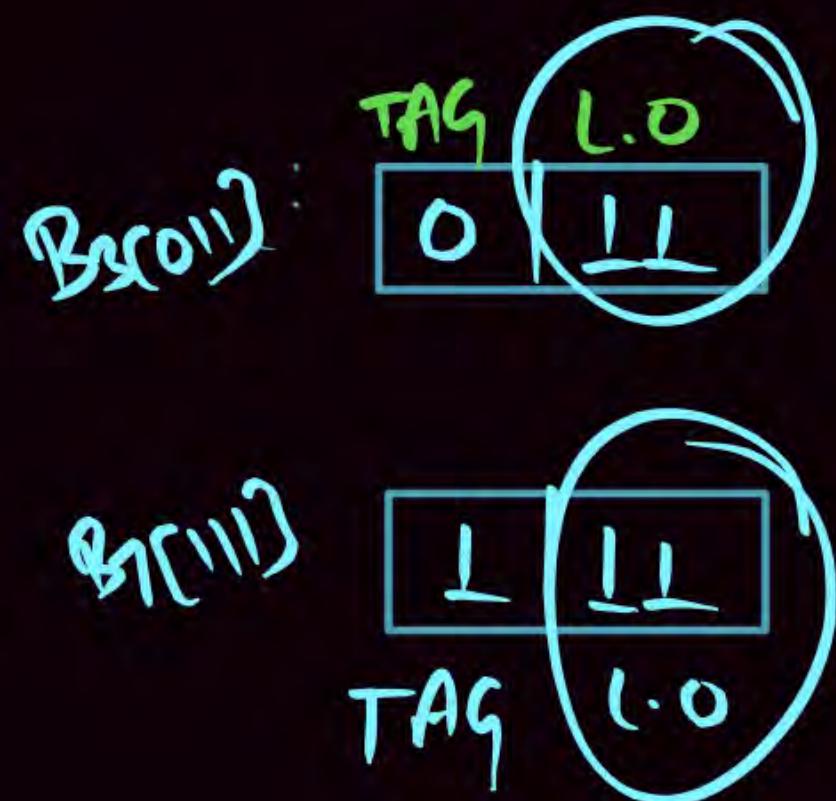
(ii) B_L & B_S mapped to CM Line '1'



(iii) B_2 & B_6 mapped to CM Line '2'.



(iv) B_3 & B_7 mapped to CM Line '3'



**THANK
YOU!**

