

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

01 Cache Memory

02 Mapping Techniques

✓ Memory Hierarchy.

✓ Cache Memory

↳ Simultaneous Access

↳ Hierarchical Access

✓ Locality of Reference [LOR]

① Temporal LOR

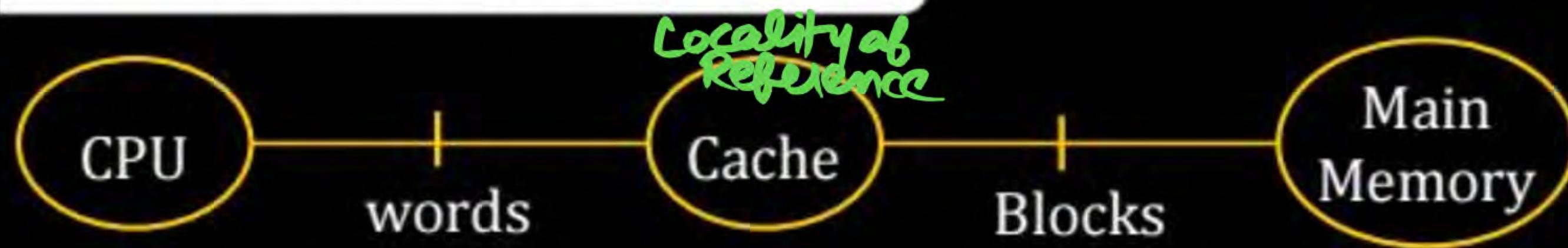
② Spatial LOR

& P4Q & Question.

Cache Memory

- ① Cache org.
- ② Mapping Technique
- ③ Replacement Algo
- ④ Cache Updation Technique
- ⑤ Multi level Cache.

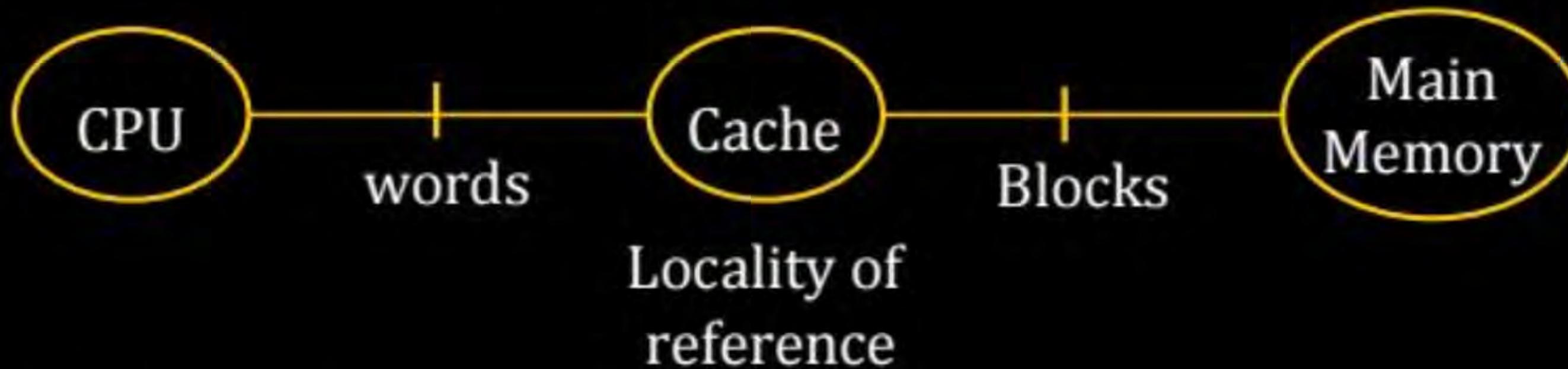
Memory Organization



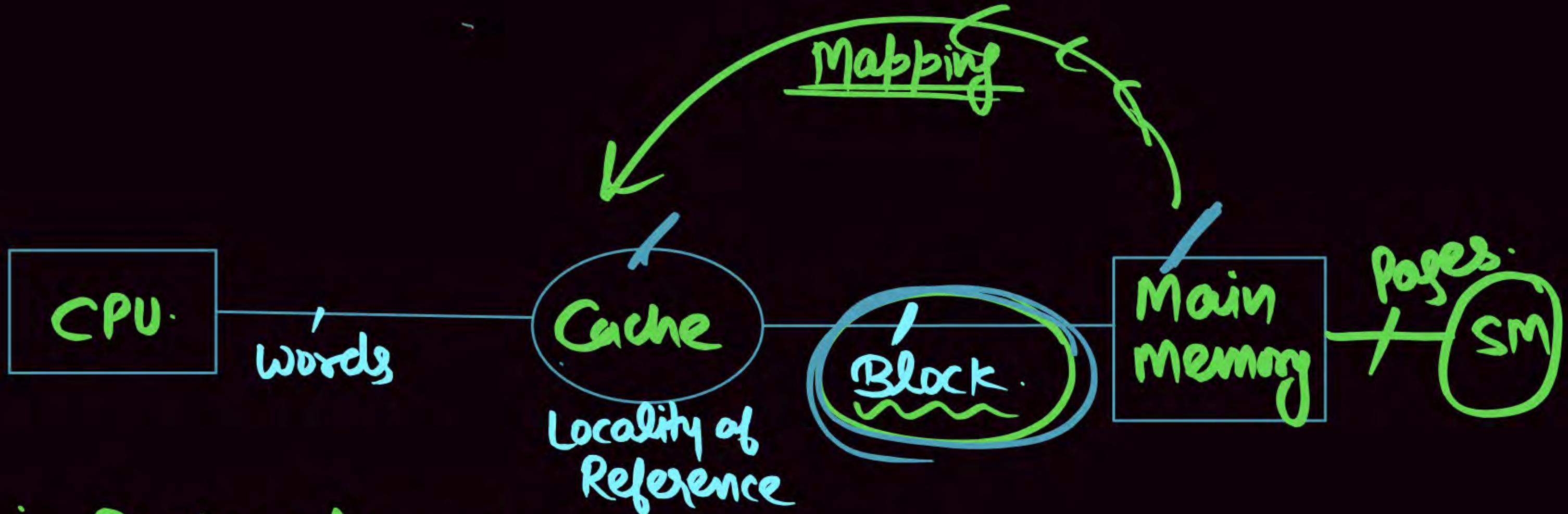
- Cache Memory is the fastest Memory in the Memory Hierarchy
- Cache Memory is used as intermediate Memory between Main Memory & CPU.
- Cache Memory is used to Fill the Gap between fastest Processor & slowest memory

- Cache Use Locality of Reference to Reduce the Average Access time.
- Cache Work on Locality of Reference .
- So CPU Perform Read|write operation on Cache Memory.

Cache Memory



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



Basic Design element of Cache memory

- ① Memory Org.
- ② Mapping Technique
- ③ Replacement Algo
- ④ Updating Technique
- ⑤ Multi level Cache

MM Size > CM Size

① Memory Organization :

In the Cache Design, Data is transferred from Main Memory to Cache Memory in the form of Block.

So Both the Memory Must be organized Based on Block Size.

Memory Organization :

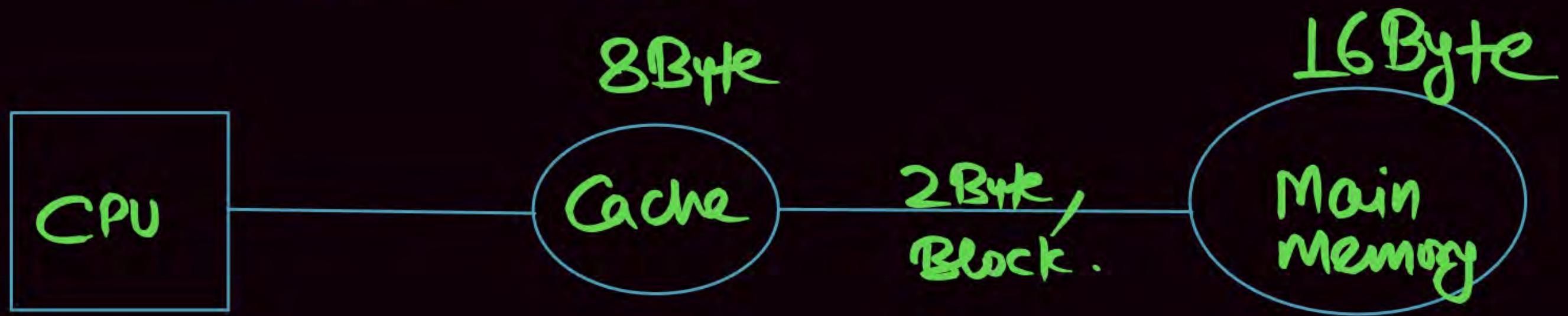
Main Memory & Cache Memory are divided into
Parts [equal size] Based on Block Size, called
Main Memory Block & Cache Memory Block Respectively

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

(#LINES)

$$\#MM\text{ Block} = \frac{MM\text{ Size}}{\text{Block Size.}}$$

Before Org.



MM size > CM size .

(a) Cache Memory

③ Consider a 8 Byte Cache with 2 Byte Block Size :

Before org.

~~CM Size = 8 Byte~~

000	1B
001	1B
010	1B
011	1B
100	1B
101	1B
110	1B
111	1B

Cache Index = 3 bit

AFTER ORG.

$$\#CM\ Block (\#LINES) = \frac{CM\ Size}{Block\ Size} = \frac{8B}{2B} = 4$$

0	2B
1	2B
2	2B
3	2B

Line offset 2bit

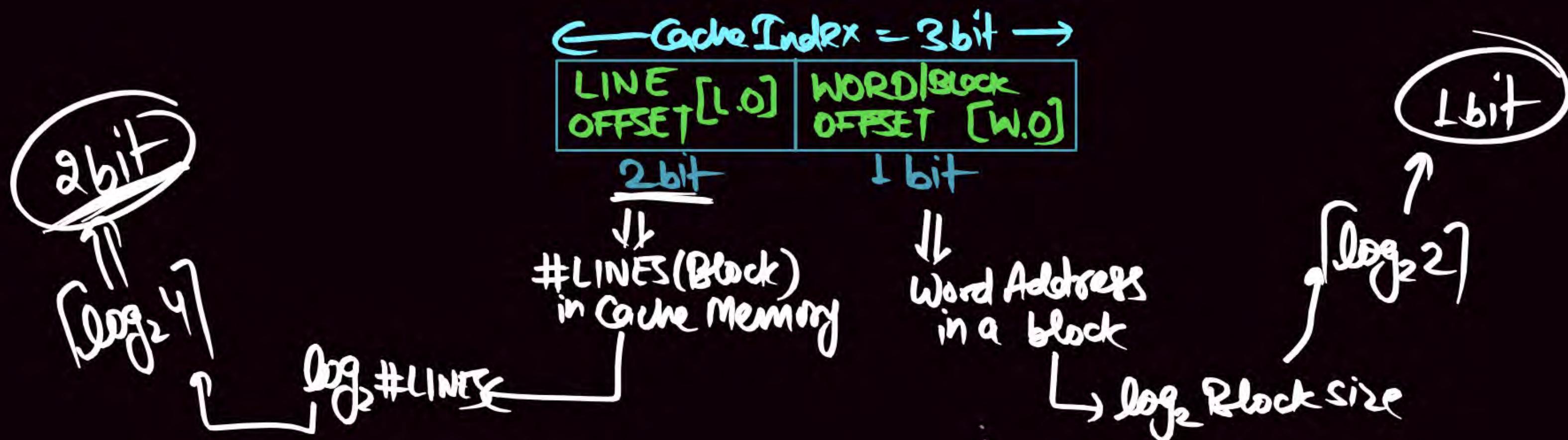
word offset 1bit

$$\begin{aligned} CMSize &= \#LINE \times \text{Block size} \\ &= 4 \times 2 \text{Byte} \end{aligned}$$

~~CMSize = 8 Byte~~

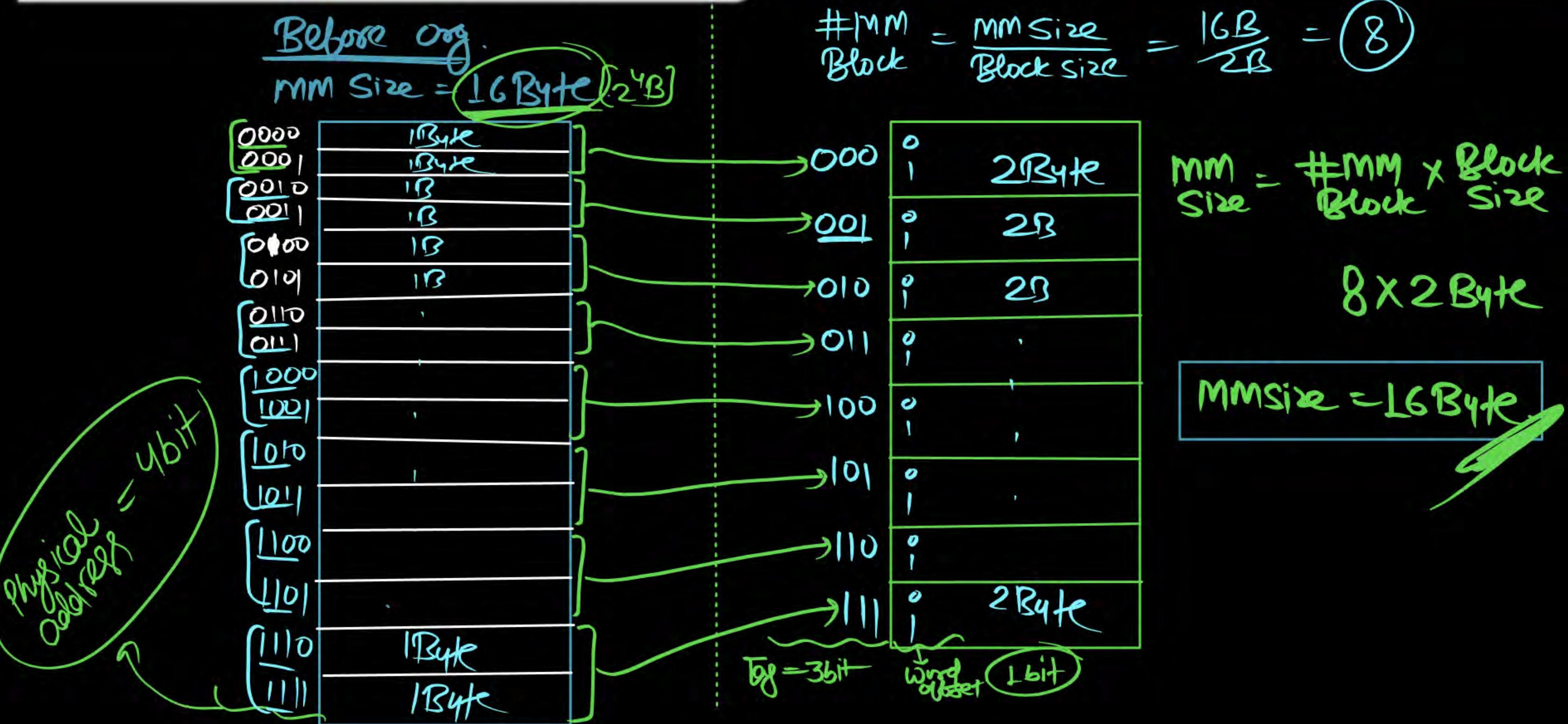
Before & After the organization Cache Memory Capacity is Same but internal Structure is Different.

Cache memory address interpreted as



Main Memory

Consider a MM Size of 16 Byte with
Block Size = 2Byte the MM org as:



Main Memory

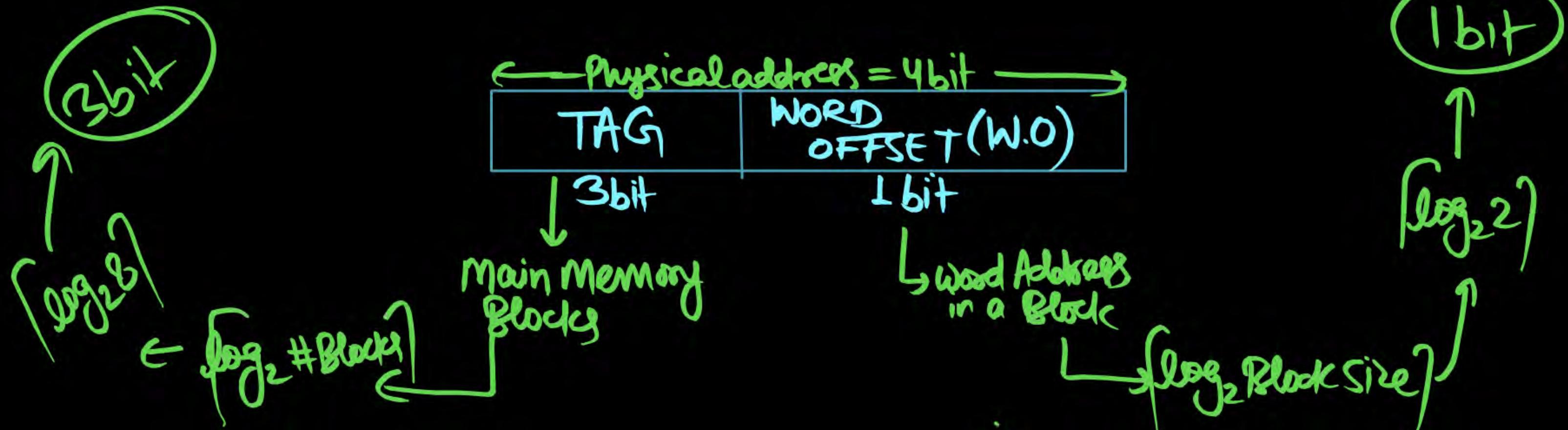
Here TAG → Indicate

MM Block Number

Number of bit to represent MM Blocks

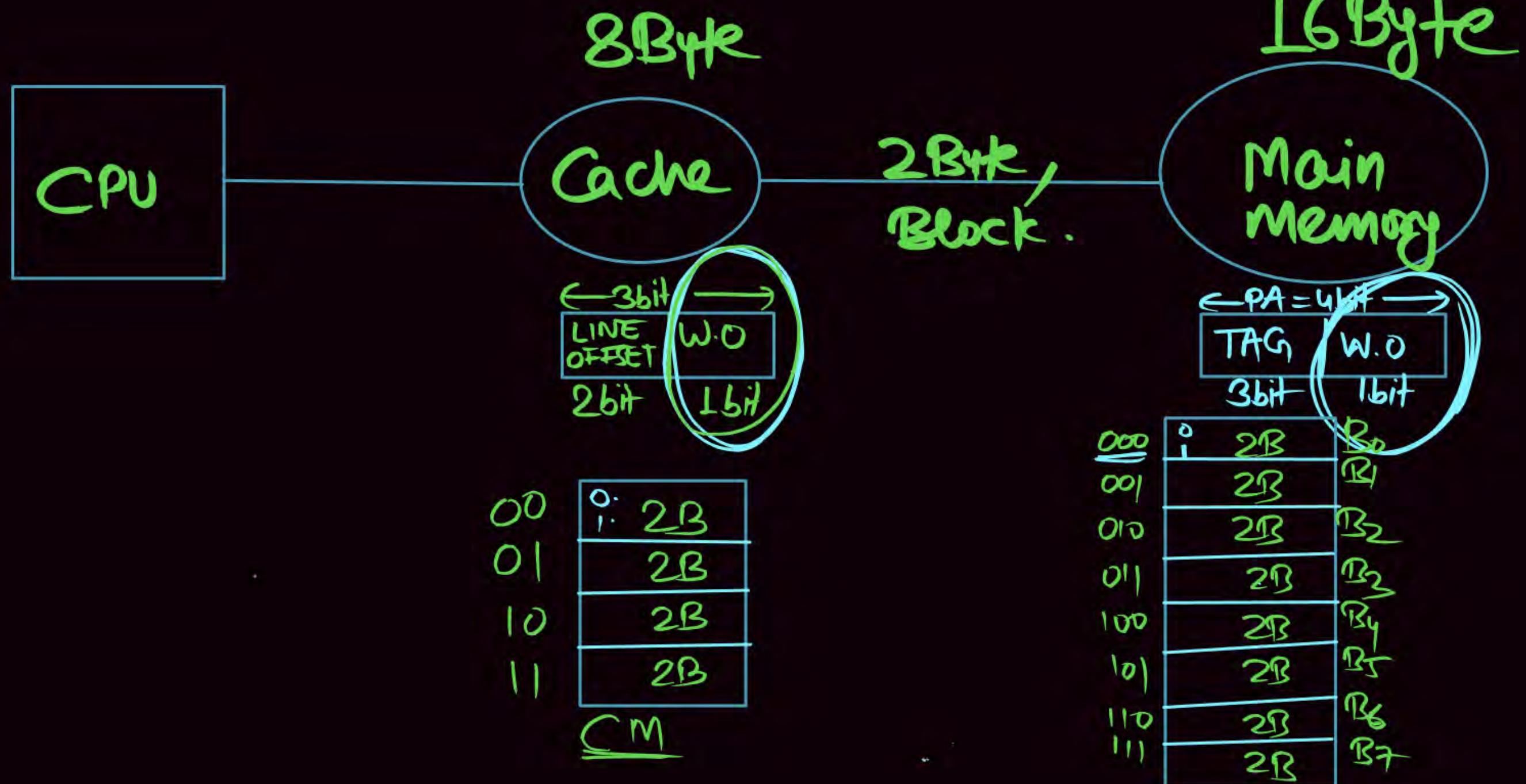
Before & After the organization Main Memory Capacity is same But internal structure is different.

Now Physical address is interpreted as:



AFTER ORG.

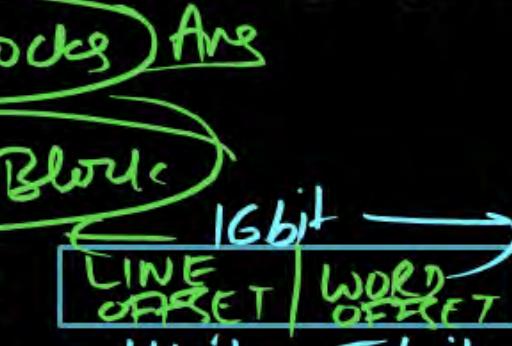
V.V. VDref

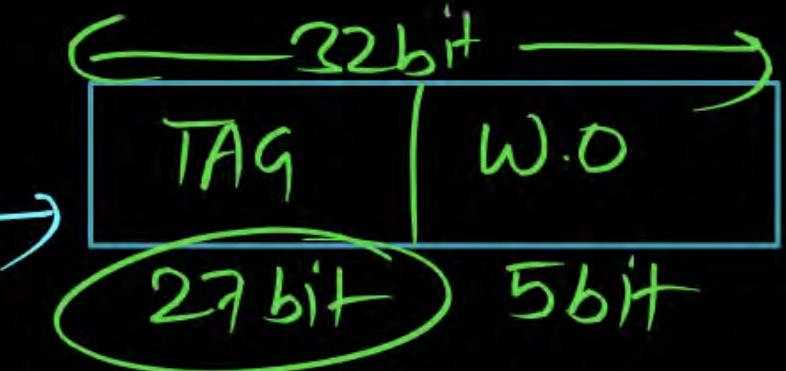


Q.

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

P
W

- Block size = 32B*
- (i) # CM Block \rightarrow 2K Blocks Ans
 - (ii) # MM Blocks \rightarrow 128M Blocks
 - (iii) CM Address format 
 - (iv) MM Address format



Solⁿ

$$\text{Cache Index} = \underline{16 \text{ bit}} \Rightarrow \text{CM Size} = 2^{16} \text{ Byte} = 64 \text{ KB}$$

$$\text{Block Size} = \underline{32 \text{ Byte}}$$

$$\text{Physical address} = 32 \text{ bit} \Rightarrow \text{main Memory Size} = 2^{32} \text{ Byte} = 4 \text{ G Byte}$$

$$(i) \# \text{CM Block} = \frac{\text{CM Size}}{\text{Block Size}} \\ (\# \text{LINES})$$

$$= \frac{64 \text{ KB} [2^16]}{32 \text{ Byte} [2^5]} = \left(2^{11} \text{ Blocks}\right) \text{ Ans}$$

(2K Block)

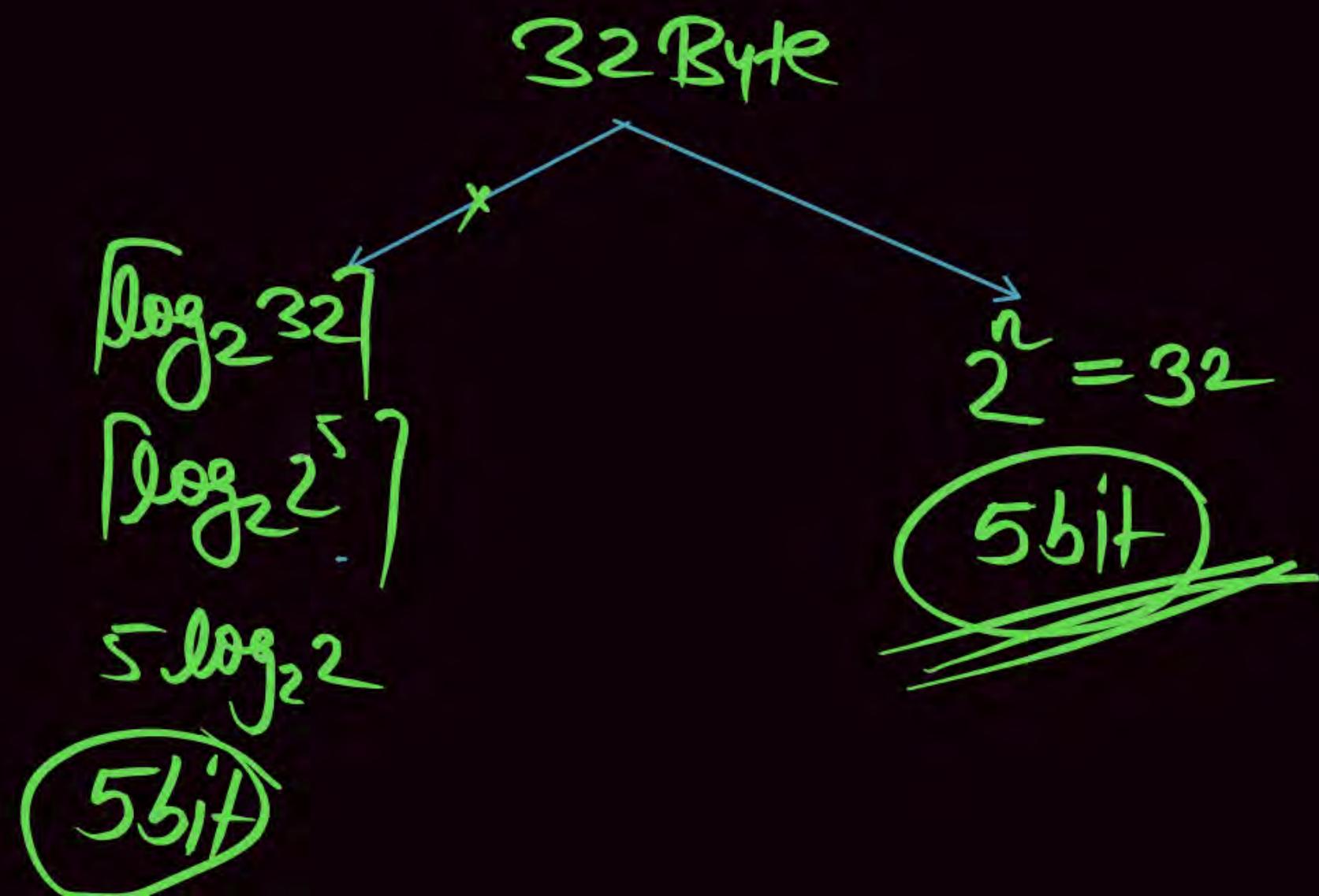
$$(ii) \text{MM Blocks} = \frac{\text{MM Size}}{\text{Block Size}} = \frac{4 \text{ GB} [2^{32}]}{32 \text{ Byte} [2^5]}$$

$$\text{MM Block} = 2^{27} \\ = \left(128 \text{ M Block}\right) \text{ Ans}$$

Q

32 Byte & asking How Many No. of bits Required.

Soln



$$2^1 = 2$$

$$2^2 = 4$$

$$2^3 = 8$$

$$2^4 = 16$$

$$2^5 = 32$$

$$2^6 = 64$$

$$2^7 = 128$$

$$2^8 = 256$$

$$2^9 = 512$$

$$2^{10} = 1024 [1 \text{ kilo}]$$

$$1 \text{ K(kilo)} = 2^{10}$$

$$1 \text{ M(mega)} = 2^{20}$$

$$1 \text{ G(Giga)} = 2^{30}$$

$$1 \text{ T(Tera)} = 2^{40}$$

$$1 \text{ Byte} = 8 \text{ bit}$$

$$1 \text{ Nibble} = 4 \text{ bit}$$

1b number

$$1 - 2 = 1 \text{ bit}$$

$$0 \text{ to } 4 \quad 4 = 2 \text{ bit}$$

$$0 \text{ to } 8 \quad 8 \Rightarrow 3 \text{ bit}$$

$$(9 \text{ to } 16) \quad 16 = 4 \text{ bit}$$

$$17 \text{ to } 32 \quad 32 = 5 \text{ bit}$$

$$33 \text{ to } 64 \quad 64 = 6 \text{ bit}$$

$$(65 \text{ to } 128) \quad 128 = 7 \text{ bit}$$

$$(129 \text{ to } 256) \quad 256 = 8 \text{ bit}$$

$$(257 \text{ to } 512) \quad 512 = 9 \text{ bit}$$

$$(513 \text{ to } 1024) \quad 1024 [1 \text{ Kilo}] = 10 \text{ bit}$$

$$1 \text{ K(Kilo)} = 2^{10}$$

$$1 \text{ M(Mega)} = 2^{20}$$

$$1 \text{ G(Giga)} = 2^{30}$$

$$1 \text{ T(Tera)} = 2^{40}$$

$$1 \text{ Byte} = 8 \text{ bit}$$

$$1 \text{ Nibble} = 4 \text{ bit}$$

- Diagram illustrating memory hierarchy and bit depths:
- (1) 50 → 6bit Ans
 - (2) 30 → 5bit Ans
 - (5) 256 Byte → 8bit Ans
 - (3) 45 → 6bit Ans
 - (4) 100 → 7bit Ans
 - (6) 500 Byte → 9bit Ans
 - (7) 150 Byte → 8bit Ans
 - (8) 8KB → $2^3 \cdot 2^{10}$ → 13bit Ans
 - (9) 4GB → 32bit Ans
 - (10) 16Byte → 4bit Ans
 - (11) 128MBYTE → 27bit Ans
 - (12) 256 KB → 18bit Ans
 - (13) 16GB → 34bits Ans

n bit addresses can represent 2^n word (if word addressable)
 2^n byte (if byte addressable memory)

16

(i) 16bit $\longleftrightarrow 2^{16}\text{B} = 64\text{KB}$

(ii) 22bit $\longleftrightarrow 2^{22}\text{B} = 4\text{MB}$

(iii) 29bit $\longleftrightarrow 2^{29}\text{B} = 512\text{MB}$

(iv) 18bit $\longleftrightarrow 2^{18}\text{B} = 256\text{KB}$

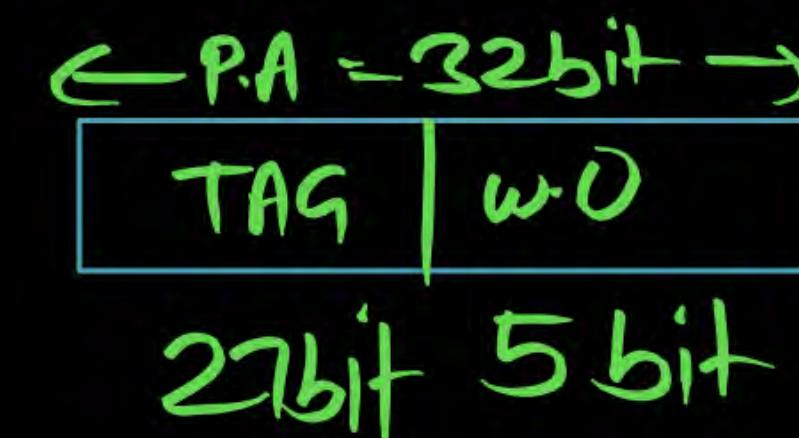
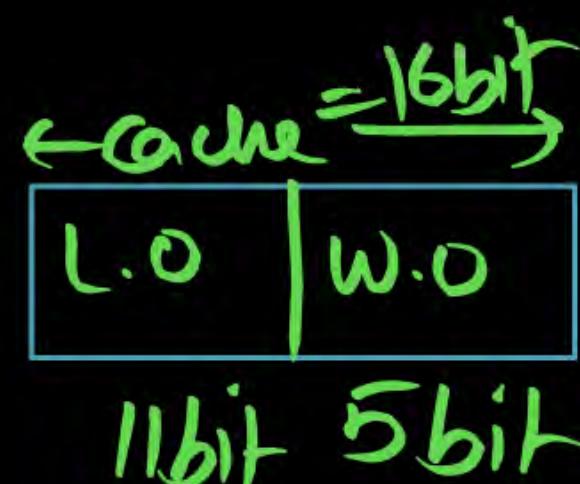
(v) 32bit $\longleftrightarrow 2^{32}\text{B} = 4\text{GB}$

(vi) 34bit $\longleftrightarrow 2^{34}\text{B} = 16\text{GB}$

Q.

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

- (i) # CM Block $\rightarrow w.o = 5 \text{ bit}$
- (ii) # MM Blocks
- (iii) CM Address format
- (iv) MM Address format



$$(i) = 2^{11} = 2K \text{ Block}$$

$$(ii) 2^{27} = 128M \text{ Block}$$

Q.

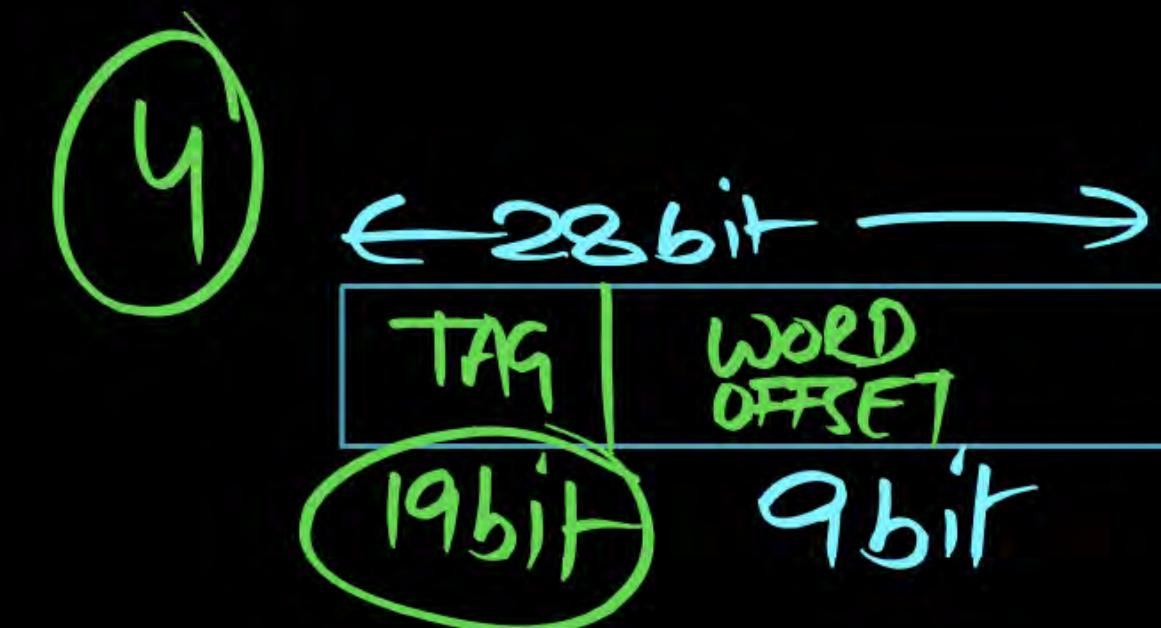
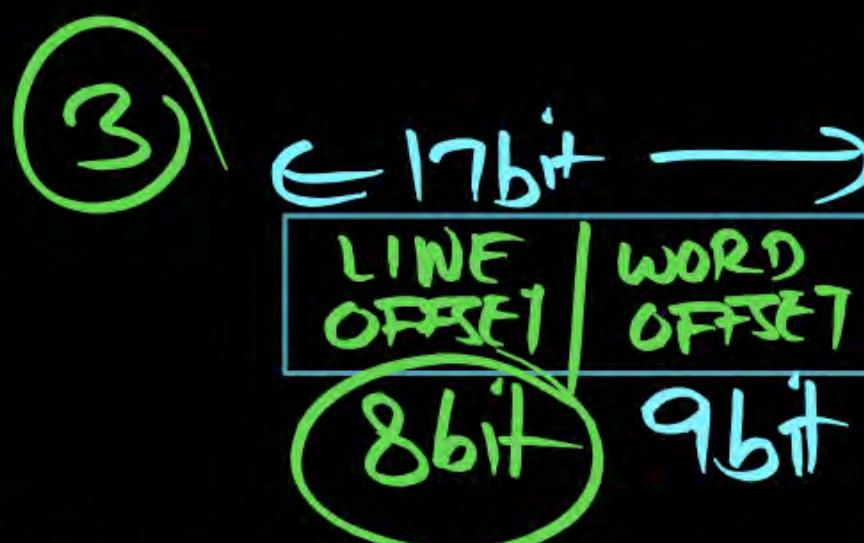
Consider a system MM = 256MB, Cache = 128KB & Block size = 512 Byte then

- ✓
a bit
1) # LINES?
3) L.O & W.O Format

28bit

17bit

- 2) #MM Blocks
4) TAG & W.O Format



① #LINES = $2^8 = 256$ Ang

② #MM Blocks = $2^{19} = 512M$ Block

P
W

Mapping

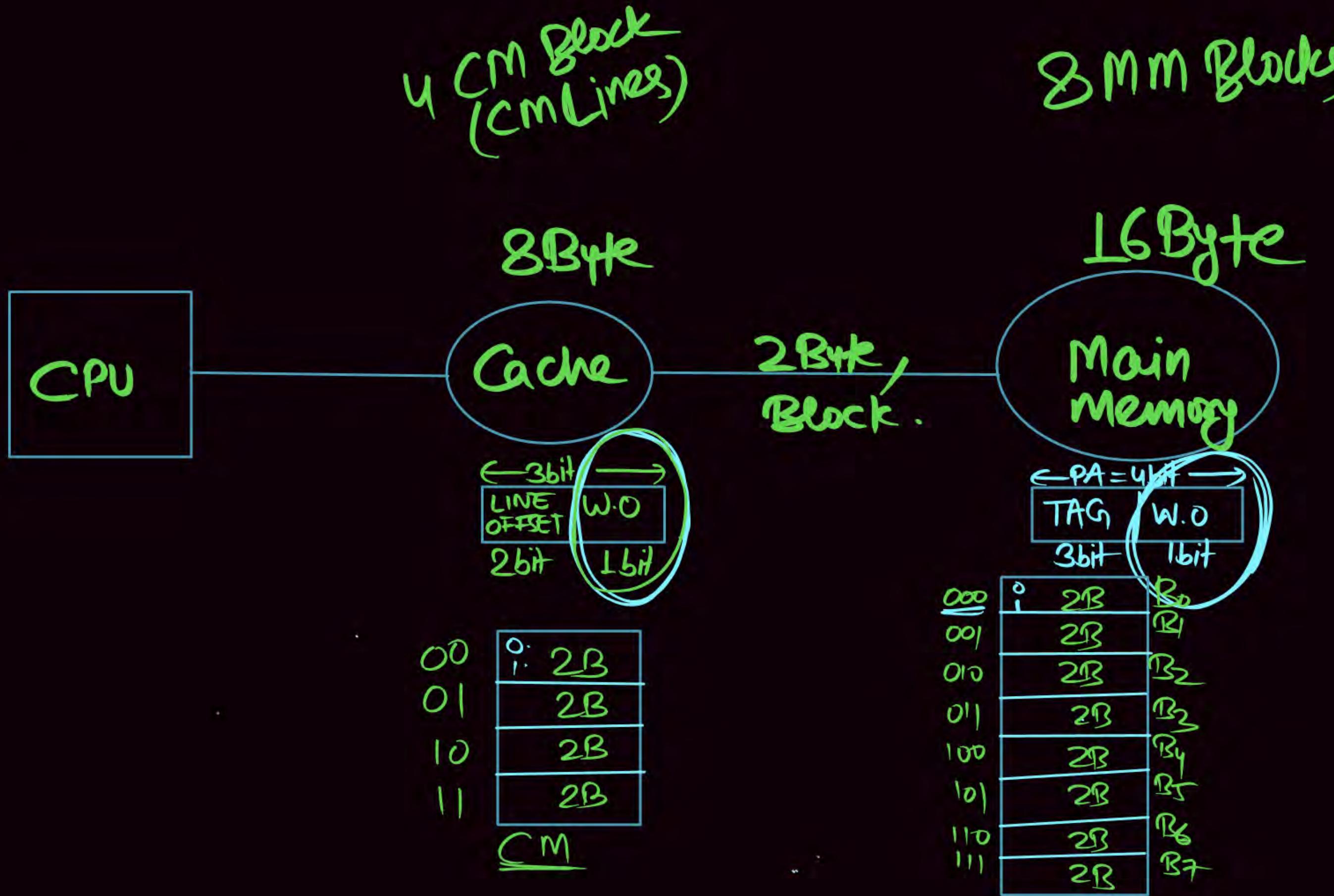
The process of transfer the Data from Main Memory to Cache

Memory is called mapping. There are 3 Type of Mapping

Technique

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

U.V. V Dref



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

Cache Controller interpret the P.A as

This TAG is
different
from MM tag.



$$\# \text{LINES} = \frac{\text{CM SIZE}}{\text{Block Size}}$$

$$\text{Line offset} = \log_2 \# \text{LINE} = 2^n$$

$$\text{TAG} = \text{P.A} - [\text{L.O} + \text{W.O}]$$

Direct Mapping

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



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

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

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

Q.

Consider a Direct Mapping if the size of Cache memory is 512KB

P
W

& Main Memory 512MB & Cache line size (Block) is 64KB the

2^{19}

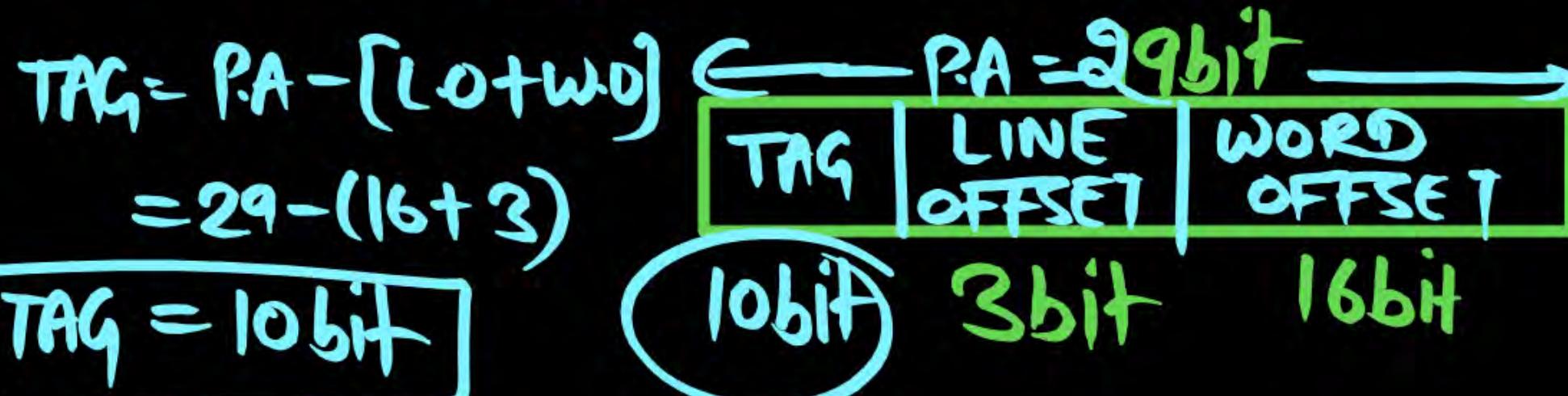
16bit

calculate the number of bit required for

(i) P.A (29bit) (ii) TAG (10bit) (iii) B.O (3bit) (iv) W.O (16bit)

(v) #LINES
(8) (vi) TAG Memory Size

$$\#LINE = \frac{CM\ Size}{Block\ Size}$$



$$\frac{2^{19}}{2^{16}} = 2^3$$

$$L.O = 3bit$$

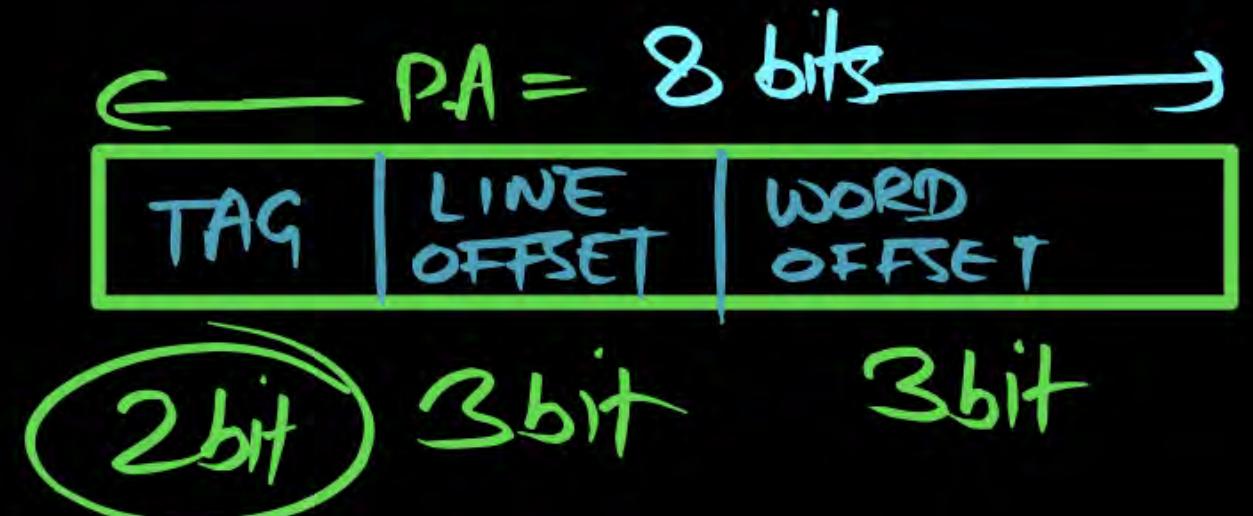
Q.

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

P
W

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

8 2 3 3



$$\# \text{LINES} = \frac{\text{CM Size}}{\text{Block size (LINE)}}$$

$$= \frac{64B}{8B} = \frac{2^6}{2^3}$$

$$= 2^3$$

$$L.O = 3bit$$

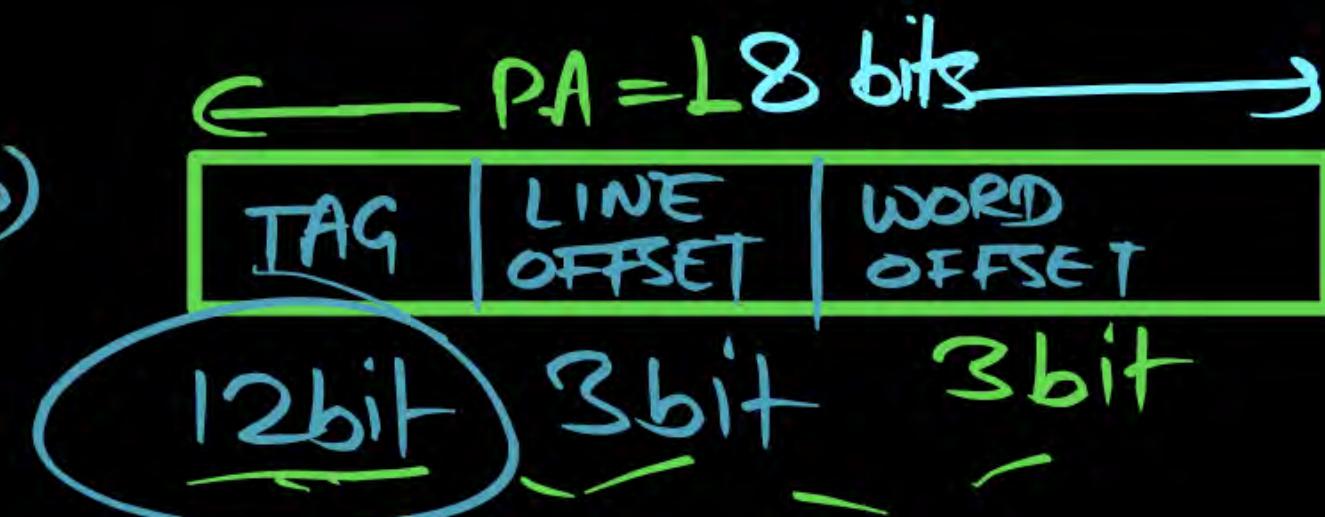
Q.

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

P
W

Byte. MM = 256 KByte then #bits for P.A, TAG, L.O, W.O Tag
18bit 3bit
memory size?

$$\begin{aligned} \text{TAG} &= \text{P.A} - (\text{L.O} + \text{W.O}) \\ &= 18 - (3 + 3) \\ &= 18 - 6 \\ &= 12 \text{ bits } \underline{\text{Ans}} \end{aligned}$$



$$\begin{aligned} \# \text{LINES} &= \frac{\text{CM Size}}{\text{Block Size}} \\ &= \frac{64B}{8B} = 8 \end{aligned}$$

$$L.O = 3 \text{ bit}$$

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.

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]

P
W

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

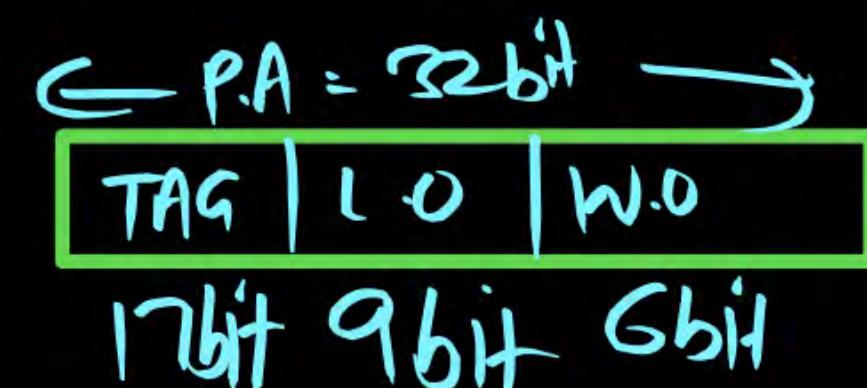
NAT

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 17 bits. Are

J
G

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



$$\frac{2^5}{2^6} = 2^9$$

1) Direct Mapping

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

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

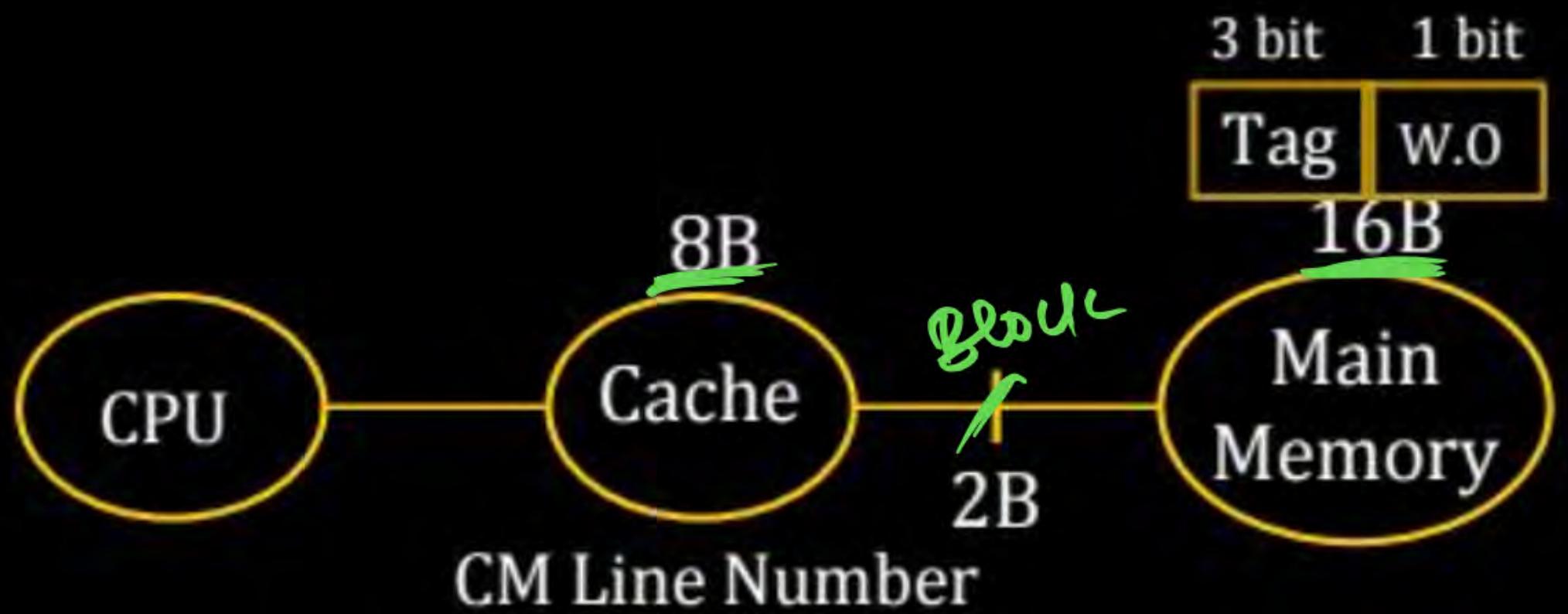
(Or)

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

K: MM Block No.

N: # of Cache Line

i: CM Line Number



CM Line Number

$$0 \bmod 4 = 0$$

$$1 \bmod 4 = 1$$

$$2 \bmod 4 = 2$$

$$3 \bmod 4 = 3$$

$$4 \bmod 4 = 0$$

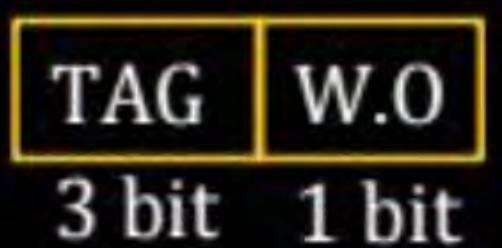
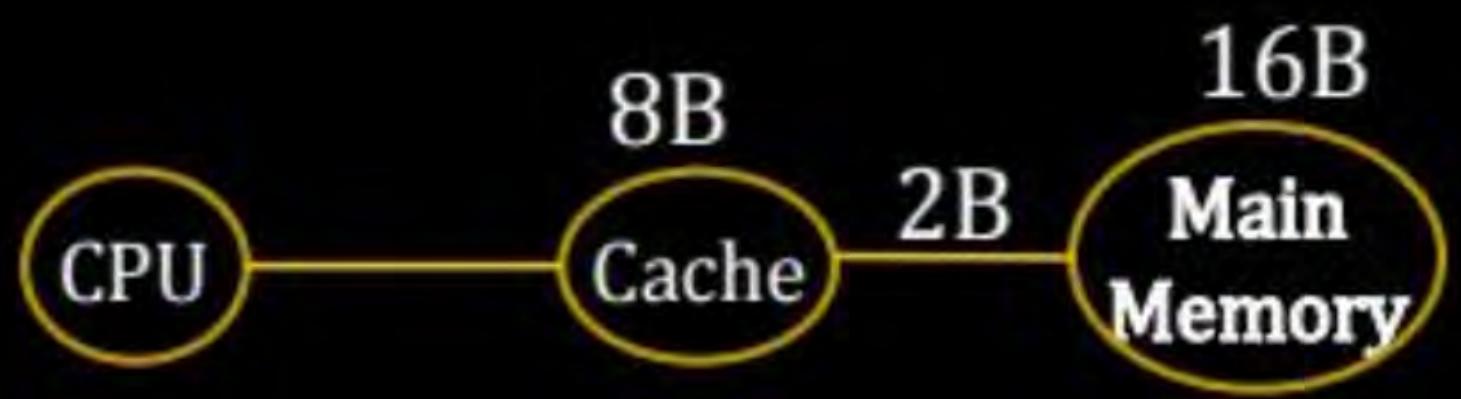
$$5 \bmod 4 = 1$$

$$6 \bmod 4 = 2$$

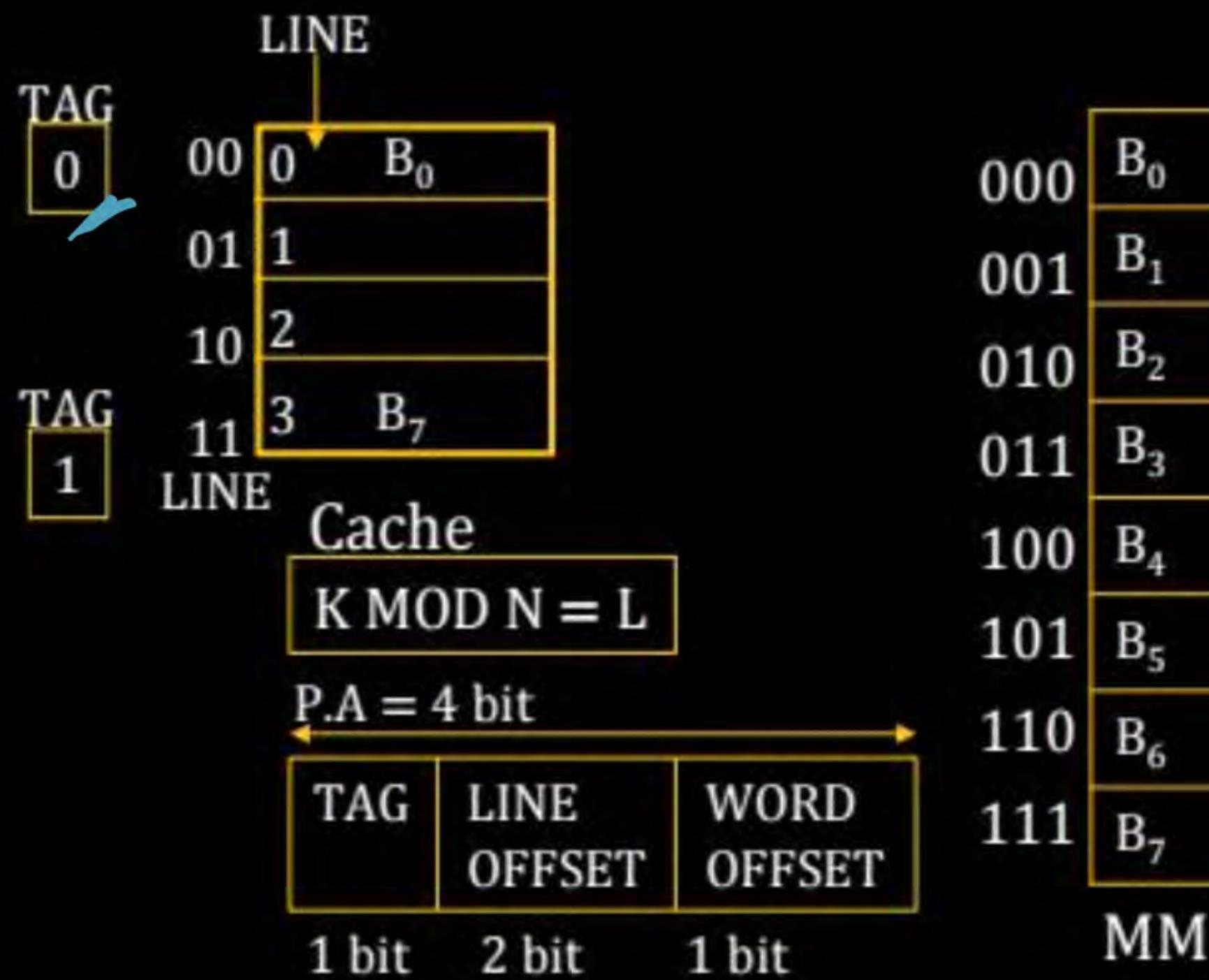
$$7 \bmod 4 = 3$$

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



P
W



Direct Mapping

MM Block

TAG LINE

0	00
---	----

 $B_0_{[000]}$

Direct Mapping

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

CM LINE

LINE '0'

TAG LINE

1	11
---	----

 $B_7_{[111]}$

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

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)

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

Consider the following program

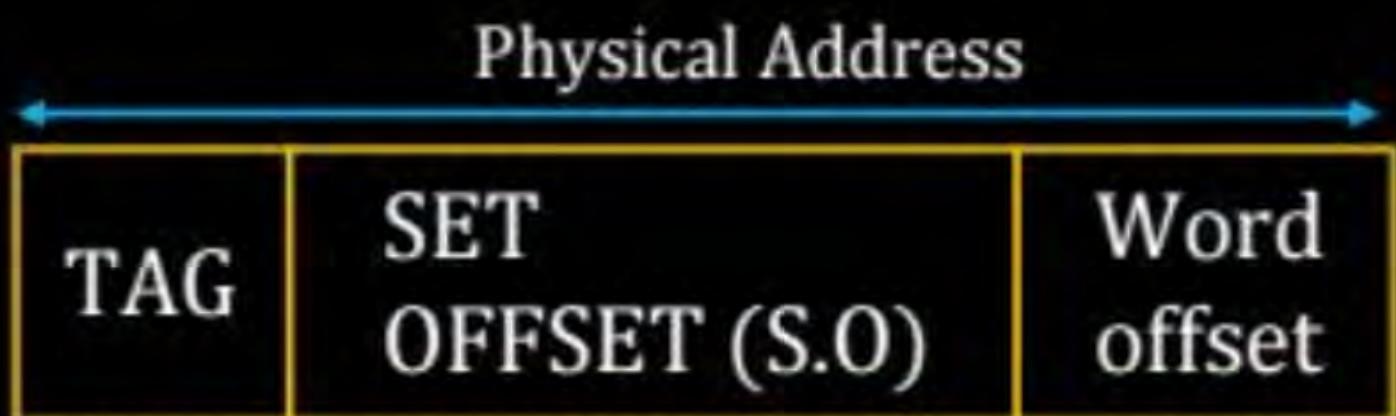
I₁: MOV r₀ [0000]

I₂: MOV r₁ [1000]

I₃: ADD r₀r₁

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.

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

P
W

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 $(ABCDE)_{16}$?

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.

- (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

[GATE - 1999]

P
W

Q.

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]

P
W

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]

**THANK
YOU!**

