COMPUTER SCIENCE

Computer Organization and Architecture

Cache Memory

Lecture_03

Vijay Agarwal sir

01 Memory Access

02 Cache Memory

Memory Hierarchy.

Type of Access.

Average Access time Calculation

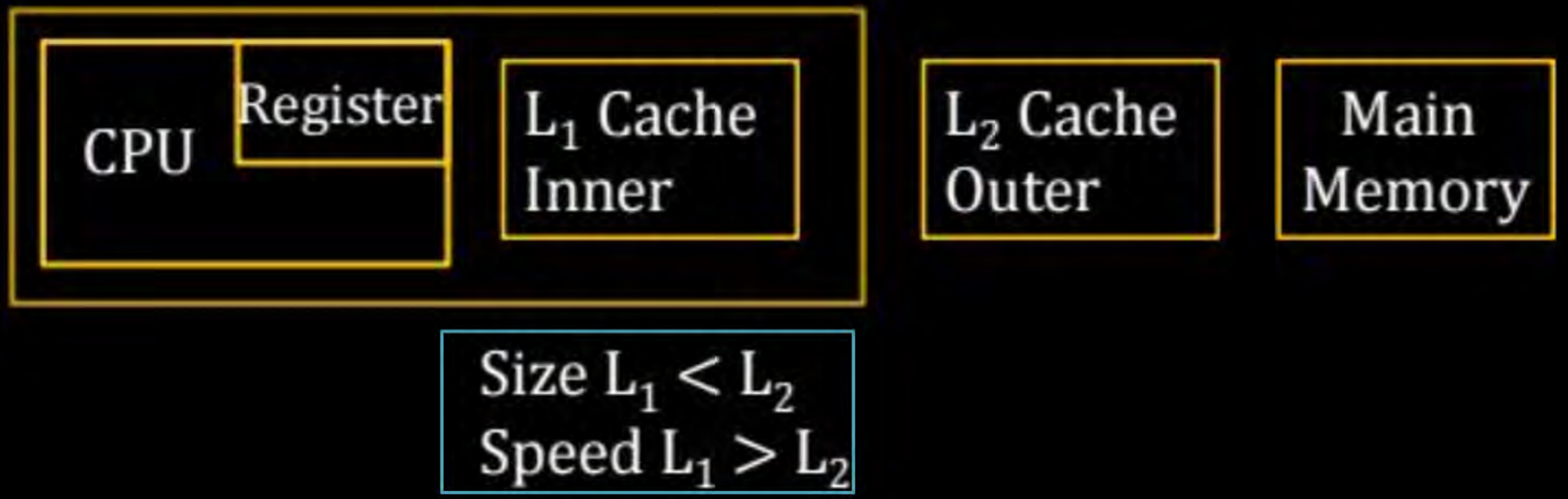
Locality of Reference (LOR).

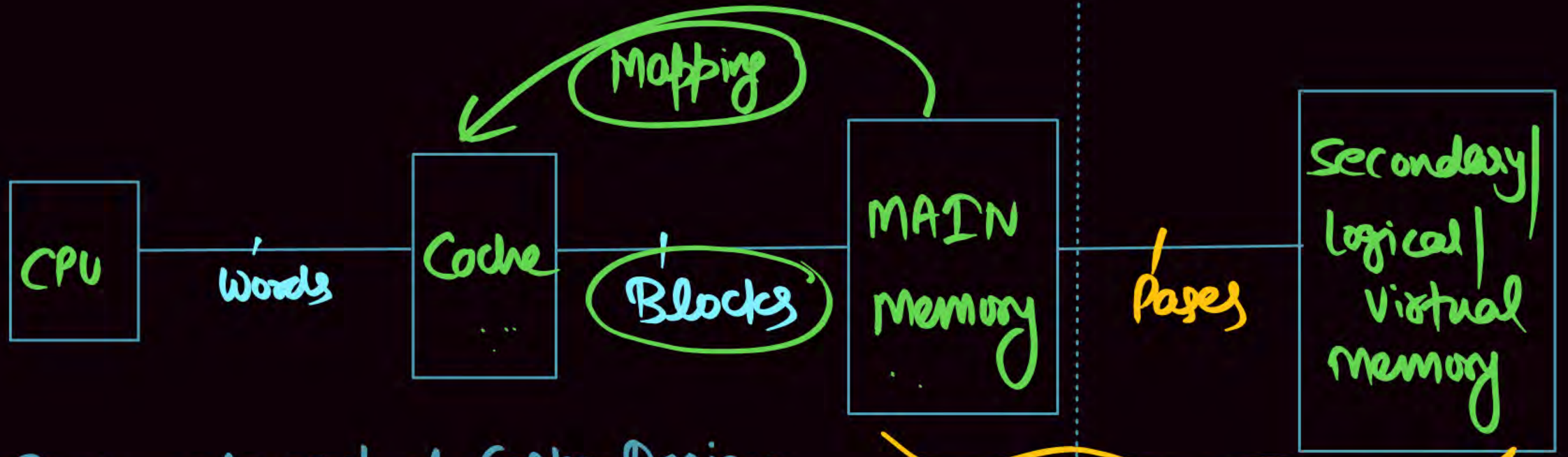
GATE - Question.

Type of Cache.

Types of Cache

1) Unified Cache: Instruction & Data both are placed in Same Cache.
2) Split Cache: This Cache logically Divide into two parts
   (i) Instruction Cache [I - cache]
   (ii) Data Cache [D- cache]
3) Multilevel Cache:





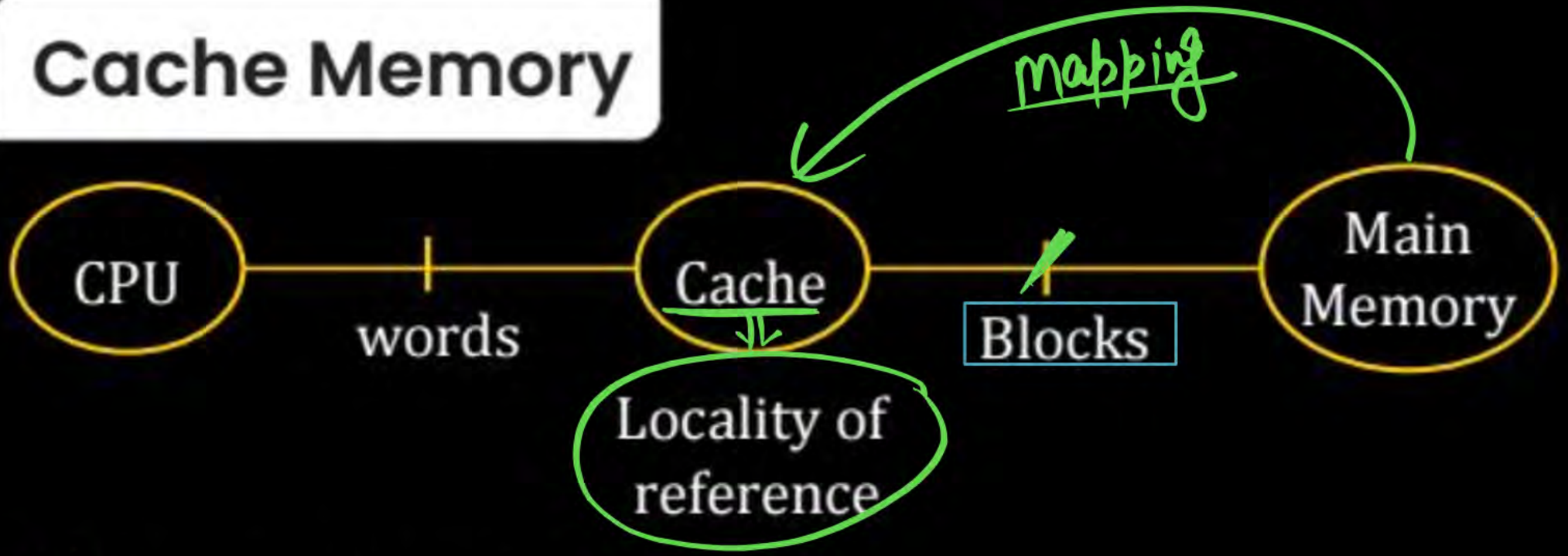
Basic element of Cache Design.

1. Memory Organization.
2. Mapping Technique → (i) Direct Mapping. (ii) Set Associative Mapping (iii) Fully Associative Mapping.
3. Replacement Algo.
4. Updating Technique
5. Multi Level Cache.

MMsize > CMsize.

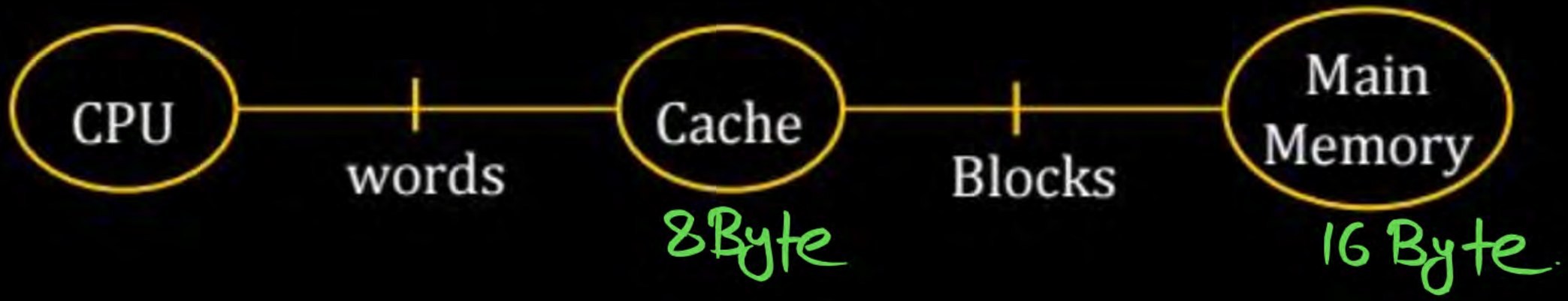
already Done in OS. (Paging)

Cache Memory



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

Memory Organization



Memory organization

In the Cache Design. Data is transfered from Main Memory to Cache Memory in the form of 'Blocks'.

- So Both Memory (Main Memory & Cache Memory) Must be organized Based on Block Size.

Memory organization

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

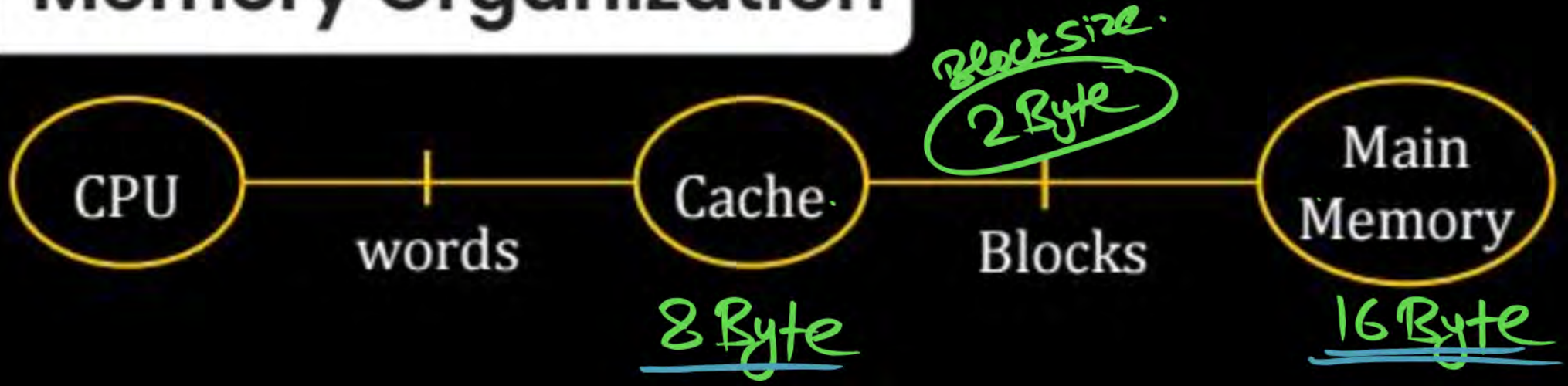
Called MM Block & CM Block (Number of Lines) Respectively.

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

$$\#\text{CM Blocks (\#LINE's)} = \frac{\text{CM Size}}{\text{Block Size}}.$$

∴

Memory Organization



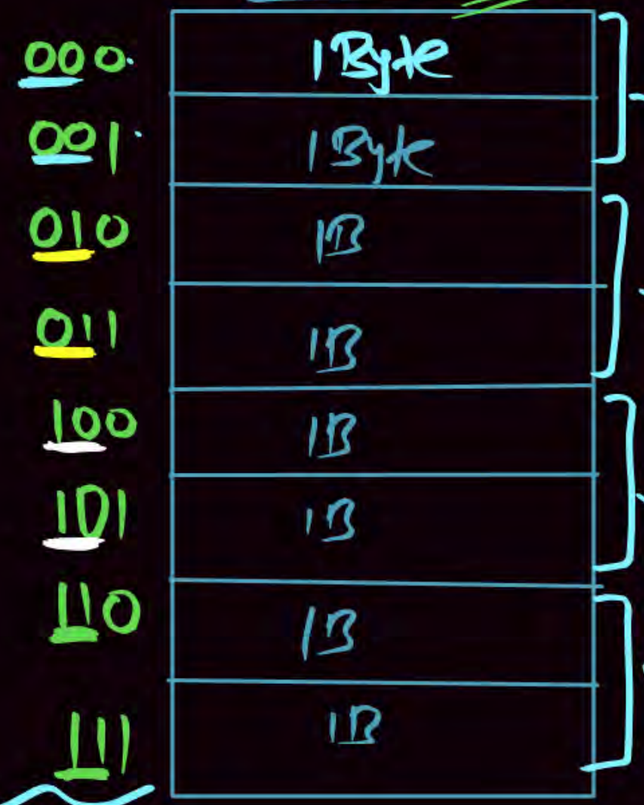
MM Size > CM Size.

Cache Memory:

Consider a 8Byte Cache with 2Byte Block size then CM org as:

Before Org.

CM = 8Byte



AFTER ORG.

$$\#CM\,LINES\ (\#CM\,Blocks) = \frac{CM\,Size}{Block\,Size}$$

$$= \frac{8B}{2B} = 4$$

$$\#CM\,LINE = 4$$

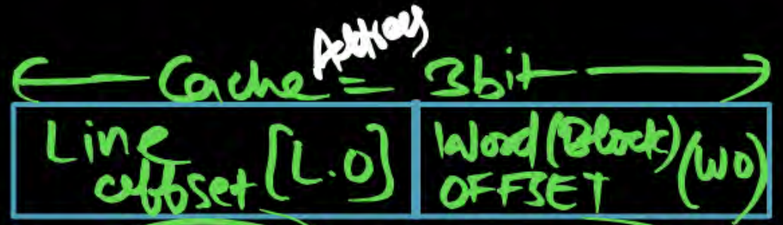
$$CM\,Size = 4 \times 2Byte$$

$$CM\,Size = 8Byte$$

After Org

Cache Memory

Before & After the organization Cache Memory Capacity is same but Internal structure is Different.



$\lceil \log_2 \#LINES \rceil$

$\lceil \log_2 4 \rceil$

2bit

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

$= \lceil \log_2 2 \rceil = 1bit$

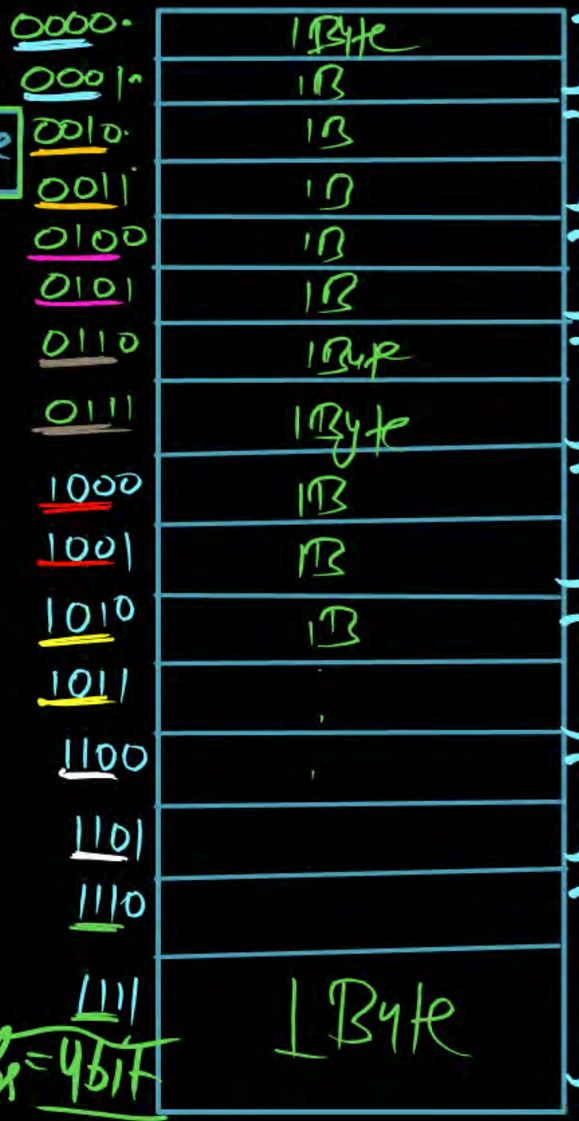
$\therefore$

Main Memory

Consider a main memory of 16 Byte with Block size of 2 Byte. the MM org shown as:

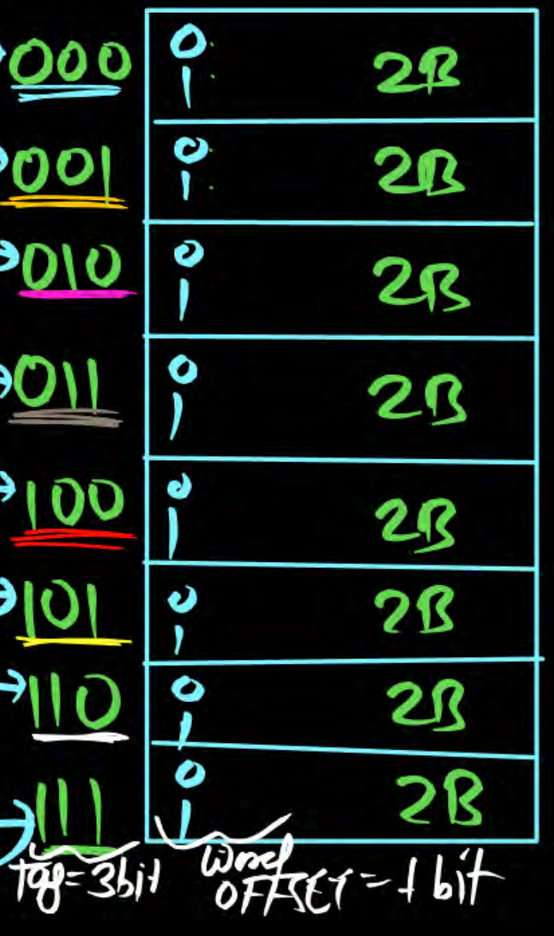
Before org

MM = 16 Byte



AFTER ORG.

$$\#\text{MM Blocks} = \frac{\text{MM size}}{\text{Block size}} = \frac{1\text{GB}}{2\text{B}} = 8$$



AFTER org:

MM Size = #Block x Block size

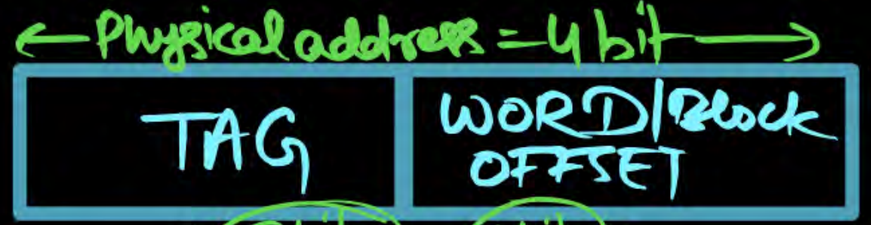
⇒ 8 x 2Byte

MM Size = 16 Byte.

Main Memory

Before & After the Organization main memory Capacity is Same but Internal structure is Different

Now Physical Address interpreted as:



$\lceil \log_2 8 \rceil$

Tag = 3bit

$\lceil \log_2 \#MM\ Block \rceil$

↳ $\lceil \log_2 Block\ size \rceil$

↳ $\lceil \log_2 2 \rceil \Rightarrow 1bit$

∴

Q.1 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
(ii) # MM Blocks
(iii) CM Address format
(iv) MM Address format

Soln

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

$2^6 \cdot 2^{10}$ B
64 KB

Block Size = 32 Byte

Block/Word offset = $\lceil \log_2 32 \rceil$

= 5 bit

Physical address = 32 bit

MM Size = 2^{32} Byte $\Rightarrow 2^2 \cdot 2^{30}$ B

$\Rightarrow$ 4G Byte

(i) #CM Blocks (#LINES) = CM Size / Block Size

$\Rightarrow \frac{64KB}{32B}$ or $\frac{2^{16}B}{2^5B}$

$\Rightarrow 2^{11} \Rightarrow 2k$ (2048)

#LINES = 2k

L.O = 11 bit

(ii) #MM Block = MM Size / Block Size

$= \frac{4GB}{32B}$ or $\frac{2^{32}B}{2^5B}$

#MM Block = 2^{27} Block

= 128 M Block Ans

(iii)

16 − 5 = 11

← Cache = 16 bit →	
LINE OFFSET	word offset
11 bit	5 bit

(iv)

← P.A = 32 bit →	
TAG	W.O
27 bit	5 bit

TAG = 27 bit

TAG = 32 − 5 = 27

Q 32 Byte is given then How write 5 bit?

$\lceil \log_2 32 \rceil$

$\lceil \log_2 2^5 \rceil$

$\Downarrow$

5 bit

$2^n = 32$

$2^n = 2^5$

n = 5 bit

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

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

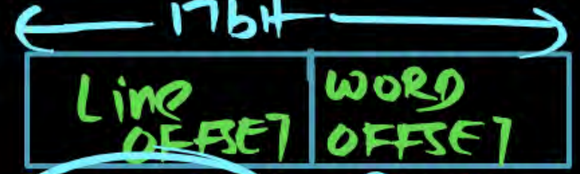
Soln

MM Size = 256MB $\Rightarrow 2^{28}$ Byte

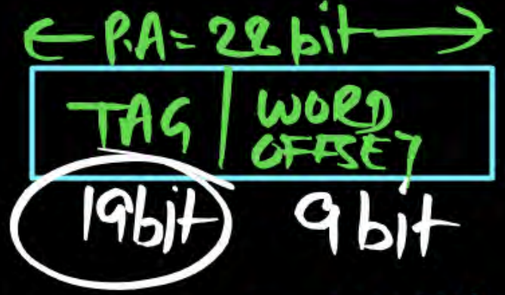
Cache Size = 128KB = 2^{17} Byte

Block Size = 512B = 2^9 Byte $\Rightarrow$ W.O = 9bit

17 − 9 = 8



$\#LINES = 2^8 =$ 256 lines



$\#MM\ Block = 2^{19} =$ 512K Blocks

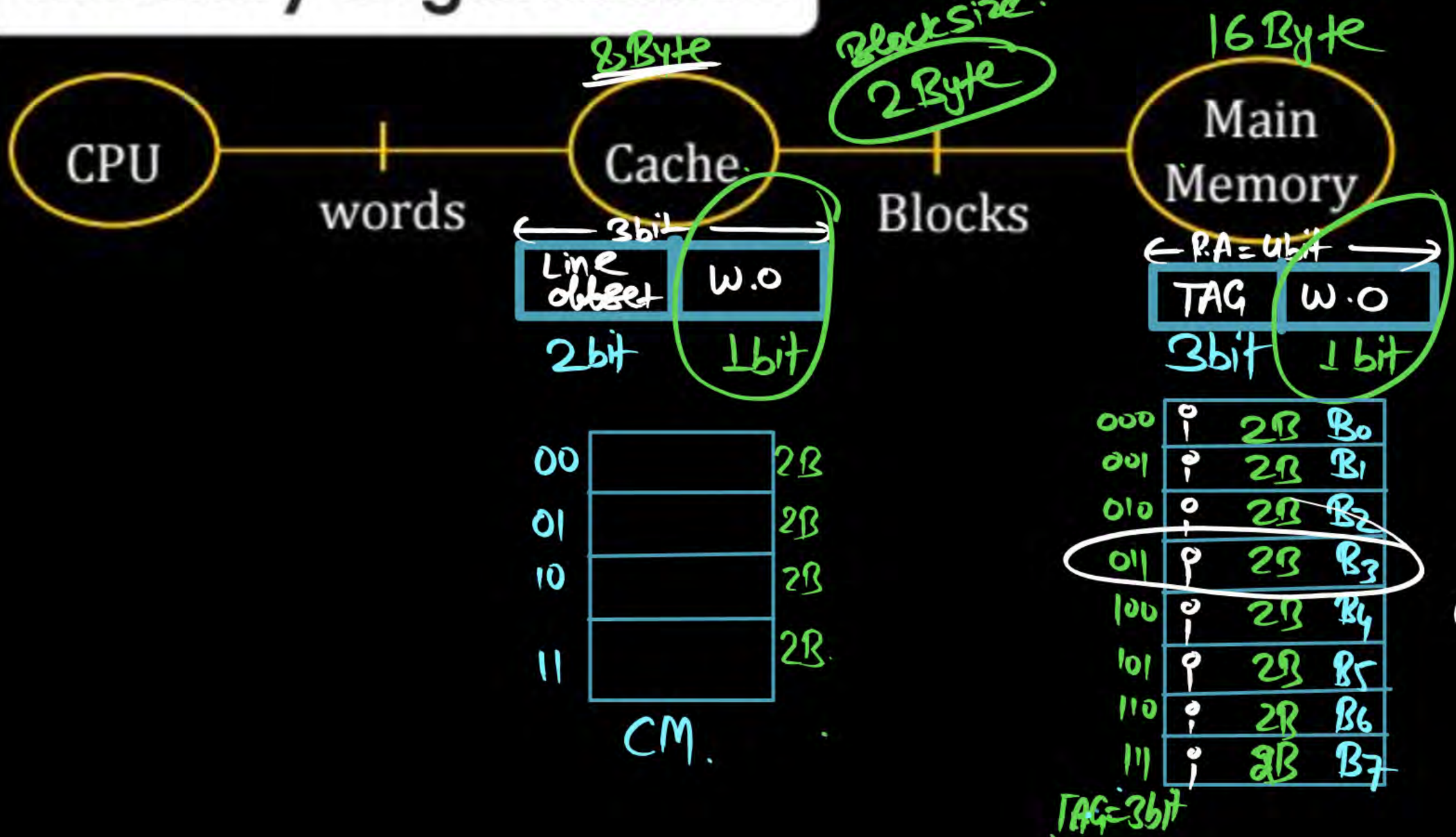
Mapping

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

Memory Organization

MM Size > CM Size.



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.

2 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

3 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 Physical address as:

This TAG is Different from the Previous.

Direct Mapping

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

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

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

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

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

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

Q.1

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

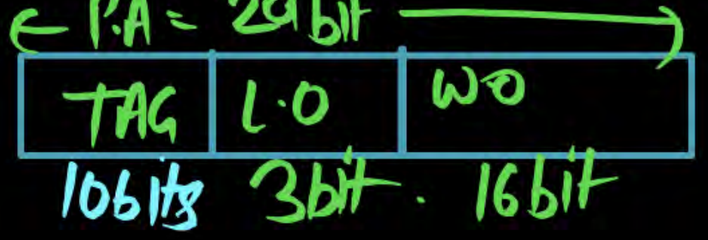
(i) P.A (29) (ii) TAG (10) (iii) L.O (3) (iv) W.O (16)

(v) #LINES (8) (vi) TAG Memory Size. (80 bit)

Cache Size = 512kB $\Rightarrow 2^{19}$ Byte

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

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



$\#LINES = \frac{CM\,Size}{Block\,size}$

$\Rightarrow \frac{512kB}{64kB} = \frac{2^{19}B}{2^{16}B}$

= 8 Lines

Line offset = 3 bit

TAG memory Size = # Lines x Tag bits

$\Rightarrow$ 8 x 10

= 80 bits


Now
Direct
Approach

Q.1 Consider a Direct Mapping if the size of Cache memory is 512KB & Main Memory 512MB & Cache line size (Block) is 64KB the calculate the number of bit required for

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

(v) #LINES (8) (vi) TAG Memory Size. (80 bits)

19bit

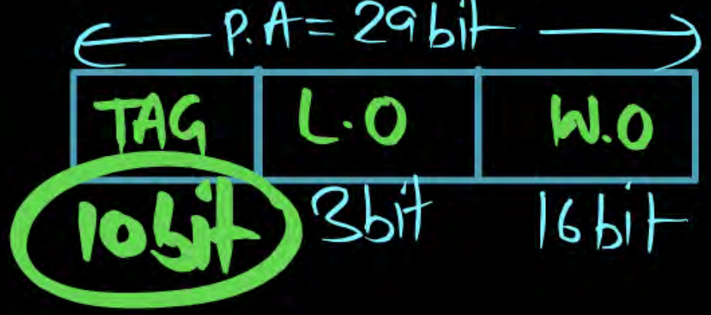
29bit

W.O = 16bit

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

$$\Rightarrow \frac{2^{19}B}{2^{16}B} = 2^3 \text{ lines}$$

L.O = 3bit



TAG Memory Size = #LINES × Tagbit

$\Rightarrow 8 \times 10 = 80$ bits Ans

Q.2 Consider a Direct Mapping, Cache size = 64 byte, Line Size = 8 B, Byte. MM = 256 Byte (8bit) then #bits for P.A, TAG, L.O, W.O Tag memory size?

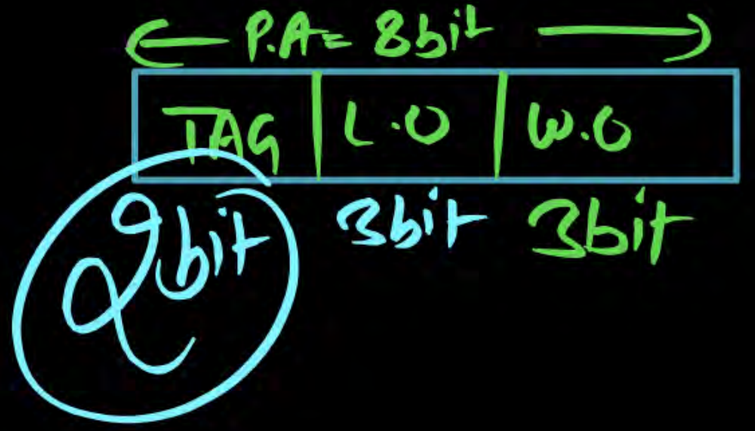
W.O = 3bit

8 2 3 3 16

$$\#LINE = \frac{CM\,Size}{B.S}$$

$$\Rightarrow \frac{64B}{8B} = 8\ Lines$$

L.O = 3bit



Tag Memory Size = #LINES × Tag bits

$\Rightarrow 8 \times 2$

$= 16$ bit

Q.

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

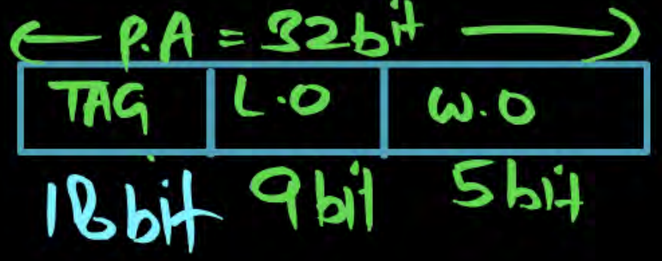
Q.

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]

18 Ans

L.O = 9



TAG = 32 − (9 + 5)

= 32 − 14

= 18 bits Ans

1) Direct Mapping

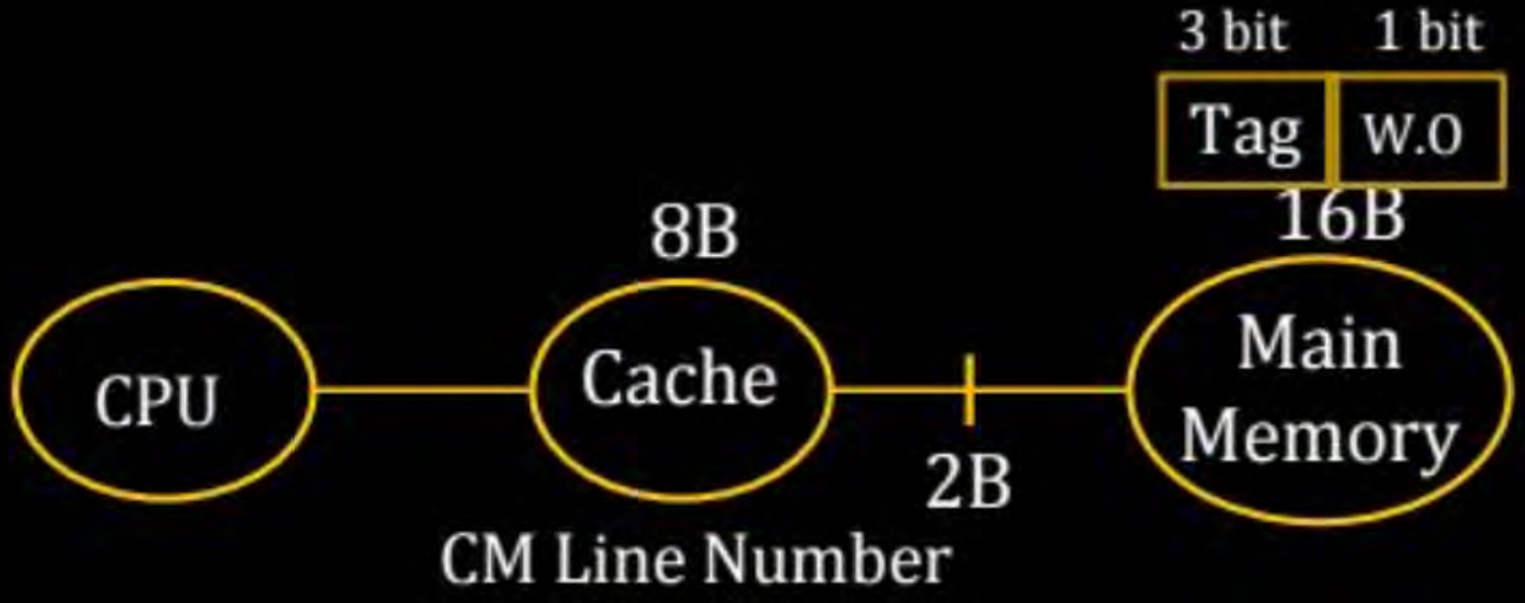
In this Technique mapping function is used to transfer the data from Main Memory to Cache Memory. The Mapping Function is

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

(Or)

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

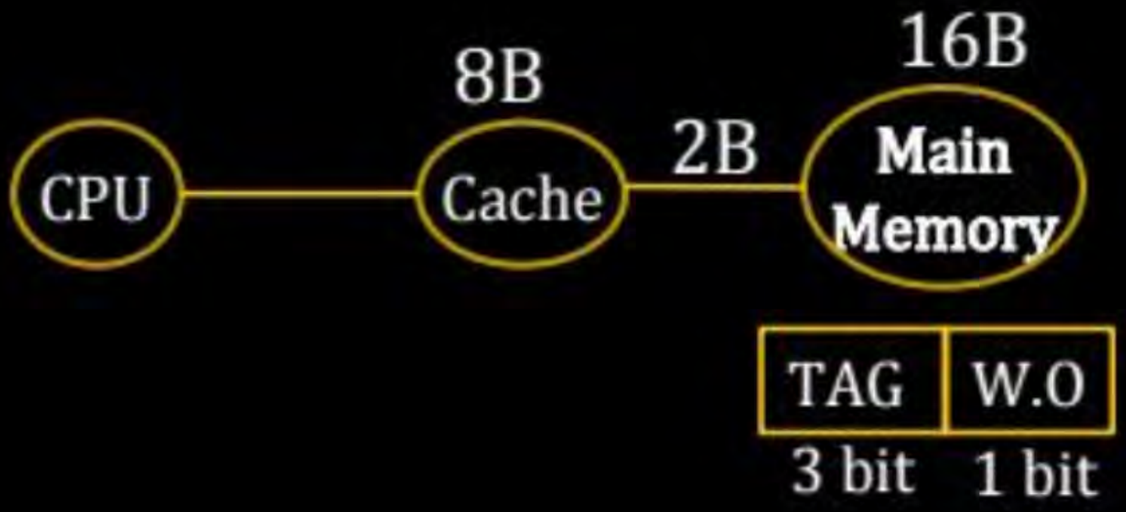
K: MM Block No.
N: # of Cache Line
i: CM Line Number

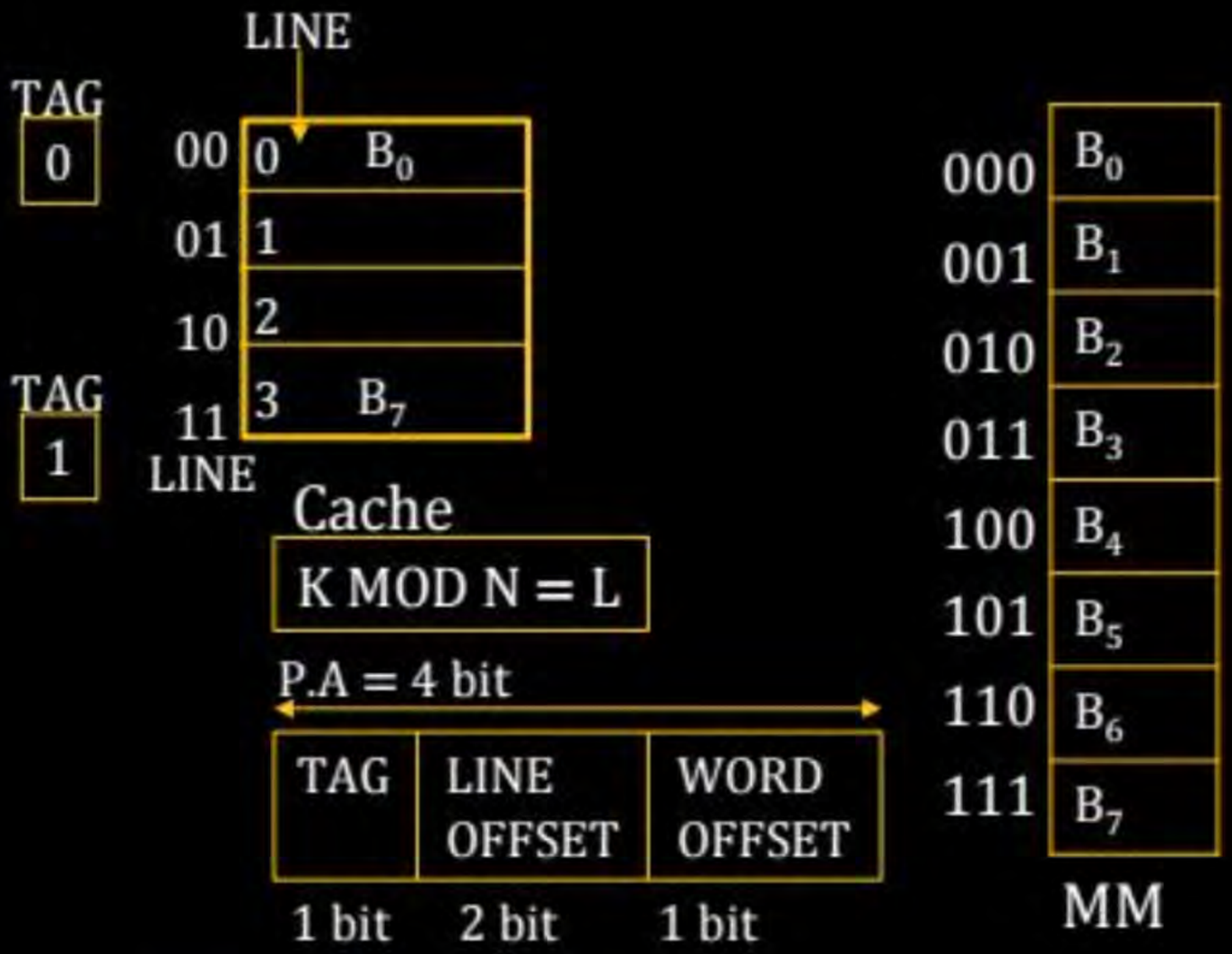


0 MOD 4 = 0
1 MOD 4 = 1
2 MOD 4 = 2
3 MOD 4 = 3
4 MOD 4 = 0
5 MOD 4 = 1
6 MOD 4 = 2
7 MOD 4 = 3

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


16B
8B
2B
CPU
Cache
Main Memory
TAG
W.O
3 bit
1 bit



Direct Mapping

MM Block | Direct Mapping | CM LINE

TAG	LINE
0	00

$B_{0_{[000]}}$

K MOD N = i

0 MOD 4 = '0'

LINE '0'

TAG	LINE
1	11

$B_{7_{[111]}}$

K MOD N = i

7 MOD 4 = '3'

LINE '3'

Tag Memory Size = #LINE's × Tag bits [Depends On the Mapping technique]

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

Tag Memory Size = 4 × 1 = 4 bits

Consider the following program

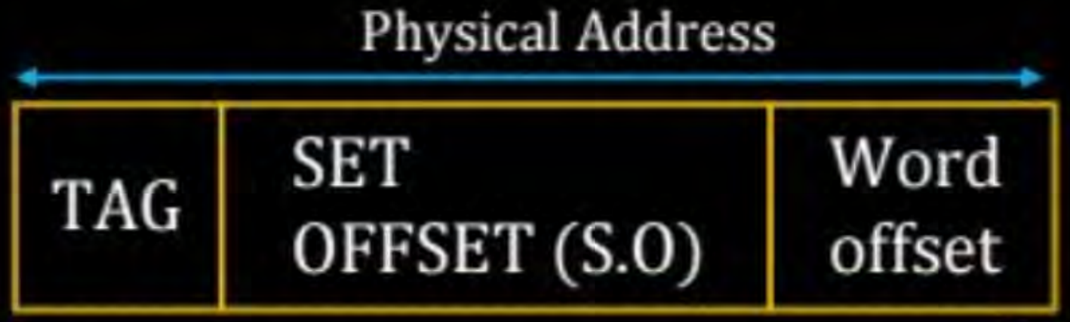
I_1: MOV r_0 [0000]

I_2: MOV r_1 [1000]

I_3: ADD $r_0 r_1$

2) Set Associative Cache

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



Word Offset = $\log_2$ Block Size

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

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

SET OFFSET = $\log_2$ #SETS

TAG = Physical address - (S.O + 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

1. (i) P.A (ii) TAG (iii) L.O (iv) W.O

(2) #lines (3) TAG memory size

Q.

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

Q. The main memory of a computer has 2 cm blocks while the cache has 2 c blocks. If the cache uses the set associative mapping scheme with 2 blocks per set, then block k of the main memory maps to the set. [GATE - 1999]

(a) (k mod m) of the cache
(b) (k mod c) of the cache
(c) (k mod 2 c) of the cache
(d) (k mod 2 cm) of the cache

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]

Q.

[Common Data for this and next question]
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]
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

[GATE - 2014]

THANK YOU!