COMPUTER SCIENCE

Computer Organization and Architecture

Cache Memory

Lecture_07

Vijay Agarwal sir

01 Mapping Techniques

02 Replacement Algo & Updating Technique

Cache Memory

① Memory Org
- CM

L.O	W.O

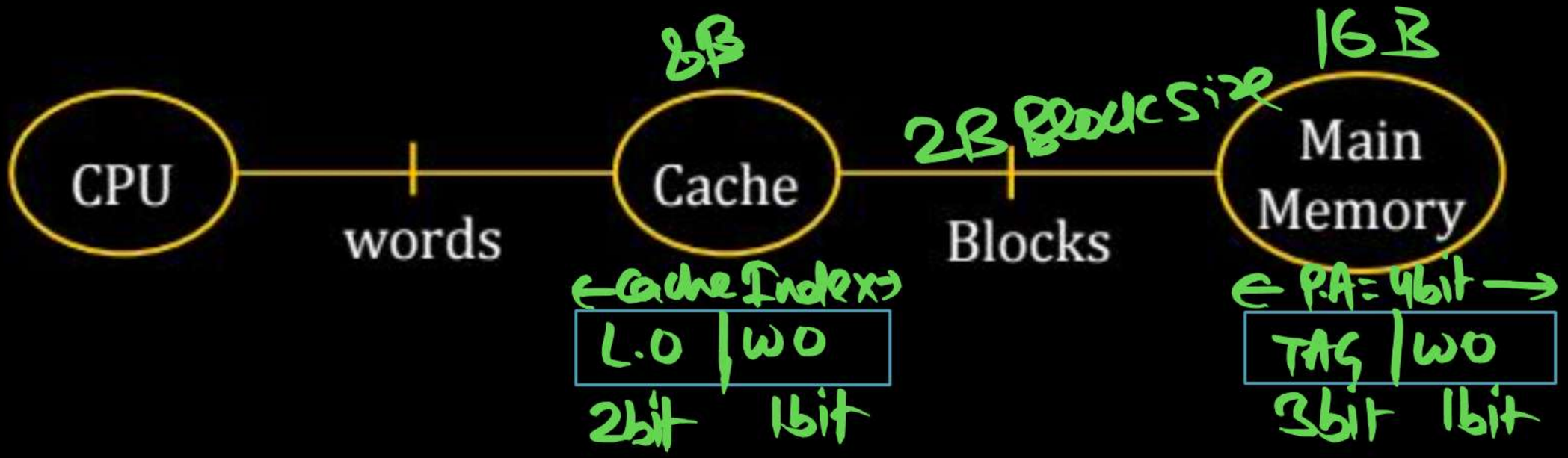
- Main Memory

TAG	W.O

② Mapping Technique. MM to CM

1. Direct Mapping
2. Set Associative Mapping
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	Associative	Set Associative
• The simplest technique • Maps each block of main memory into only one possible cache lines.	• 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	• 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:

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

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

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

TAG = Physical Address - (Line offset + Word offset)

TAG Memory Size = #LINE's × Tag bits (Depend on the mapping technique)

1) Direct Mapping

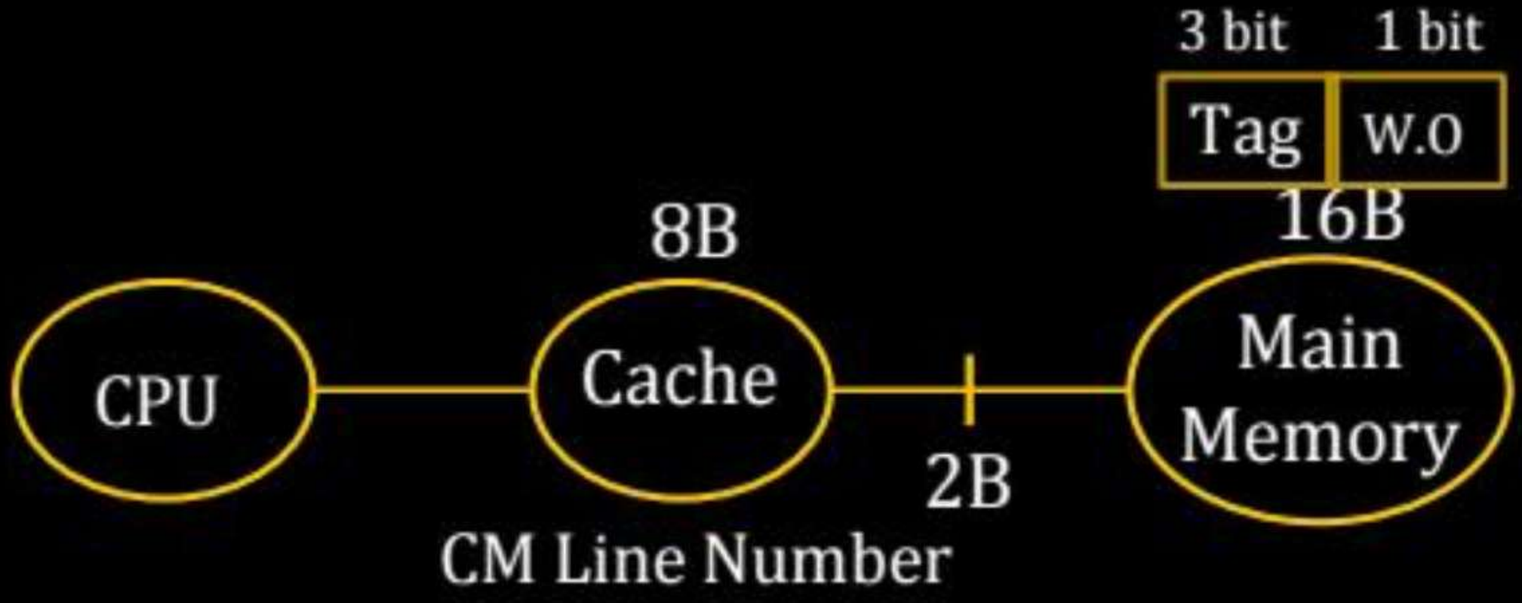
In this Technique mapping function is used to transfer the data from Main Memory to Cache Memory. The Mapping Function is

Cache address = Main Memory request MOD # CM LINES

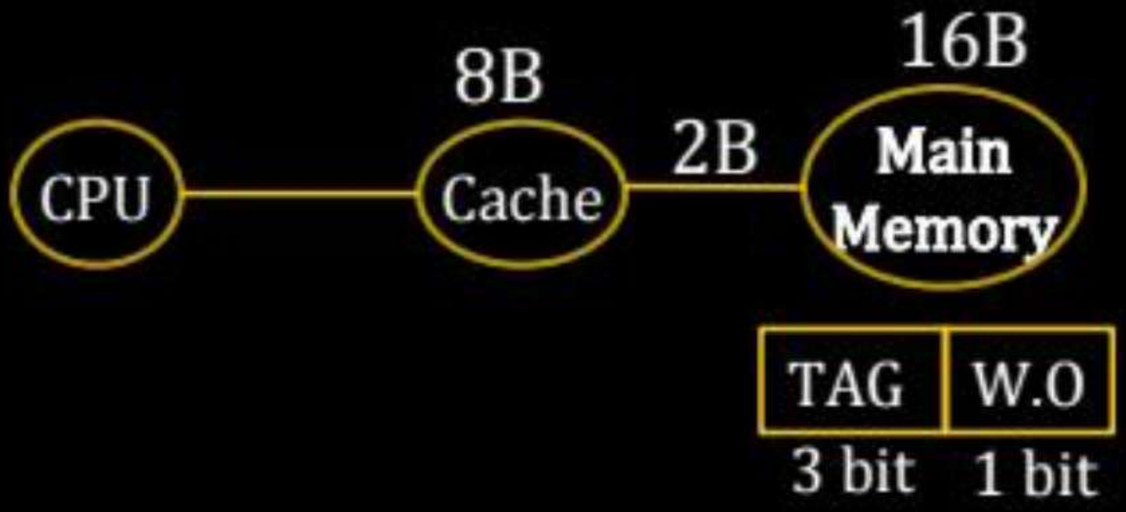
(Or)

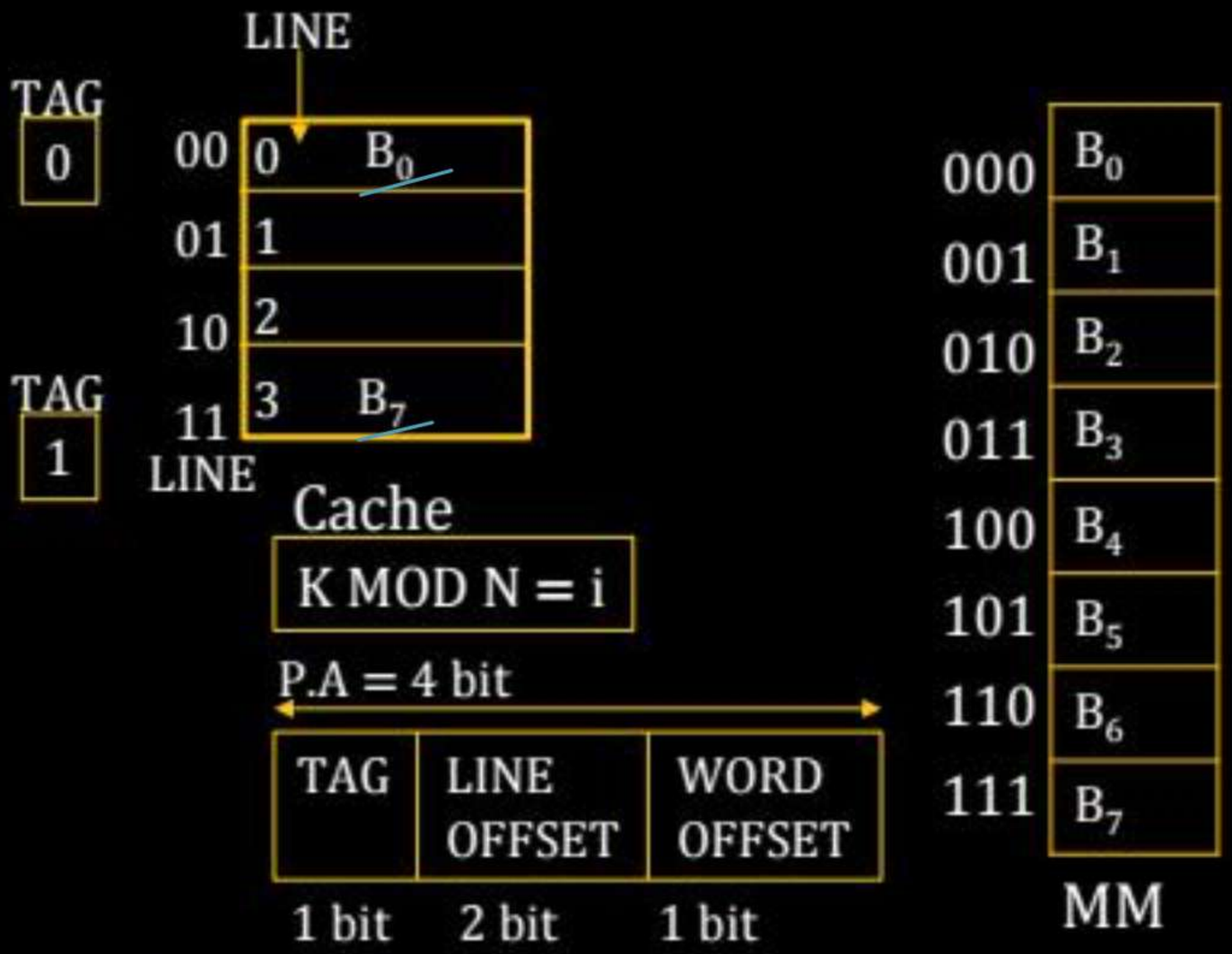
K MOD N = i

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



0 MOD 4 = 0	CM
1 MOD 4 = 1	B_0 & B_4: LINE 0
2 MOD 4 = 2	B_4 & B_5: LINE 1
3 MOD 4 = 3	B_2 & B_6: LINE 2
4 MOD 4 = 0	B_3 & B_7: LINE 3
5 MOD 4 = 1	
6 MOD 4 = 2	
7 MOD 4 = 3	


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



Direct Mapping

MM Block

TAG	LINE
0	00

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

Direct Mapping

K MOD N = i

0 MOD 4 = '0'

CM LINE

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

LOR [Locality of Reference]

Consider the following program

I_1: MOV r_0 [0000]

I_2: MOV r_1 [0001]

I_3: MOV r_2 [1000]

I_4: MOV r_3 [1001]

Disadvantage of Direct Mapping

I_1: MOV r_0 [0000] B0

I_2: MOV r_1 [1000] B4

I_3: MOV r_2 [0001] B0

I_4: MOV r_3 [1001] B4

I_5: MOV r_4 [0000] B0

I_6: MOV r_5 [1000] B4

Conflict Miss Increase.

2) Set Associative Cache

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

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

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

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

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

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

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

Set Associative Mapping function

Cache Set Address = MM Request MOD #SET is in Cache

(OR)

K MOD S = i

k: MM Block Number
s: # Cache Set
i: Cache Set Number

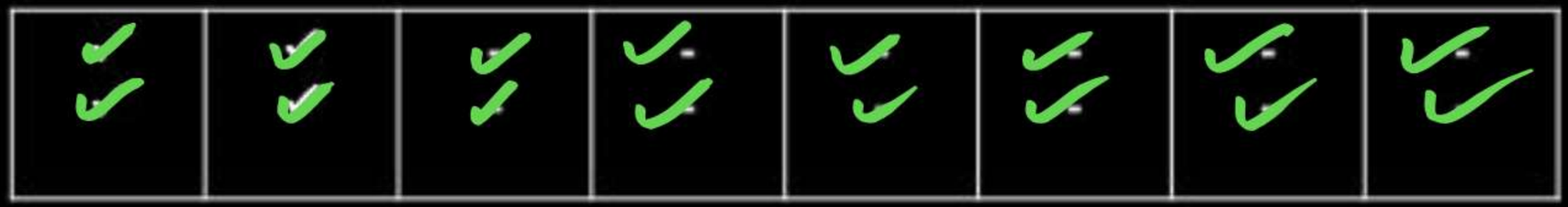
Example1: # Line's = 16 & 2 way set Associative

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

#SET = 8 S = 8

K MODS = i

K MOD 8 = i



SET_0 SET_1 2 3 4 5 SET 6 SET 7

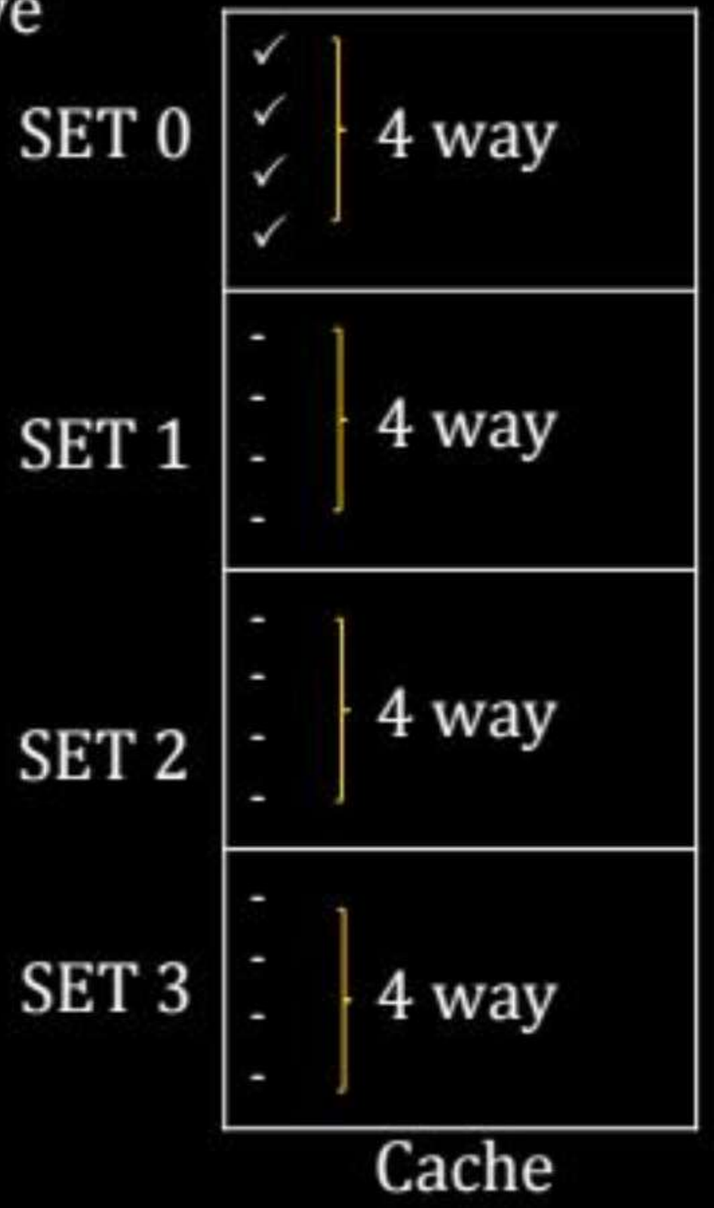
K MOD 8 = i 0 - 7

Example2:

#LINE = 16 & 4 way set Associative

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

K MOD 4 = i



Example3:

#LINE = 16 & 8 way set Associative

SET = $\frac{16}{8} \Rightarrow 2$ S = 2

K MOD 2 = i

Example4:

#LINE = 16 & 16 way set Associative

$\#\,SET = \frac{16}{16} \Rightarrow 1$ S = 1

K MOD 1 = i

#LINE=16

16 Way

S=1

Fully Associative (Whole cache as a set)

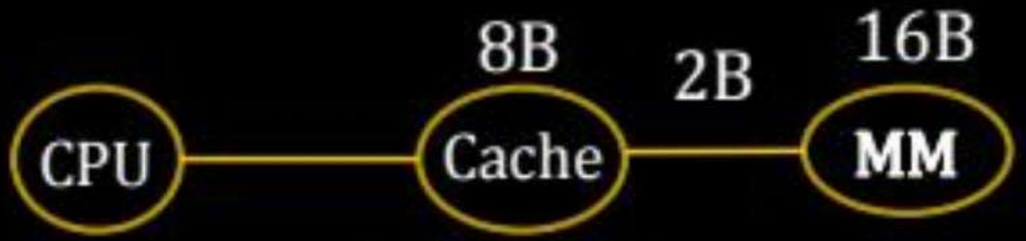
Set Associative Mapping function

Cache Set Address = MM Request MOD #SET is in Cache

(OR)

K MOD S = i

k: MM Block Number
s: # Cache Set
i: Cache Set Number

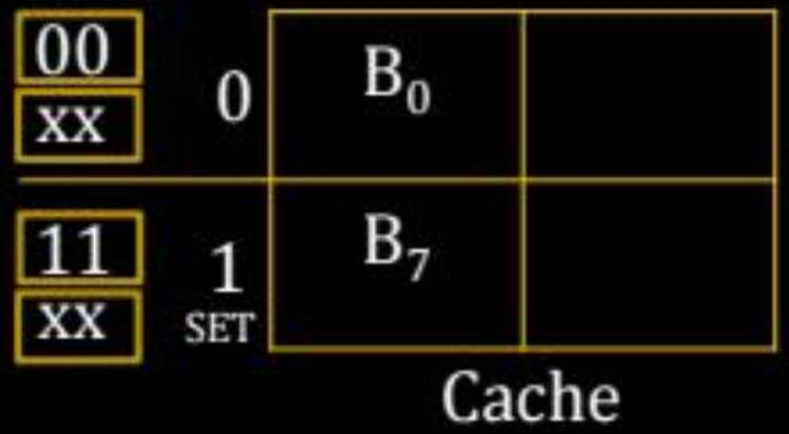


2 way set associative

K MOD S = i

P.A

TAG	S.O	W.O
2 bit	1 bit	1 bit



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

$B_{0[000]}$ →

TAG	S.O
00	0
2 bit	1 bit

→ SET '0'

$B_{7[111]}$ →

TAG	S.O
11	1

→ SET '1'

MM BLOCK Mapping Tech. CM SET

$B_{0[000]}$ $\xrightarrow[0 \text{ MOD } 2 = \text{'0'}]{K \text{ MOD } S = i}$ SET '0'

$B_{7[111]}$ $\xrightarrow[7 \text{ MOD } 2 = \text{'1'}]{K \text{ MOD } S = i}$ SET '1'

BLOCK

Tag Memory Size = #SETS × #LINES In a each set × Tag bits

Example: # SET = 2 # LINE in Each set =2 TAG = 2 bit

Tag Memory size = 2 × 2 × 2 = 8 bits

1) In direct Mapping

Hit Latency = Latency of Tag comparator

2) In Set Associative Mapping

Hit Latency = Latency of Tag comparator + Latency of Multiplexer

Set Associative Mapping:

k MOD S = i

$$S = \frac{\#LINES}{N\text{-ways}}$$

From Mapping onwards

29 GATE QUESTION

+ Practice Question

Important Points About Set Associative

#LINE = L & N way set Associative

$$\#\ SET[S] = \frac{L}{N}$$

If N=1; Direct Mapping

If N=L ; Associative Mapping; (S=1)ie Only 1 Set

K MOD S = i

k: MM Block Number
s: # Cache Set
i: Cache Set Number

1 way

N=1

#LINE = 8
1 way
#SET = 8/1 = 8
Direct Mapping.

Fully

#LINE = 8
8 way set Associative
#SET = 8/8 = 1 [Fully Associative]

Example4:

#LINE = 16 & 16 way set Associative

$\#\,SET = \frac{16}{16} \Rightarrow 1$ S = 1

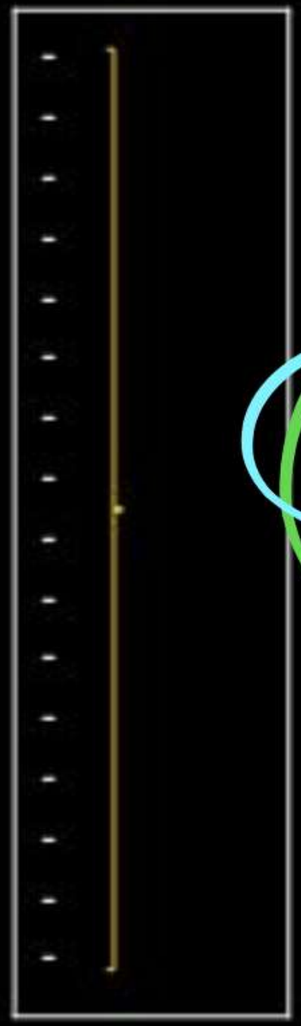
K MOD 1 = i

L = N

(L) #LINE = 16

(N) 16 Way

S = 1



Fully Associative (Whole cache as a set)

Important Points About Set Associative

#LINE = L & N way set Associative # SET[S] = $\frac{L}{N}$

If N=1; Direct Mapping

If N=L ; Associative Mapping; (S=1)ie Only 1 Set;

Set Associative Mapping Follows Direct Mapping as well as ~~Set~~ Fully Associative Mapping. Using Mapping Function as:

K MOD S = i

within the Set MM Block Can be placed anywhere.

- **In the Set Associative Mapping Replacement Algorithm is required along with the Mapping function when cache set is Full.**

Important Points About Set Associative

#LINE = L & N way set Associative

SET[S] = $\frac{L}{N}$

If N=1; Direct Mapping

If N=L ; Associative Mapping; (S=1)ie Only 1 Set.

V.V Imp Tag Bits in N way Set Associative Mapping:

N: N way set Associative

Tag bits in N Set Associative = Tag bits in Direct Mapping + $\log_2 N$

Important Points About Set Associative

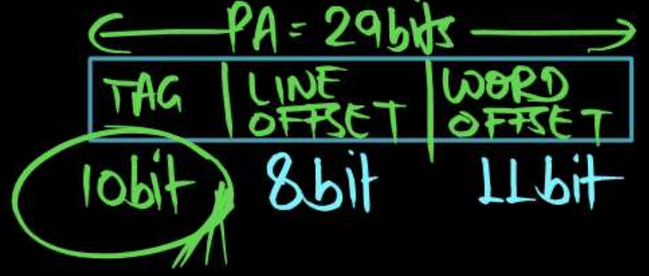
Tag bits in N Set Associative = Tag bits in Direct Mapping + $\log_2 N$

Eg. Consider a **Direct Mapping** if the size of cache memory is 512KB & Main Memory 512MB & Cache line size is 2KB then calculate the Number of bit Required for TAG?

29bit

11bit

19bit



$$\#LINES = \frac{CM\ Size}{Block\ Size} = \frac{512KB}{2KB} = \frac{2^{19}}{2^{11}} = 2^8$$

L.O = 8bit

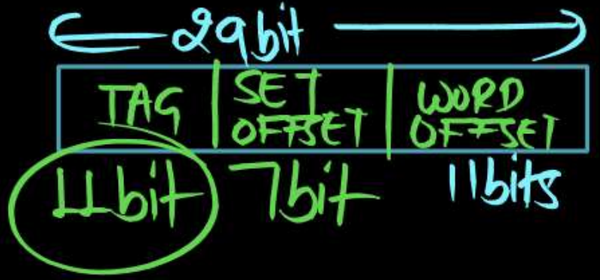
Tag bit in Direct Mapping = 10bit

Important Points About Set Associative

Tag bits in N Set Associative = Tag bits in Direct Mapping + $\log_2 N$

Q.1 Eg. Consider a **2-way set associative** if the size of cache memory is 512KB & Main Memory 512MB & Cache line size is 2KB then calculate the Number of bit Required for TAG?

2 way Set Associative



$\#LINES = 2^8$

$\#SET = 2^8/2$ (2^7 SET)

So = 7 bit

2 way Set Associative

Tag bit = 10 bit + $\log_2 1$ (Direct mapping)

= 10 + 1

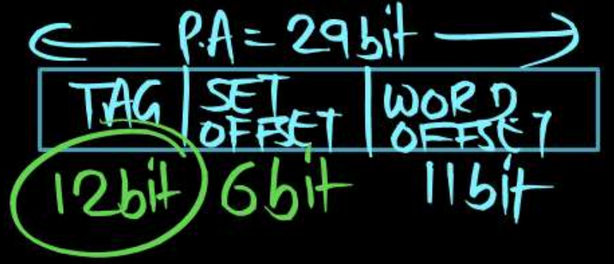
Tag = 11 bits

Important Points About Set Associative

Tag bits in N Set Associative = Tag bits in Direct Mapping + $\log_2 N$

(Q2) Eg. Consider a **4-way set associative** if the size of cache memory is 512KB & Main Memory 512MB & Cache line size is 2KB then calculate the Number of bit Required for TAG?

4 way set Associative.



#LINES = 2^8

#SET = $2^8/2^2 = 2^6$ SET

S.O = 6 bit

4 Way Set Associative

Tag bit = $10 + \log_2 4$

$= 10 + 2$

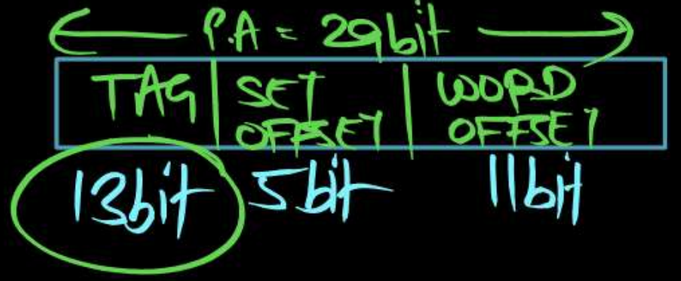
$= 12$ bit

Important Points About Set Associative

Tag bits in N Set Associative = Tag bits in Direct Mapping + $\log_2 N$

Q.3 Eg. Consider a **8-way set associative** if the size of cache memory is 512KB & Main Memory 512MB & Cache line size is 2KB then calculate the Number of bit Required for TAG?

8 Way Set Associative



$\#LINE = 2^8$

$\#SET = 2^8 / 2^3 = 2^5$ Set

SO = 5bit

8 Way Set Associative

Tag bit = $10 + \log_2 8$

$= 10 + 3$

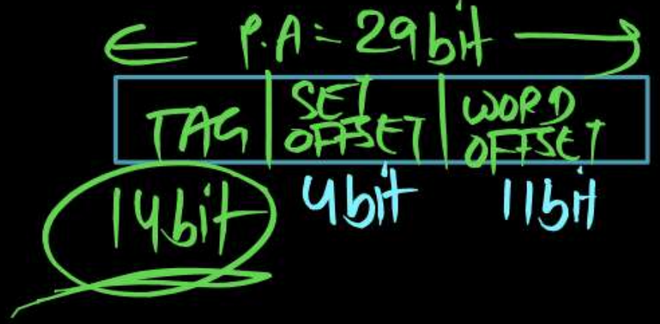
Tag = 13bit

Important Points About Set Associative

Tag bits in N Set Associative = Tag bits in Direct Mapping + $\log_2 N$

Q.4 Eg. Consider a **16-way set associative** if the size of cache memory is 512KB & Main Memory 512MB & Cache line size is 2KB then calculate the Number of bit Required for TAG?

16 Way Set Associative



#LINE = 2^8

#SET = $2^8 / 2^4 = 2^4$ Set

SO = 4bit

16 Way Set Associative

Tag bit = 10 + $\log_2 16$

= 10 + 4

Tag = 14 bit

③ Associative Mapping :

Till
71 Pipeline
10 Approx in Cache
29 Today Homework

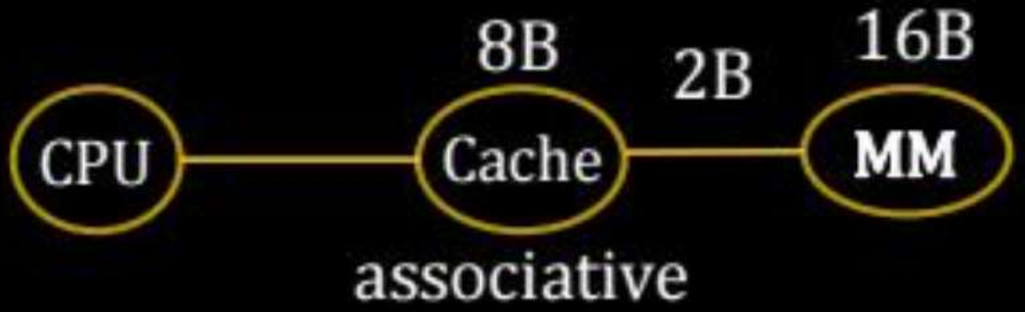
110 PYQ's.

3) Associative Mapping

- In this no mapping function is used to transfer the data from MM to CM.
- Any Block of Main Memory can be placed Anywhere in Cache Memory.
- No Conflict Miss;
- This Cache is designed without address called as Content Addressable Memory.
- In Associative Mapping More Tag bits is required & More Tag Memory size.
- In Associative Mapping More Hardware Required, Expensive(N Tag Comparator, Here N is Number of Lines., For Each line Comparator Required)
- In Associative Cache Design, Counter Sequence is used to map the Data, means any Main Memory Block any cache memory line in a sequence.
- In Associative Mapping Physical Address is Interpreted as:

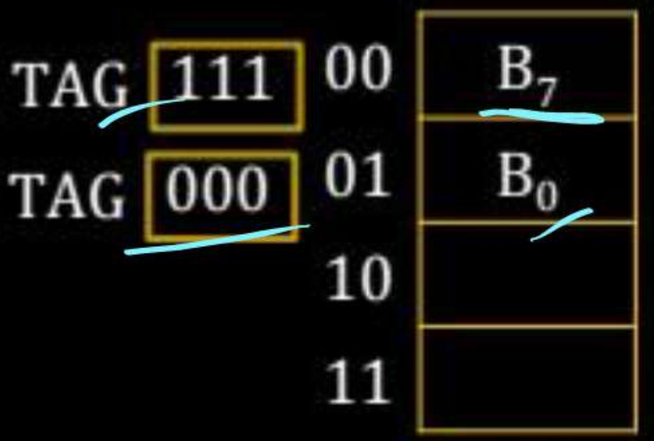
3) Associative Mapping

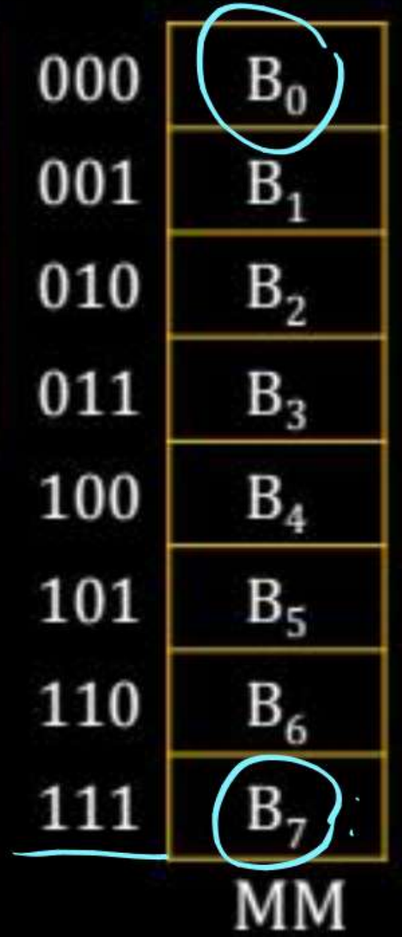
- In this no mapping function is used to transfer the data from MM to CM.
- No Conflict Miss;



(Fully Associative)

4 CM LINE ⇒ 4 Tag Comparator Required





MM Block	Associative Mapping	CM Line
$B_{7[111]}$	No mapping Function	Any line
$B_{0[000]}$	No mapping Function	Any line

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

Example: #LINE = 4 & Tag bits = 3

Tag Memory Size $= 4 \times 3 = 12$ bits

① In Direct mapping

Tag memory size $= 4 \times 1 = 4$ bit

② In 2 way set Associative

Tag memory size $= 4 \times 2^{(\text{Tag bit})} = 8$ bit

③ In fully Associative

Tag memory size $= 4 \times \underset{\text{Tag bit}}{3} = 12$ bit

In Direct Mapping

$$\text{Tag bit} = \frac{\text{MM Size}}{\text{CM Size}}$$

Directly Calculate Tag bit

Today I told Directly in Set Associative Mapping

$$\text{Tag bit} = \#\text{Tag bit in Direct Mapping} + \log_2 N$$

N: N way Set Associative

I Mixing type of Question.

Q. Consider a 64KB Direct Mapped Cache organized into a 64 word blocks. Word length of the CPU is 32bits. Main Memory 4GB. In the Cache Controller is comprising of 1 Valid bit & 1 Update bits.

the calculate the number of bit required for

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

(v) #LINES **(vi) TAG Memory Size.**

Direct Mapped Cache Controller: 1 Valid bit + 1 Updated bit

Cache Size = 64KB [2^{16}B]

Block Size = 64 Words

[1] Word length (Size) = 32 bit

Main Memory = 4GB [2^{32}B]

← PA = 32 bit →

TAG	LINE OFFSET	WORD OFFSET
16 bit	8 bit	8 bit

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

$\Rightarrow \frac{2^{16}\text{B}}{2^{8}\text{B}}$

L.O = 8 bit = 2^{8} Lines

Block Size = 64 Words

1 Word Size = 32 bit = 4 Byte

Block Size = 64 × 4 Byte $\Rightarrow 2^{6} \times 2^{2}$ B

$\Rightarrow 2^{8}$ Byte

Word offset = 8 bit

Tag entry Size = Tag bit + extra bit (if given)

$\Rightarrow 16 + 1 + 1$

Tag entry Size = 18 bit

Tag Memory Size = #LINE × Tag bits

$= 2^{8} \times 18$

$= 256 \times 18$ bit

$= 4608$ bits

Q.

Consider fully associative cache consists 8 Block & MM contain 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?

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

Total Miss = 12

Cache Work on Locality of Reference.

1. Memory Cog.
2. Mapping (MM to CM) Technique
   - (1) Direct Mapping
   - (2) Set Associative Mapping
   - (3) Fully Associative Mapping.
3. Replacement Algorithms.

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 [Least Recently Used]

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

In the Cache There are 3 Type of Miss.

1. Compulsary | First Reference | Cold Start Miss.
2. Capacity Miss
3. Conflict Miss | Collision Miss.

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 (Collison miss / reference miss)

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

Q.1 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

Number of Cache (Block) Line = 4

MM Block Request : 7, 8, 10, 15, 7, 8, 16, 7, 8, 10

↳ Total Request (Access) = 10

Q.1 Consider 4 block cache memory (initially empty) with the following MM block references.

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

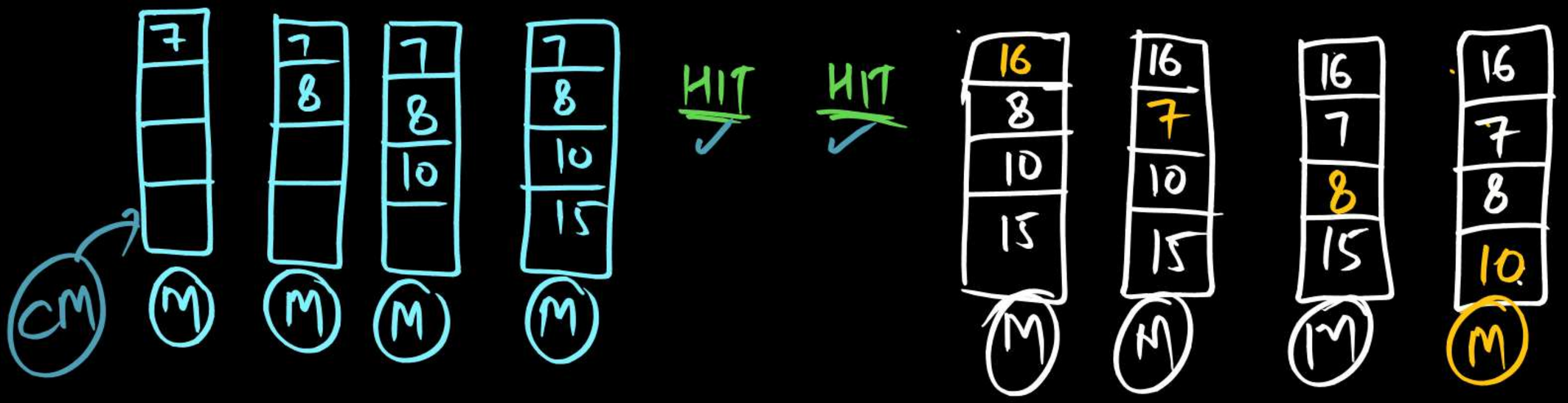
Identify the Hit Ratio using

(i) FIFO [FIRST IN FIRST OUT]

M: Miss

H: Hit

7 8 10 15 7 8 16 7 8 10



#HIT = 2

#MISS = 8

HIT Ratio = 2/10

= 0.2

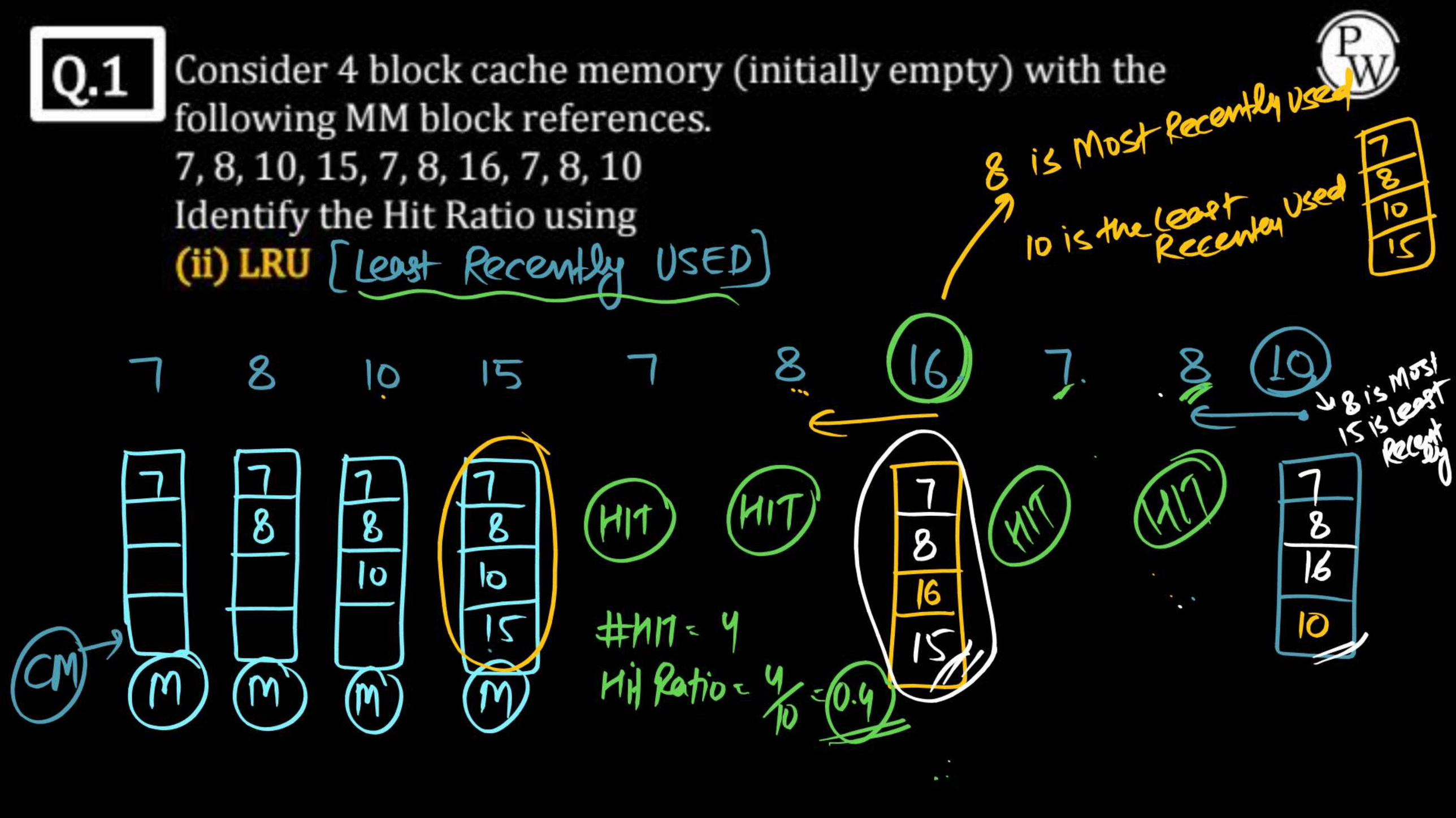
Ans

Q.1 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

(ii) LRU [Least Recently USED]



Q.1 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

(iii) Direct Mapped cache

% MODULUS

k MOD N = i

k MOD 4 = i

			Line No	
7	%4	=	3	M
8	%4	=	0	M
10	%4	=	2	M
15	%4	=	3	M
7	%4	=	3	M
8	%4	=	0	HIT
16	%4	=	0	M
7	%4	=	3	HIT
8	%4	=	0	M
10	%4	=	2	HIT

MODU

#HIT = 3

Hit Ratio = 3/10

LINE 0	~~8~~ ~~16~~ 8
LINE 1	
LINE 2	10
LINE 3	~~7~~ ~~15~~ 7

CM

Q.1 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

(iii) Direct Mapped cache

MOD US

k MOD N = i

k MOD 4 = i

Line No

Compulsory 7 → 7 MOD 4 = 3 M

Compulsory 8 → 8 MOD 4 = 0 M

10 → 10 MOD 4 = 2 M

15 → 15 MOD 4 = 3 M

Conflict 7 → 7 MOD 4 = 3 M

Conflict 8 → 8 MOD 4 = 0 → HIT

Compulsory 16 → 16 MOD 4 = 0 → M

7 → 7 MOD 4 = 3 → HIT

8 → 8 MOD 4 = 0 → M

10 → 10 MOD 4 = 2 → HIT

#HIT = 3

Hit Ratio = 3/10

	CM
LINE 0	8 ~~16~~ 8
LINE 1	~~7~~ Empty ~~15~~ 7
LINE 2	10
LINE 3	

No Capacity Miss

Q.1 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

(iv) 2 - way Set Associative with LRU

#SET = #LINE / N-ways = 4/2 = 2 Set

S = 2

k MODS

MOD 2

k MOD 2 = i [Set Number]

	Set Number	
7 % 2 =	1	M
8 % 2 =	0	M
10 % 2 =	0	M
15 % 2 =	1	M
7 % 2 =	1 [Set No 1]	→ HIT
8 % 2 =	0	→ HIT
16 % 2 =	'0' [Set No '0' But Set is Full, So Apply LRU]	M
7 % 2 =	1	→ HIT
8 % 2 =	0	→ HIT
10 % 2 =	'0' → Miss, Set Full Apply LRU	M

#HIT = 4

#Miss = 6

Hit Ratio = 4/10 = 0.4

SET 0	8 → 1 ~~10~~ ~~16~~ 10	2 way + LRU
SET 1	7 15	

2 way set Associative with LRU

Q.1 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

(iv) 2 - way Set Associative with LRU

#SET = #LINE / N-ways = 4/2 = 2 Set

S = 2

k MODS

MOD 2

k MOD 2 = i [Set Number]

	Set Number	
7 % 2 =	1	M
8 % 2 =	0	M
10 % 2 =	0	M
15 % 2 =	1	M
7 % 2 =	1 [Set No 1]	→ HIT
8 % 2 =	0	→ HIT
16 % 2 =	'0' [Set No '0' But Set is Full, So Apply LRU]	M
7 % 2 =	1	→ HIT
8 % 2 =	0	→ HIT
10 % 2 =	'0' → Miss, Set Full Apply LRU	M

#HIT = 4.
#Miss = 6
Hit Ratio = 4/10 = 0.4

SET 0	8 ~~10~~ ~~16~~ 10	2 way + LRU
SET 1	7, 15	

2 way set Associative with LRU

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]

K MODS = i

K MOD 2 = i

SET Number

CM Lines = 4

2 way set Associative

SET = 4/2 = 2

8 ! 2 = 0 M

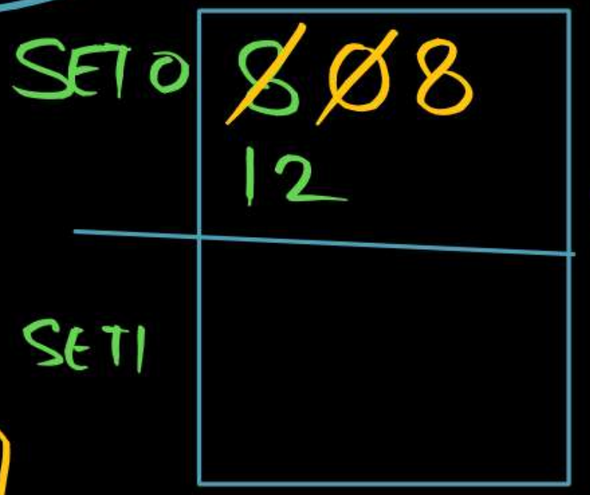
12 ! 2 = 0 M

0 ! 2 = 0 Miss (Cache Set '0' Full Apply LRU)

12 ! 2 = 0 → HIT

8 ! 2 = 0 → Miss (Cache Set '0' is Full, Apply LRU)

Miss = 4



2 way Set + LRU

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.

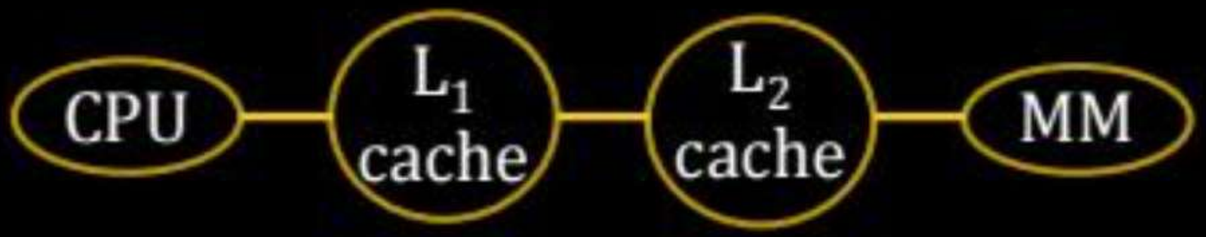
2) Capacity Miss - This miss will occur when cache is full.

3) Conflict Miss (Collison 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:

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

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

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

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 (Collison miss / reference miss)

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

Q.1 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.2 Consider a Direct Mapping, Cache size = 64 byte, Line Size = 8 Byte. MM = 256 Byte then #bits for P.A, TAG, L.O, W.O Tag memory size?

Q.3 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.4 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.5 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 $(ABCDE)_{16}$?

NAT Q.1

An 8-way set associative cache of size 64 KB (1 KB = 1024 bytes) is used in a system with 32-bit address. The address is sub-divided into TAG, INDEX, and BLOCK OFFSET.

The number of bits in the TAG is _____. [GATE-2023-CS: 2M]

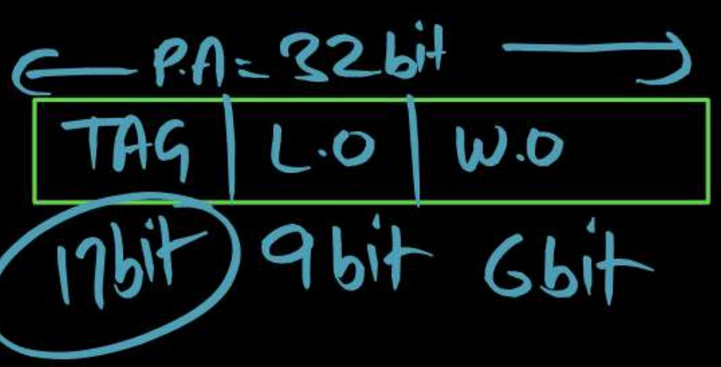
NAT **Q.2**

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.

The size of the tag field is 17 bits. Ans

$2^{15} / 2^6 = 2^9$

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



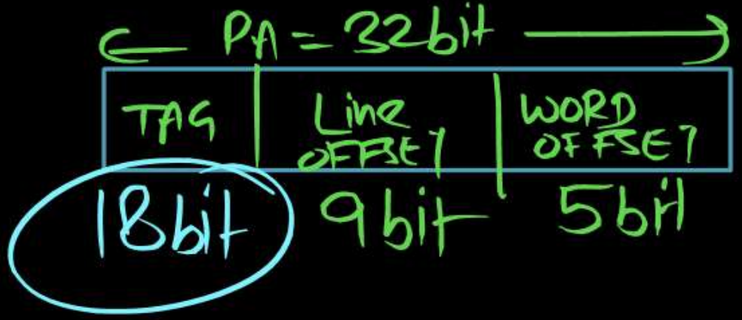
NAT Q.3

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]

29

5 bit



$$\text{Tag} = 32 - (9+5)$$

$$32 - 14$$

$$\text{Tag} = 18 \text{ bit}$$

MCQ Q.4

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

[GATE-2015(Set-3)-CS: 1M]

L.O = 12bit

PA = 20bit

W.O = 4bit

(A) E, 201

Ans(A)

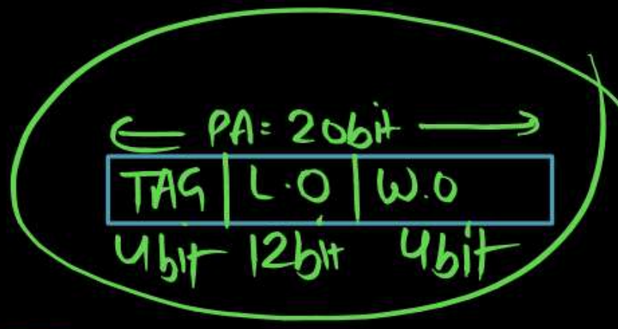
(B) F, 201

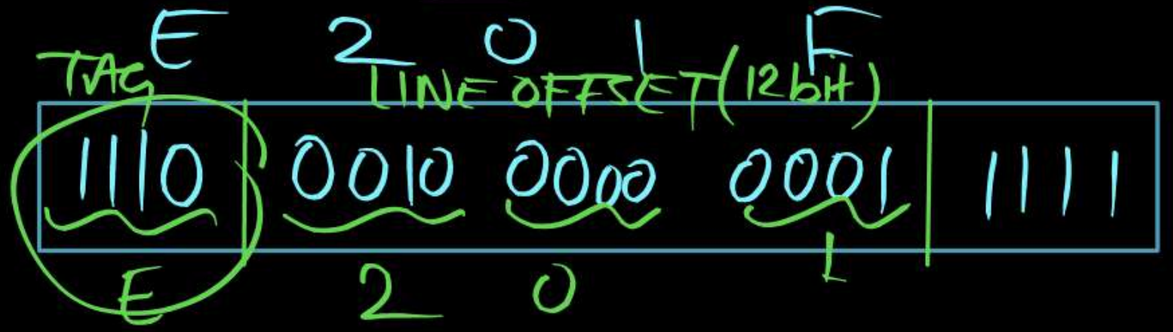
(C) E, E20

TAG: 4bit

(D) 2, 01F

PA = 20bit
W.O = 4bits
L.O = 12bit





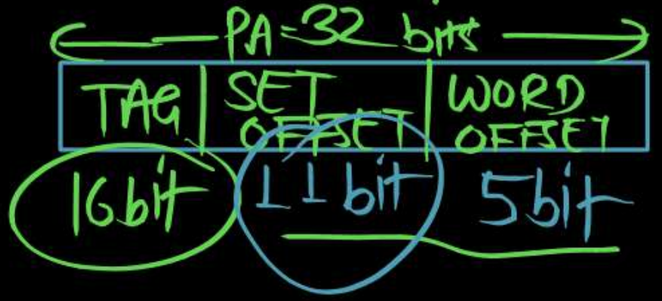
Q.5 4 marks

[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 1 replacement bit.

2^5

The number of bits in the tag field of am address is
(a) 11 (b) 14 (c) 16 (d) 27

[GATE - 2012: 2 Marks]



$\#LINE = \frac{CMSize}{BlockSize} = \frac{256KB}{32B} = \frac{2^{18}}{2^5} = 2^{13}$ LINES

$\#SET = \frac{2^{13}}{2^2} = 2^{11}$ SET

S.O = 11 bit

Q.6 [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 1 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

Tag entry Size = Tag bit + extra bit = 16 + 2 + 1 + 1 = 20 bit

Tag memory Size = #LINES X Tag Entry Size

$\Rightarrow 2^{13} \times 20$ bits

$\Rightarrow 8 \times 20 \times 2^{10}$ bit

$= 160$ k bits

4 way set Assoc.

Tag Memory Size = #SET × LINES per SET × Tag entry

$= 2^{11} \times 4 \times 20$ bit

$\Rightarrow 2 \times 4 \times 20 \times 2^{10}$ bit

$=$ 160 k bits Ans

MCQ Q.7

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 6144bits

C 6656 bits

D 5376 bits

Physical address = 32 bit

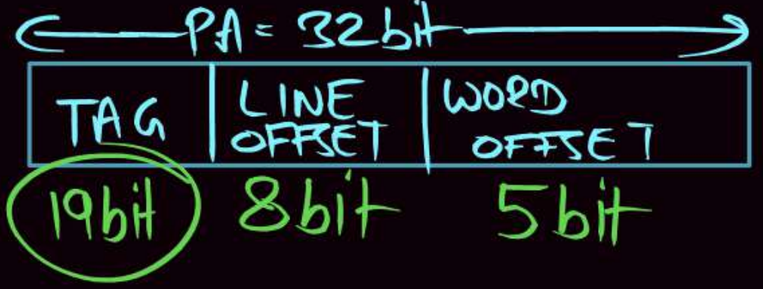
CM Size = 8KB = 2^{13} Byte

Block Size = 32B = 2^{5} Byte

Direct Mapped Cache

#LINES = $\frac{2^{13}}{2^{5}}$ = 2^{8} Lines

L.O = 8bit



Tag entry Size = Tag bit + 1 + 1 (Valid, Modified)

= 19 + 1 + 1

Tag entry = 21 bit

Tag Memory Array Size = #LINES × Tag entry Size

= 2^{8} × 21

= 256 × 21

= 5376 bits Ans

Q.8 Q.9

Common Data for next two questions:

Consider a computer with a 4-ways 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.

MCQ **Q.8**

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

[GATE-2008-CS: 2M]

A 000011000

B 110001111

C 00011000

D 110010101

0 C 7 9 5

MCQ **Q.9**

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

[GATE-2008-CS: 2M]

A 7, 6, 7

B 8, 5, 7

C 8, 6, 6

D 9, 4, 7

Q.10 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]

MCQ **Q.11**

A certain processor uses a fully associative cache of size 16 kB. The cache block size is 16 bytes. Assume that the main memory is byte addressable and uses a 32-bit address. How many bits are required for the Tag and the Index fields respectively in the addresses generated by the processor?

[GATE-2019-CS: 1M]

A 24-bits and 0-bits

B 28-bits and 4-bits

C 24-bits and 4-bits

D 28-bits and 0-bits

NAT Q.12

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 **Q.13**

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]

NAT **Q.14**

A cache memory unit with capacity of N words and block size of B Words is to be designed. If it is designed as a direct mapped cache, the length of the TAG field is 10 bits. If the cache unit is now designed as a 16-way set-associative cache, the length of the TAG field is _____ bits.

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

Q.15 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

MCQ **Q.16**

The size of the physical address space of a processor is 2^p bytes. The word length is 2^wbytes. 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

[GATE-2018-CS: 2M]

A $P - N - \log_2 K$

B $P - N + \log_2 K$

C $P - N - M - W - \log_2 K$

D $P - N - M - W + \log_2 K$

MCQ **Q.17**

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.

$A1=0\times42C8A4, A2=0\times546888, A3=0\times6A289C, A4=0\times5E4880$

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 **Q.18**

Consider a set-associative cache of size 2 kb (1 KB = 2^{10} bytes) with cache block size of 64 bytes. Assume that the cache is byte - addressable and a 32-bit address is used for accessing the cache. If the width of the tag field is 22 bits, the associativity of the ache is _____.

[GATE-2021(set-2)-CS: 1M]

MCQ **Q.19**

Consider a 4-way set associative cache (initially empty) with total 16 cache blocks. The main memory consists of 256 blocks and the request for memory blocks is in the following order:

K MOD 16

0, 255, 1, 4, 3, 8, 133, 159, 216, 129, 63, 8, 48, 32, 73, 92, 155

Which one of the following memory block will NOT be in cache if LRU replacement policy is used?

[GATE-2009-CS: 2M]

A 3

B 8

C 129

D 216

NAT Q.20

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 _____.

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

— : Compulsory

— : Conflict

MCQ **Q.21**

If the associativity of a processor cache is doubled while keeping the capacity and block size unchanged, which one of the following is guaranteed to be NOT affected?

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

A. Width of tag comparator

B. Width of set index decoder

C. Width of way selection multiplexer

D. Width of processor to main memory data bus

NAT **Q.22**

Consider a two-level cache hierarchy with L_1 and L_2 caches. An application incurs 1.4 memory accesses per instruction on average. For this application, the miss rate of L_1 cache is 0.1; the L_2 cache experiences on average, 7 misses per 1000 instructions. The miss rate of L_2 expressed correct to two decimal places is _____.

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

MCQ **Q.23**

In a two-level cache system, the access times of L_1 and L_2 caches are 1 and 8 clock cycles, respectively. The miss penalty from the L_2 cache to main memory is 18 clock cycles. The miss rate of L_1 cache is twice that of L_2. The average memory access time (AMAT) of this cache system is 2 cycles. The miss rates of L_1 and L_2 respectively are:

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

A 0.111 and 0.056

B 0.056 and 0.111

C 0.0892 and 0.1784

D 0.1784 and 0.0892

Q.24

Common Data for next two questions:

Consider a machine a 2-way set associative data cache of size 64Kbytes and block size 16 bytes. The cache is managed using 32 bit virtual addresses and the page size is 4 Kbytes. A program to be run on this machine begins as follows:

```
Double ARR [1024] [1024]
Int i, j;
/ * Initalize array ARR to 0.0 */
for (i = 0; i < 1024; i ++)
for (j = 0; j < 1024; j++)
ARR [i] [j] = 0.0;
```

The size of double 8 bytes. Array ARR is in memory starting at the beginning of virtual page 0×FF000 and stored in row major order. The cache is initially empty and no pre-fetching is done. The only data memory references made by the program are those to array ARR.

MCQ

The total size of the tags in the cache directory is

[GATE-2008-CS: 2M]

A 32 kbits

B 34 kbits

C 64 kbits

D 68 kbits

Q.25

[Common Data for this and next question]

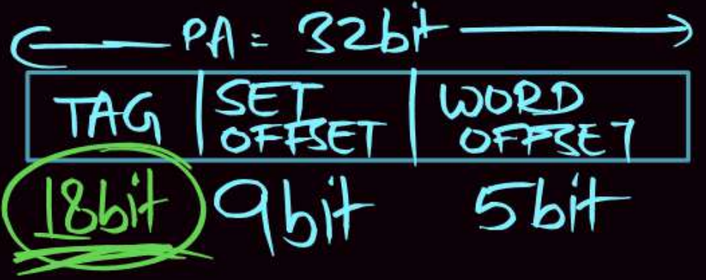
Consider two cache organization. The first one is 32 KB 2-way set associative with 32-byte block size. The second one is of the same size but direct mapped. The size of an address is 32 bits in both cases. A 2-to-1 multiplexer has latency of 0.6 ns while a k-bit comparator has a latency of k/10 ns. The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

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

[GATE - 2006: 2 Marks]

2 Way Set Associative

Cache = 32kB, 2 way set Ass.
Block size = 32B, P.A = 32 bit

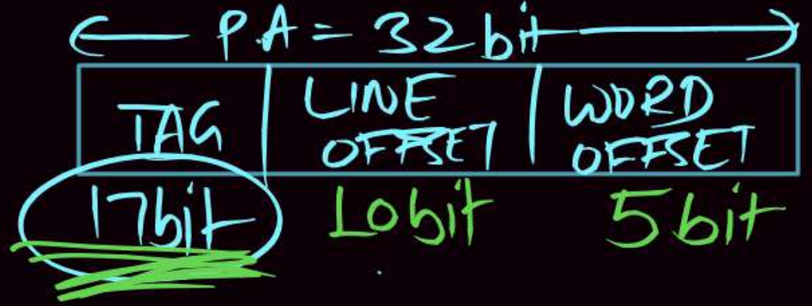


$$\#SET = \frac{\#LINES}{N\text{-}Ways} \Rightarrow \frac{2^{10}}{2^1} = 2^9 \text{ Set}$$

Direct Mapping

Cache Size = 32kB, Direct Mapping
Block Size = 32B (2^5) P.A = 32 bit

Direct



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

$$\Rightarrow \frac{32kB}{32B} = \frac{2^{15}}{2^5}$$

L.O = 10bit $\log_2 10$

2 way Set Associative

Hit Latency = Latency of Tag Comparator + Latency of Multiplexer

$= \frac{k}{10} + 0.6$

$= \frac{18}{10} + 0.6$

$= 1.8 + 0.6$

$h_1 = 2.4$ ns Ans

Direct Mapping

Hit latency = Latency of Tag Comparator.

$= \frac{k}{10}$ (k: # of Tag bits)

$= \frac{17}{10}$

$h_2 = 1.7$ nsec Ans

Q.26 (4 marks)

[Common Data from previous question]

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.27 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×18-bit, 1K×19-bit, 4K×16-bit (b) 1K×16-bit, 1K×19-bit, 4K×18-bit

(c) 1K×16-bit, 512×18-bit, 1K×16-bit (d) 1K×16-bit, 512×18-bit, 1K×18-bit

[GATE - 2006: 2 Marks]

Q.28 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]

Q.29 Consider a system with 2 KB direct mapped data cache with a block size of 64 bytes. The system has a physical address space of 64 KB and a word length of 16 bits. During the execution of a program, four data words P, Q, R and S are accessed in that order 10 times (i.e., PQRSPQRS.....) Hence, there are 40 accesses to data cache altogether. Assume that the data cache is initially empty and no other data words are accessed by the program, The addresses of the first bytes of P, Q, R and S are 0×A248, 0×C28A, 0×CA8A and 0×A262, respectively. For the execution of the above program, which of the following statements is/are TRUE with respect to the data cache? [2022: MSQ 2M]

A. Every access to S is a hit.

B. Once P is brought to the cache it is never evicted.

C. At the end of the execution only R and S reside in the cache.

D. Every access to R evicts Q from the cache.

THANK YOU!