Webinar-2

Cache Memory and Mapping Techniques

by

VIVEKANANDA M R
vivekanandamr@wilp.bits-pilani.ac.in

BITS Pilani
Pilani Campus

Direct Mapping

BITS Pilani
Pilani Campus

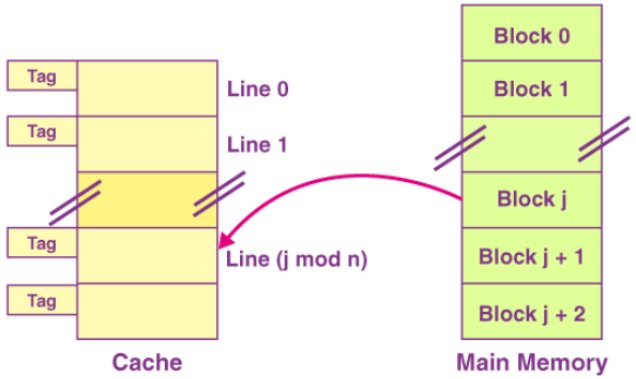
Direct Mapped Cache - Summary

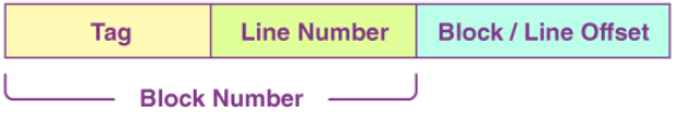
- Each block of main memory maps to only one cache line
  - **if a block is in cache, it must be in one specific place**
  - Cache line number = (Address of the Main Memory Block) Modulo (Total number of lines in Cache)
  - **i = j modulo m**

  where i = cache line number

  j = main memory block no.

  m = no.of lines in the cache
- Address is split in three parts:
  - Tag
  - Line
  - Word/Block Offset





Problem- 1: Direct mapping

Consider a cache memory of 8192KB with line size of 128B. Find the number of bits required for TAG, LINE and WORD fields of the main memory address 0xFEEDF00D?

Solution-1:

- No. of bits needed for the Main Memory Address

1	1	1	1	1	1	1	0	1	1	1	0	1	1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	0	1
F				E				E				D				F				0				0				D			

= 32 bits

- Block Size (Line Size = Block Size)
  = 128Bytes

- No. of bits to represent WORD
  128 Bytes = 2^7 => 7 bits

Problem- 1: Direct mapping (Contd..)

innovate achieve lead

Consider a cache memory of 8192KB with line size of 128B. Find the number of bits required for TAG, LINE and WORD fields of the main memory address 0xFEEDF00D?

Solution-1

- Cache Size = 8192 KB
- Total No. of Cache Lines
  = Cache size / Line size
  = 8192 KB/128B = $(2^{13}.2^{10}) / 2^{7} = 2^{16}$ = 64K lines

- Total bits to represent line field
  = 16 bits

- Total bits to represent Tag filed is:
  = 32 – Line – Word = 9 bits

Problem- 1: Direct mapping (Contd..)

Given Address : 0xFEEDF00D

F				E				E			D				F				0				0				D			
1	1	1	1	1	1	1	0	1	1	1	0	1	1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	0	1
Tag bits (9)									**Line bits (16)**															**Block offset (7)**							

Problem-2: Direct mapping

innovate achieve lead

Consider a direct mapped cache of size 16 KB with block size of 256 bytes. The size of main memory is 128 KB. Find number of bits in tag, line and word field.

Given Data:

- Main Memory Size : 128KB
- Block Size: 256 bytes
- Cache Size: 16KB

- **Total No.of bits for Main Memory address:**
- Main Memory Size = 128 KB = 2^7 x 2^{10} Bytes = 2^{17} Bytes
- Total No.of bits for Main Memory → **17 bits**

Main Memory		
Tag	Line	Block/Word

Problem-2: Direct mapping

innovate achieve lead

Consider a direct mapped cache of size 16 KB with block size of 256 bytes. The size of main memory is 128 KB. Find number of bits in tag, line and word field.

Given Data:

- Main Memory Size : 128KB
- Block Size: 256 bytes
- Cache Size: 16KB

- **Total No.of bits for Word/Block Offset:**
- Block Size = 256 byte = 2^8 Bytes
- Total No.of bits for Block/Word Field → **8 bits**

Main Memory		
Tag	Line	Block/Word

Problem-2: Direct mapping

innovate achieve lead

Consider a direct mapped cache of size 16 KB with block size of 256 bytes. The size of main memory is 128 KB. Find number of bits in tag, line and word field.

Cache of size = 16 KB = 2^{14} bytes = 14 bits
Block size = 256 bytes = 2^{8} bytes= 8 bits
Number of bits in Line number = Cache size / Block size
= 2^{14} bytes / 2^{8} bytes
= 2^{6}

Number of bits in **Cache line = 6 bits**

Main Memory		
Tag	Line	Block

Problem-2: Direct mapping

Consider a direct mapped cache of size 16 KB with block size of 256 bytes. The size of main memory is 128 KB. Find number of bits in tag, line and word field.

Number of bits in Tag =
Number of bits in main memory – number of bits in line number – number of bits in block

= 17 – Line bits – Block bits

= 17 – 6 – 8 =3

Main Memory		
Tag	Line	Block

Problem-3: Direct mapping (HW)

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} lines. Let the addresses of two consecutive bytes in main memory be $(E201F)_{16}$ and $(E2020)_{16}$. What are the tag and cache line addresses (in hex) for main memory address $(E201F)_{16}$?

Direct mapping

innovate achieve lead

Advantages of Direct-mapping

- It requires very less hardware complexity.
- It is a very simple method of implementation.
- Access time is very low.

Disadvantages of Direct-mapping

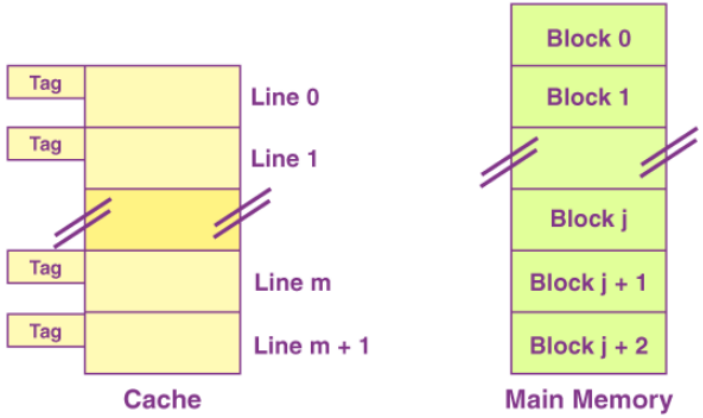
- Use of cache space is inefficient.
- Conflict misses are high.

Associative Mapping

BITS Pilani
Pilani Campus

Associative Mapping - Summary

- A main memory block can load into any line of cache
- Memory address is interpreted as tag and word
- Tag uniquely identifies block of memory
- Every line's tag is examined for a match
- Cache searching gets expensive
- The memory address has only 2 fields
- i)word
- ii)tag.



Problem 4: Fully Associative

A system has main memory of size 128Byte with on chip cache 32 Bytes and block size of 8 Bytes, with the system having fully associative cache mapping, find the following:

a) The number of main memory address bits.

Main Memory Size = 128 Bytes = 2^7 Bytes

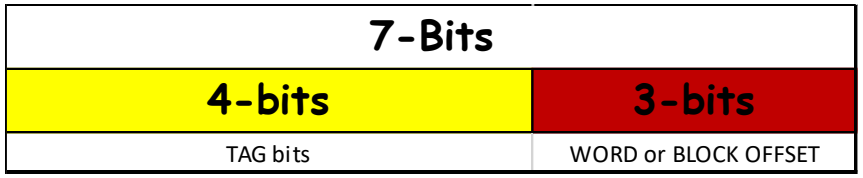
Thus, the size of Physical Address = 7 bits.

b) The number of Tag bits and Word bits.

Block size = 8 Bytes = 2^3 Bytes

# of bits for WORD offset field = 3 bits

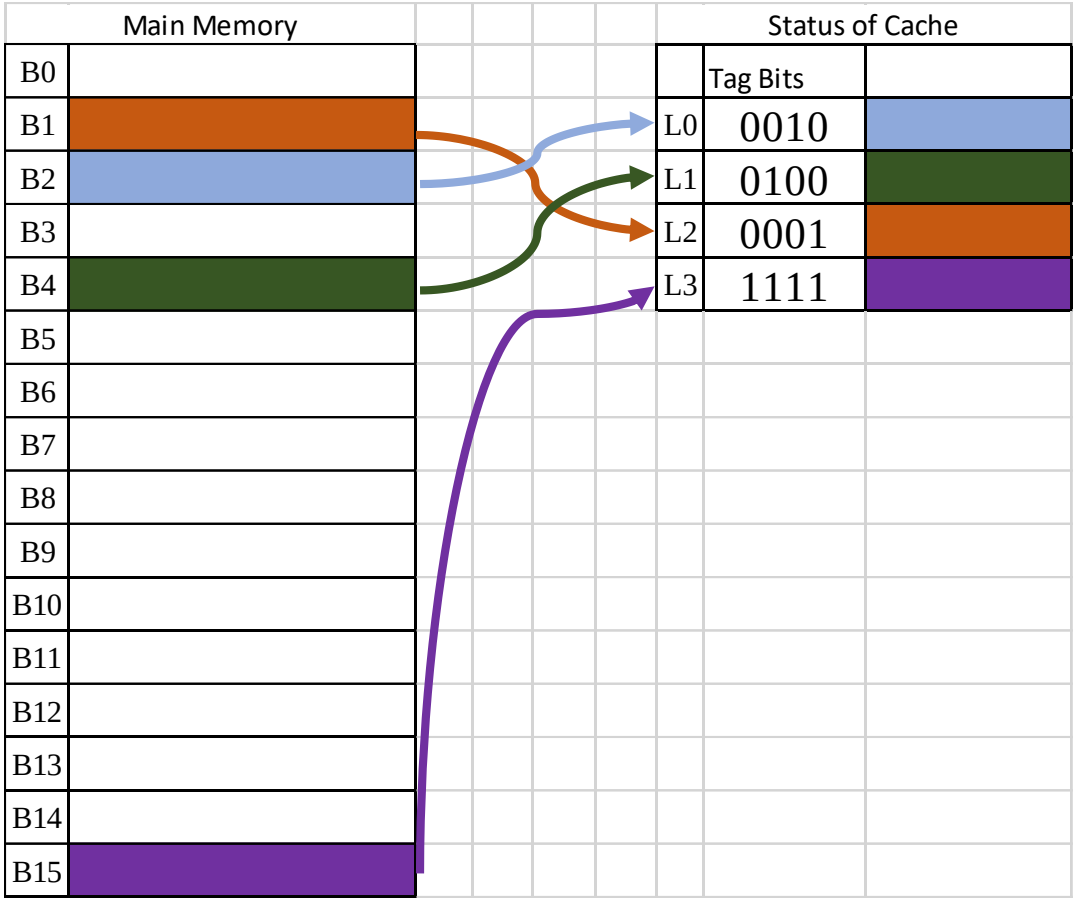
# of Tag bits = 7 – WORD offset = 7-3 = 4 bits



Problem 4: Fully Associative

innovate achieve lead

c) If the CPU request the addresses 0001100, 0011001 find whether is a cache hit or cache miss for each of the address.



0001100

0001100 - Hit

0011001

0011001-Miss

Problem 5: Fully Associative

innovate achieve lead

An 8KB associative cache has a line size of 32 Bytes. Main memory size is 1GB. Find the number of TAG bits and number of comparators required for search.

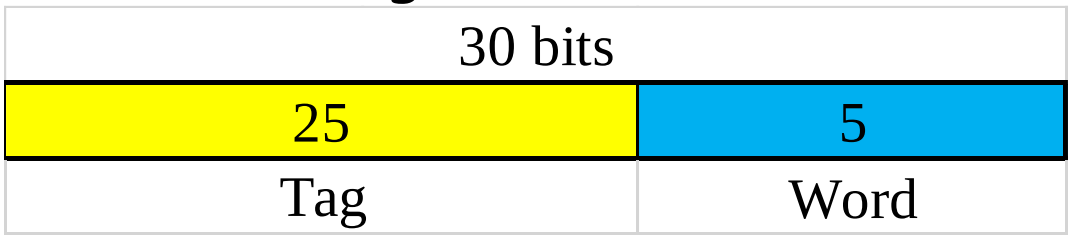
Given Data:

Cache size = 8KB = $2^3.2^{10}$ = 2^{13} Bytes

Block Size = 32 Bytes = 2^5 Bytes

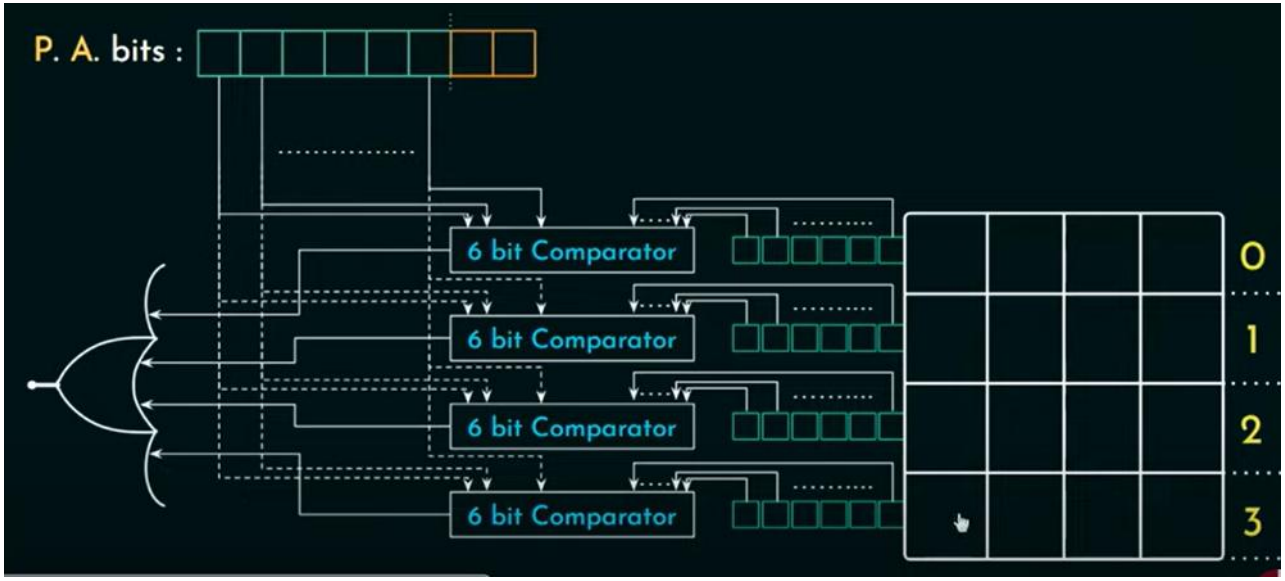
Memory size = 1GB = 2^{30} Bytes

- # of bits for main memory address = 30 bits
- # of bits for Word offset field = 5 bits
- # of bits for Tag field = 30 – 5 = 25 Bits



Problem 5: Fully Associative

- # of cache lines
  = (Cache Size)/(Line Size) = $2^{13}/2^5 = 2^8$ lines = 256 lines



- # of Comparators required = # of Cache lines = 256
- Size of comparator = size of tag bits = **25-bit** comparator

Problem 5: Fully Associative (Replacement-LRU)

- **LRU**
- When the cache memory is full, LRU picks the data that is **least recently used (not accessed for longest time)** and **removes it in order** to make space for the new data.

- Consider a fully associative cache using LRU replacement policy with 8 cache blocks (0-7) and the following sequence of memory block requests: 4 3 25 8 19 6 25 8 16 35 45 22 8 3 16 25 7. Find Hit and Miss Ratio, which cache block will have memory block 7?

Problem 5: Fully Associative (Replacement-LRU)

innovate achieve lead

Block Req	4	3	25	8	19	6	25	8	16	35	45	22	8	3	16	25	7
Time	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
L0																	
L1																	
L2																	
L3																	
L4																	
L5																	
L6																	
L7																	
H/M																	

Problem 5: Fully Associative (Replacement-LRU)

innovate achieve lead

Block Req	4	3	25	8	19	6	25	8	16	35	45	22	8	3	16	25	7

Block Req	4	3	25	8	19	6	25	8	16	35	45	22	8	3	16	25	7
Time	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
L0	4_0	4_0	4_0	4_0	4_0	4_0	4_0	4_0	4_0	4_0	45_{10}	45_{10}	45_{10}	45_{10}	45_{10}	45_{10}	45_{10}
L1		3_1	3_1	3_1	3_1	3_1	3_1	3_1	3_1	3_1	3_1	22_{11}	22_{11}	22_{11}	22_{11}	22_{11}	22_{11}
L2			25_2	25_2	25_2	25_2	25_6	25_6	25_6	25_6	25_6	25_6	25_6	25_6	25_6	25_{15}	25_{15}
L3				8_3	8_4	8_4	8_4	8_7	8_7	8_7	8_7	8_7	8_8	8_8	8_8	8_8	8_8
L4					19_4	19_4	19_4	19_4	19_4	19_4	19_4	19_4	19_4	3_{13}	3_{13}	3_{13}	3_{13}
L5						6_5	6_5	6_5	6_5	6_5	6_5	6_5	6_5	6_5	6_5	6_5	7_{16}
L6									16_8	16_8	16_8	16_8	16_8	16_8	16_{14}	16_{14}	16_{14}
L7										35_9	35_9	35_9	35_9	35_9	35_9	35_9	35_9
	M	M	M	M	M	M	H	H	M	M	M	M	H	M	H	H	M

Set Associative Mapping

BITS Pilani
Pilani Campus

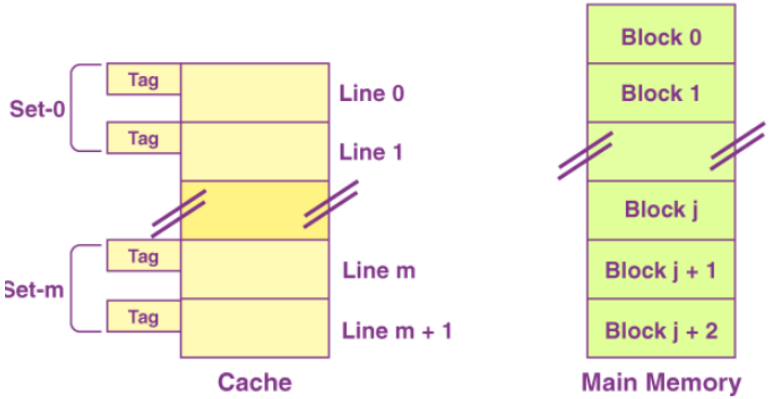
Set Associative - Summary

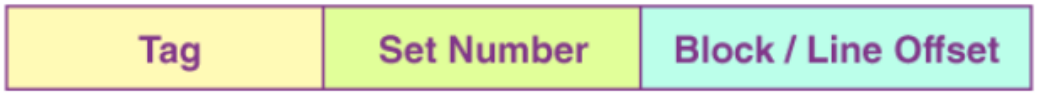
- Makes use of advantages of Direct Mapped and Associative mapping
  - Block is mapped to a set (Direct mapping)
  - Within the block can be placed in any line (Associative)
- K-Way associative cache ➔ K lines in a set
- Mapping function : i= j % v

Where i = cache set number

j = main memory block number

v = number of sets in the cache





Problem 6: Set Associative

innovate achieve lead

A set associative cache memory consists of 128 lines divided into four line sets. The main memory consists of 16,384 blocks and each block contains 256 eight bit words.

a) How many bits are required for addressing the main memory?

Size of main memory = 16384 blocks

= 16384 x 256 x 8 bits

= 16KB x 256B

= $2^{14} \times 2^{8}$

= 2^{22} bytes

Problem 6: Set Associative

innovate achieve lead

A set associative cache memory consists of 128 lines divided into four line sets. The main memory consists of 16,384 blocks and each block contains 256 eight bit words.

b) How many bits are needed to represent the TAG, SET and WORD fields?

Block size = 256 bytes

= 2^8 bytes

# of bits in WORD offset = 8 bits

Number of sets in cache = Number of lines in cache / Set size

= 128 blocks / 4 blocks = 32 sets

= 2^5 sets

# of bits in set number = 5 bits

Problem 6: Set Associative

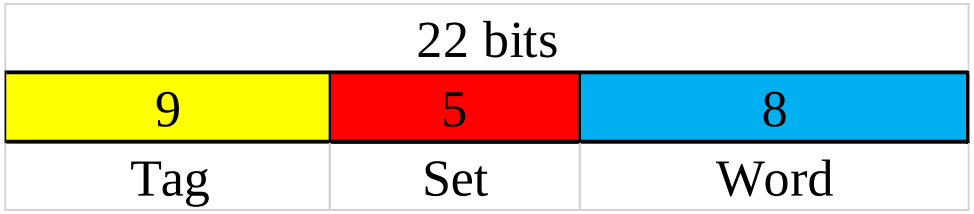
innovate achieve lead

Number of bits in TAG = Number of bits in physical address – (Number of bits in set number + Number of bits in word)

= 22 bits – (5 bits + 8 bits)

= 9 bits

Thus, Number of bits in TAG = 9 bits



Problem 6: Set Associative

innovate achieve lead

Consider 4-way set associative cache and Main Memory size of 4 MB Block Size 64B and Tag bits 10. Find Physical Address split and the size of Cache memory in terms of Bytes.

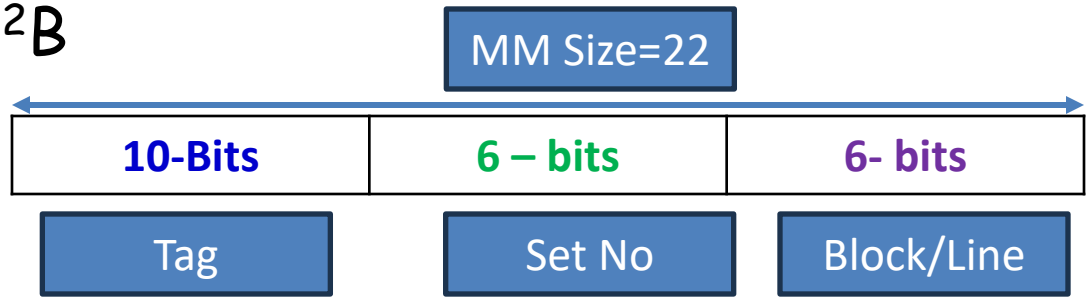
Solution:

Main Memory= 4MB = 2^{22}B

PA Size = 22 bits

Block Size = 64B = 2^{6}B

Block Offset = 6 bits



No. of blocks in MM = (MM Size/Block Size)

= 2^{22}B / 2^{6}B = 2^{16} Blocks

Tag Bits = 10

Problem 7: Set Associative

innovate achieve lead

Consider 4-way set associative cache and Main Memory size of 4 MB Block Size 64B and Tag bits 10. Find Physical Address split and the size of Cache memory in terms of Bytes.

Solution:

1-Set has 4 Lines

Set Size = 4 = 2^2

Set No. bits = 6 ➔ No. of Sets = 2^6

No. of Lines in Cache = (No. of sets x Set Size)

= 2^6 x 2^2 = 2^8

Cache Size = No. of Lines x Block size

= 2^8 x 2^6 = 2^{14}= **16KB**

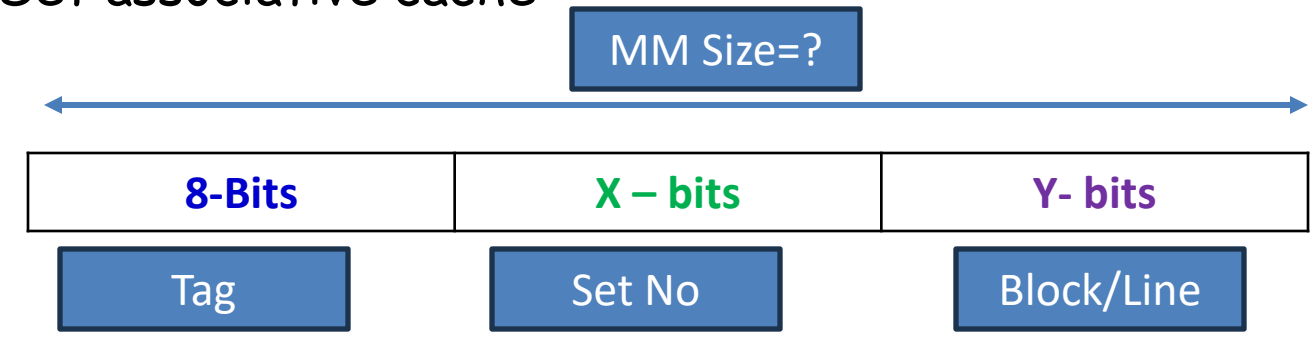
Problem 7: Set Associative

Consider 8-way set associative cache with cache size 256KB of memory and the number of TAG bits are 8. Find the size of min memory in terms of Bytes.

Solution:

Given:

- Cache Size: 256KB
- Tag Bits:8
- 8-way Set associative cache



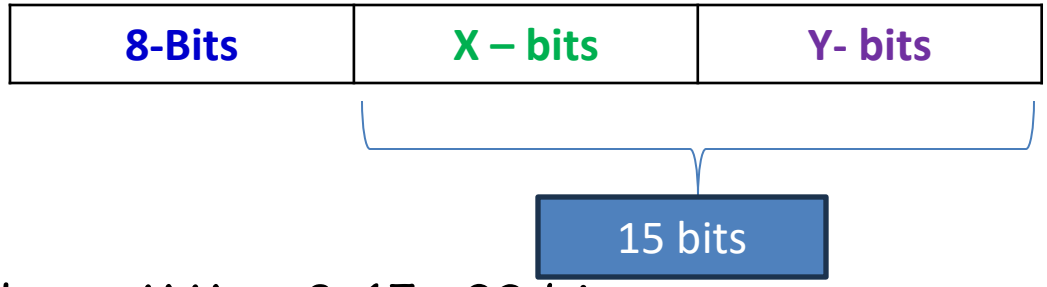
Problem 7: Set Associative

Consider 8-way set associative cache with cache size 256KB of memory and the number of TAG bits are 8. Find the size of main memory in terms of Bytes.

Solution:

Given:

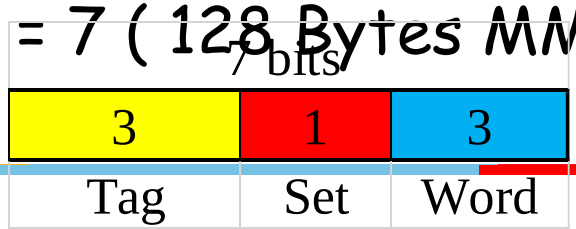
- 8-way Set associative cache ➔ 8 lines per set
- Cache Size = (No. of Sets x Set Size) x Block Size
- $256K = (2^X \times 8) \times 2^Y$
- $2^8 \times 2^{10} = (2^X \times 2^3) \times 2^Y$
- $2^{18}/2^3 = 2^{(X+Y)}$
- $2^{15} = 2^{(X+Y)}$
- No of bits required to address MM = 8+15= 23 bits
- MM Size = 2^{23} B



Problem 8: Set Associative

A system has a main memory of 128 Bytes and a cache size of 32 Bytes with 8 Bytes per cache block. Assume that the size of each memory word is 1 byte. if the cache is organized as a 2-way set-associative cache. Find out TAG, SET, and WORD field bits.

- Block size = 8 bytes = 2^3 Bytes ➔ **WORD = 3bits**
- Cache size = 32 Bytes = 2^5 Bytes
- Number of lines = Cache size / Block size
  $= 2^5/2^3$ Bytes = 2^2 ➔ 4
- Number of cache lines per set = 2 (2-way set associative)
- Number of Sets = Number of lines/lines per Set = 4 / 2 = 2
- # bits for identifying a **SET = 1**
- Total number of address bits = 7 (128 Bytes MM = 2^7)
- **TAG = 7-(3+1)=3bits**



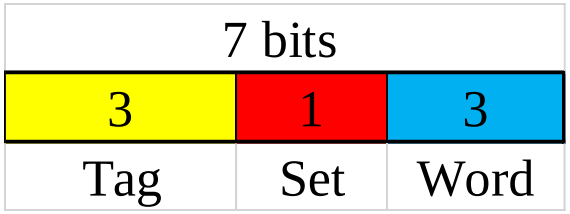
Problem 8: Set Associative

innovate achieve lead

b) When a program is executed, the processor requests data from the following word addresses:

0001010 , 0010010

For each of the above addresses, find out for which set it will map.



- For the word address: 0001010

  000 1 010 → Set-1

- For the word address: 0010010

  000 0 010 → Set-0

Problem 9: Set Associative (LRU)

Consider a 2 – way set associative cache with 4 cache lines (0-3). The memory block requests are in the order-

4, 3, 25, 8, 4, 6, 25, 8, 16, 35, 4

If LRU replacement policy is used, calculate the hit ratio and miss ratio

Problem 9: Set Associative (LRU)

		4	3	25	8	4	6	25	8	16	35	4
	Time	0	1	2	3	4	5	6	7	8	9	10
Set 0	L0											
	L1											
Set 1	L2											
	L3											
	H/M											

Problem 9: Set Associative (LRU)

BN (j)	4	3	25	8	4	6	25	8	16	35	4
#of Sets(v	2	2	2	2	2	2	2	2	2	2	2
i=j%v	0	1	1	0	0	0	1	0	0	1	0

		4	3	25	8	4	6	25	8	16	35	4
	Time											
Set 0	L0											
	L1											
Set 1	L2											
	L3											
H/M												

Problem 10: Set Associative (FIFO)

Consider a 2 – way set associative cache with 4 cache lines (0-3). The memory block requests are in the order-

4, 3, 25, 8, 4, 6, 25, 8, 16, 35, 4

If FIFO replacement policy is used, calculate the hit ratio and miss ratio

Problem 10: Set Associative (FIFO)

		4	3	25	8	4	6	25	8	16	35	4
	Time	0	1	2	3	4	5	6	7	8	9	10
Set 0	L0											
	L1											
Set 1	L2											
	L3											
	H/M											

Problem 10: Set Associative (FIFO)

BN (j)	4	3	25	8	4	6	25	8	16	35	4
#of Sets(v)	2	2	2	2	2	2	2	2	2	2	2
i=j%v	0	1	1	0	0	0	1	0	0	1	0

		4	3	25	8	4	6	25	8	16	35	4
	Time											
Set 0	L0											
	L1											
Set 1	L2											
	L3											
	H/M											

innovate achieve lead

Thank You