CS & IT ENGINEERING

COMPUTER ORGANIZATION AND ARCHITECTURE

Cache Organization

Lecture No.- 09

By- Vishvadeep Gothi sir

Recap of Previous Lecture

Topic Block Replacement

Topic Cache Miss Penalty

Topic Types of Cache Miss

Topics to be Covered

Topic Mapping Hardware

Topic Array Access With Cache

Topic Multilevel Cache

[NAT]

#Q. Cache Size = 128KB

Block size = 32 bytes

Main memory address = 29-bits

4-way set associative cache

1. Tag size? 14 bits
2. Tag Directory size? $2^{12} * 14$ bits
3. Comparator required? 4 comparators ⇒ 14-bit comparator
4. MUX required? no. $= 4 * 14 = 56$

   size $= 2^{10} : 1 = 1024 : 1$

$\longleftarrow 29 \longrightarrow$

Tag	set	byte
14	10	5

$\longleftrightarrow$

$\log 128k - \log 4$

$= 17 - 2$

$= 15$

no. of blocks in cm $= \dfrac{127\text{ KB}}{32\text{ B}}$

$= 2^{12}$

[MCQ]

#Q. Consider two cache organizations. First one is 32 KB 2-way set associative with 32-bytes block size, the second is of 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 latency of $\frac{k}{10}$ ns. The hit latency of the set associative organization is h_1 while that of direct mapped 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

2-way

32

Tag	set	byte
18	9	5

log 32k - log 2

15 - 1

14

$h_1 = 0 + \frac{18}{10} + 0.6 = 2.4\text{ns}$

$2^9:1$ MUX delay not given

Direct

32

Tag	block	byte
17	10	5

log 32k = 15 bits

$h_2 = 0 + \frac{17}{10} = 1.7\text{ns}$

$2^{10}:1$ MUX delay not given

[MCQ]

#Q. Consider two cache organizations. First one is 32 KB 2-way set associative with 32-bytes block size, the second is of 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 latency of $\frac{k}{10}$ns. The hit latency of the set associative organization is h_1 while that of direct mapped 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]

Topic : Questions on Cache and Array

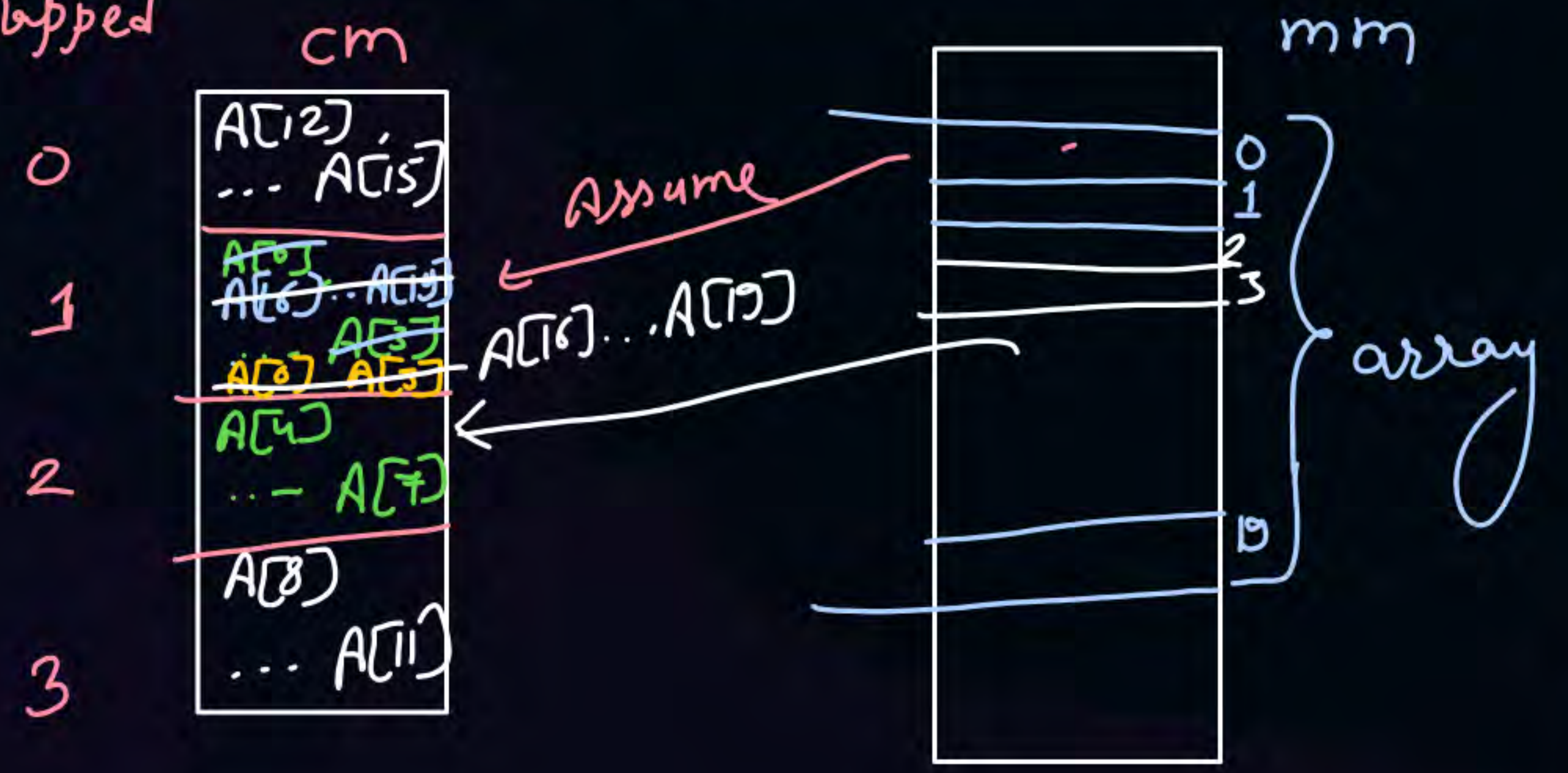
- Cache Size = 16 bytes
- Block size = 4 bytes

} no. of blocks in cache = $\frac{16B}{4B} = 4$

- Array in main memory A[20], each element is 1 byte

⇒ array size = $20 * 1B$
$= 20B$

- Direct Mapped



no. of array elements in 1 block = $\frac{4B}{1B}$
$= 4$

blocks in array = $\frac{20B}{4B}$
$= 5$ blocks

CPU access A[0] ⇒ miss

Block of 4 B (4 elements)
A[0], A[1], A[2], A[3]
ane copied to cache

—— || —— A[1] ⇒ Hit

—— || —— A[2] ⇒ Hit

—— || —— A[3] ⇒ Hit

—— || —— A[4] ⇒ Miss

A[5] ⇒ Hit

A[6] ⇒ Hit

A[7] ⇒ Hit

Block of 4 elements A[4] ... A[7] copied to cache

A[8] ⇒ miss

A[9], A[10], A[11] ⇒ hit

A[12] ⇒ miss

A[13], A[14], A[15] ⇒ hit

A[16] ⇒ miss

A[17], A[18], A[19] ⇒ hit

First time access of array will experience no. of miss = no. of blocks needed to store array in mm

	Miss	hit
first array access	5	15
2nd array access	2	18
3rd —‖—	2	18
	⋮	⋮

2nd access of array :-

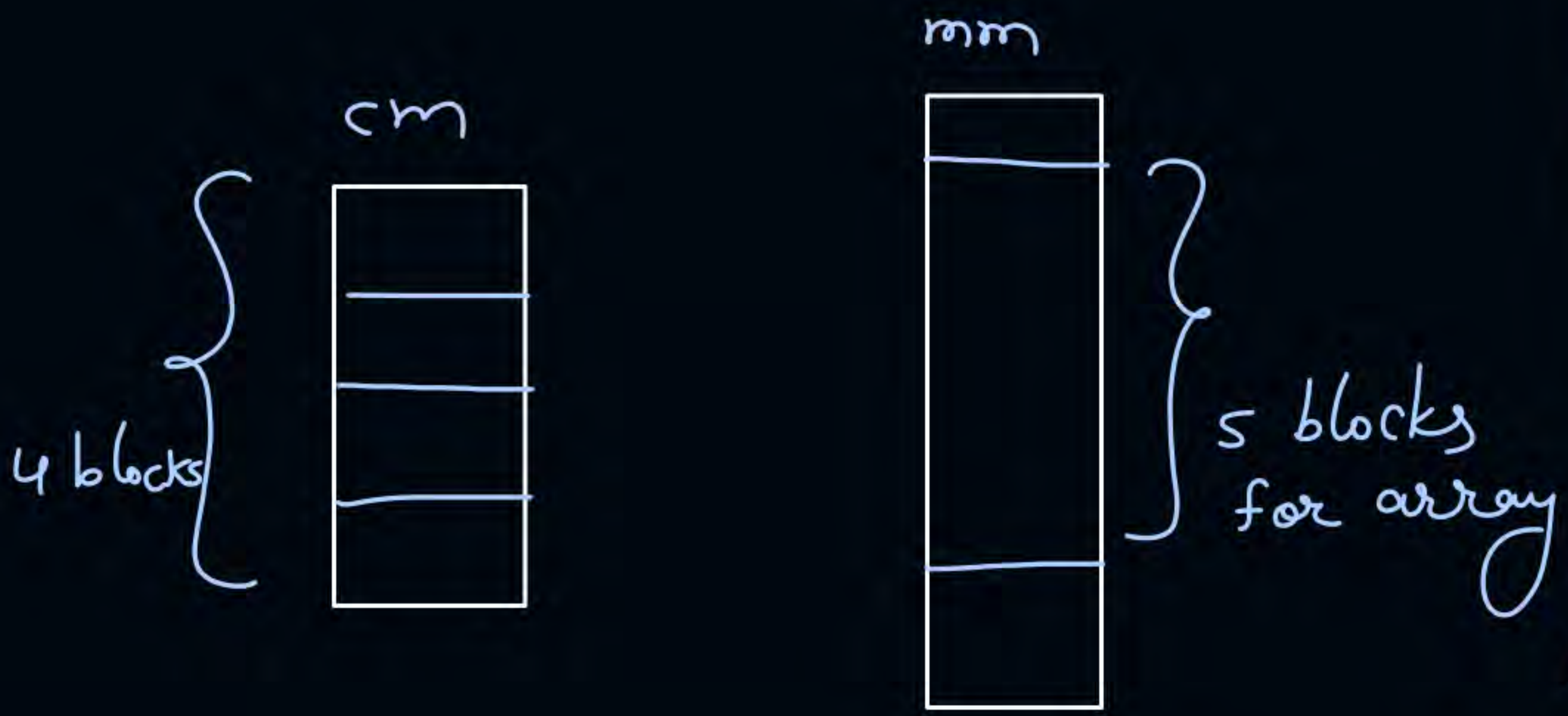
CPU accesses A[0] ⇒ Miss

A[1], A[2], A[3] ⇒ Hit

A[4] .... A[15] ⇒ Hit

A[16] ⇒ miss

A[17], A[18], A[19] = hit



no. of miss for 1st access

= no. of blocks in array

= 5

no. of overlapping blocks = 5 − 4

= 1

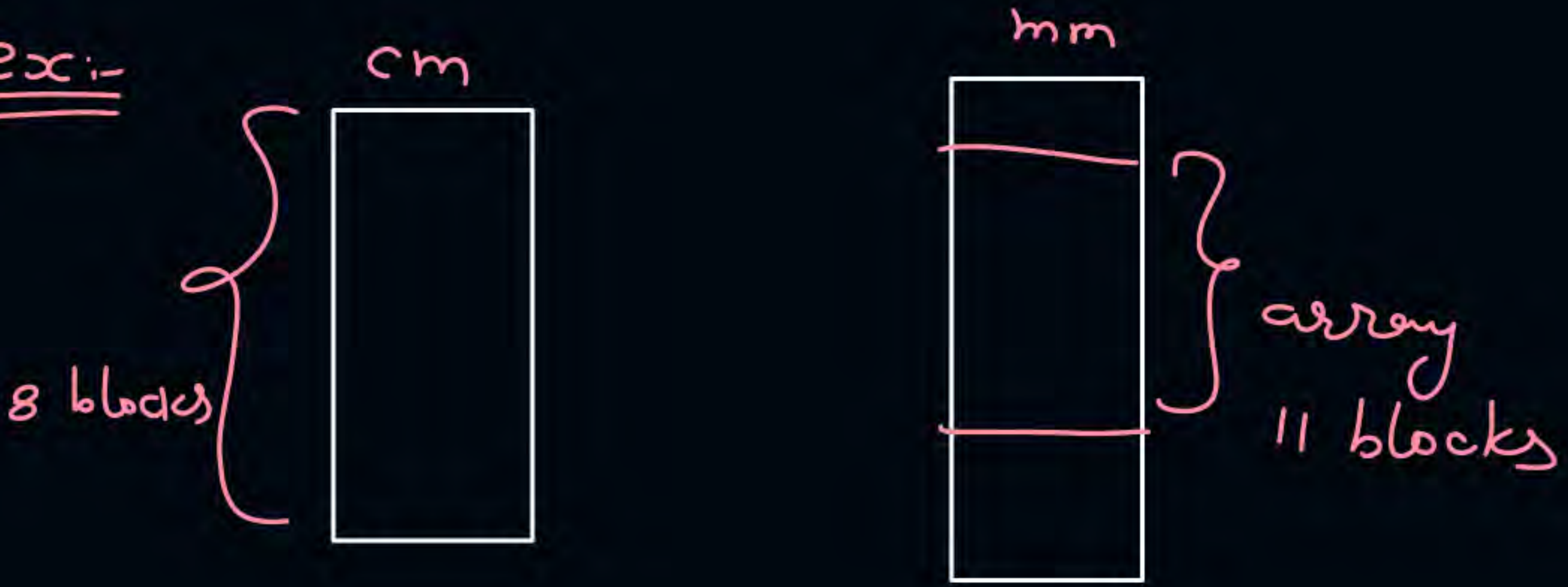
no. of miss for 2nd access

= 2 * no. of overlapping blocks

= 2 * 1

= 2

ex:-



no. of overlapping blocks

$= 11 - 8 = 3$

no. of miss for first array access = 11

——— " ——— 2nd ——— " ——— $= 2*3 = 6$

——— " ——— 3rd ——— " ——— $= 2*3 = 6$

⋮ ⋮

Topic : Questions on Cache and Array

- Cache Size = 32 bytes
- Block size = 8 bytes

} no. of blocks in cm = $\frac{32\,B}{8\,B}$ = 4 blocks

- Array in main memory int A[22], each element is 2 bytes ⇒ array size $= 22 * 2$
$= 44\,B$

- Direct mapping
- Array is accessed 4 times.
- No. of hits & misses?

no. of blocks for array

$$= \left\lceil \frac{44\,B}{8\,B} \right\rceil$$

= 6 blocks

no. of overlapping blocks $= 6 - 4$
$= 2$

no. of miss for first access = 6

— " — — 2nd — " — = $2*2 = 4$

— " — — 3rd — " — = $2*2 = 4$

— " — — 4th — " — = $2*2 = 4$

18 miss

no. of hits = $88 - 18$

= 70 hits

for 22 elements CPU generates = 22 mem. references

for 4 times access = $4*22 = 88$ — " —

hit ratio $= \frac{70}{88} = 79.5\%$

miss ratio $= \frac{18}{88} = 20.5\%$

[NAT]

blocks

mm add = 16 bits

#Q. Consider a machine with a byte addressable main memory of 2^{16} bytes. Assume that a direct mapped data cache consisting of 32 lines of 64 bytes each is used in the system. A 50 × 50 two-dimensional array of bytes is stored in the main memory starting from memory location 1100H. Assume that the data cache is initially empty. The complete array is accessed twice. Assume that the contents of the data cache do not change in between the two accesses.

How many data cache misses will occur in total?

array size $= 50 \times 50 = 2500$ elements

$= 2500$ Bytes

no. of blocks for array $= \left\lceil \frac{2500\ B}{64\ B} \right\rceil$

$= 40$ blocks

overlapping blocks = 40 − 32 = 8

no. of miss for first access = 40

—— 11 ———— 2nd —11—— = 2 * 8 = 16

Total = 56 miss

no. of mem. accesses = 2 * 2500 = 5000

hits = 5000 − 56

= 4944

[MCQ]

#Q. Which of the following $\overset{\text{blocks}}{\text{lines}}$ of the data cache will be replaced by new blocks in accessing the array for the second time?

[✓ A] line 4 to line 11

B line 4 to line 12

C line 0 to line 7

D line 0 to line 8

← 16 →

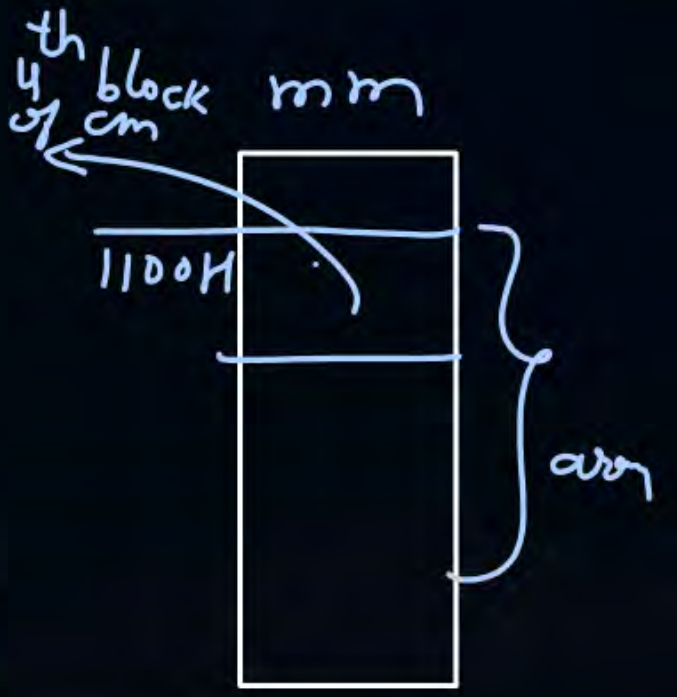
Tag	block	byte
5	5	6

$(1100)_{16}$ = 0001 0 | 001 00 | 00 0000

⇓

cm block no $= (00100)_2$

$= (4)_{10}$



[NAT]

#Q. A CPU has a 32KB direct mapped cache with 128 byte-block size. Suppose A is two-dimensional array of size 512 × 512 with elements that occupy 8-bytes each. Consider the following two C code segments, P1 and P2.

P1 :

```
for (i = 0 ; i < 512 ; i ++)
{
    for ( j = 0 j < 512 j ++)
    {
        x += A[i][j] ;
    }
}
```

Array access row wise

P2:

```
for ( i = 0 ; i < 512 i ++)
{
    for ( j = 0 j < 512 ; j ++)
    {
        x+= A[j] [i];
    }
}
```

↑ Column wise access

[NAT]

#Q. P1 and P2 are executed independently with the same initial state, namely, the array A is not in the cache and i, j, x are in registers. Let the number of cache misses experienced by P_1 be M_1 and that for P_2 be M_2.

The value of M_1 is : Ans $= 2^{14}$

no. of blocks in cache $= \frac{32\,kB}{128\,B} = 2^{8} = 256$

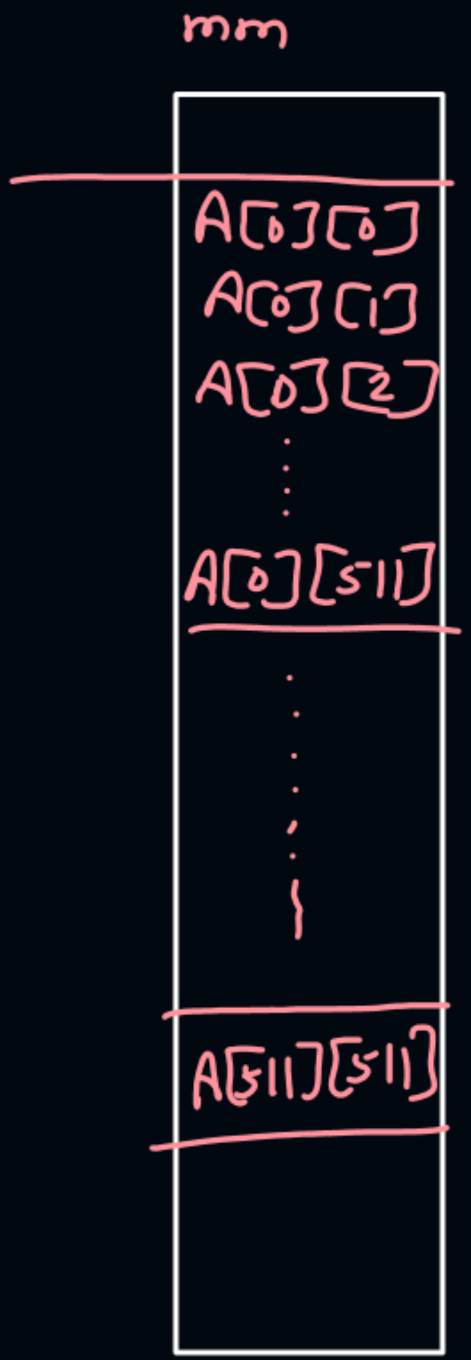
array size $= 512 \times 512 = 2^{18}$

$= 2^{18} * 8\,B$

$= 2^{21}\,B$

no. of blocks to store array $= \frac{2^{21}\,B}{128\,B}$

$= \frac{2^{21}}{2^{7}} = 2^{14}$



row-major order of array

for row wise access of array in P1,

1 miss per block of array occured

$m_1 = 2^{14}$

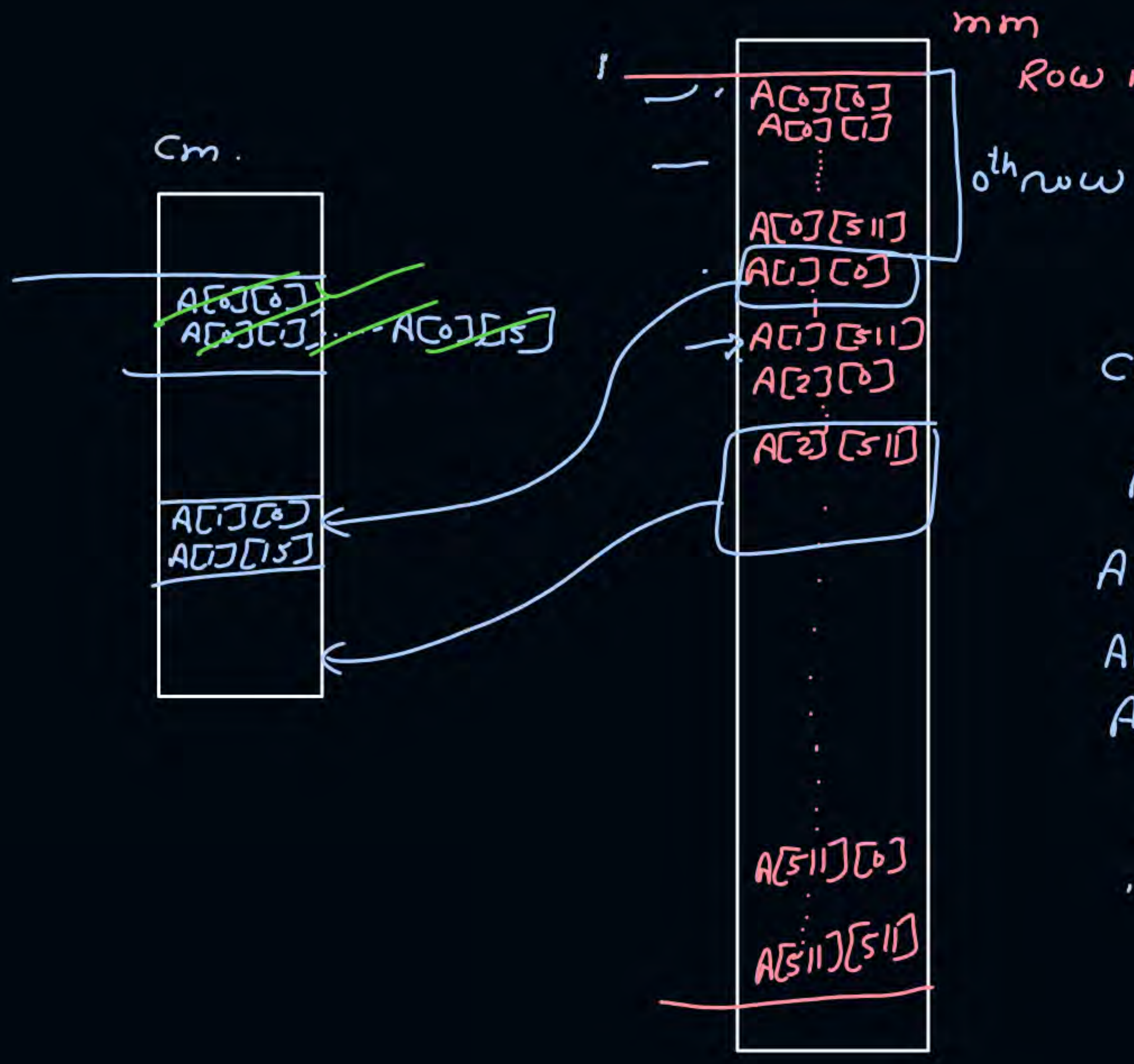
[NAT]

#Q. P1 and P2 are executed independently with the same initial state, namely, the array A is not in the cache and i, j, x are in registers. Let the number of cache misses experienced by P_1 be M_1 and that for P_2 be M_2.

The value of M_2 is : 2^{18}

no. of elements per block $= \frac{128\,B}{8\,B} = 16$

Row major order store of array



CPU accesses

A[0][0] ⇒ miss

A[1][0] ⇒ miss

A[2][0] ⇒ miss

A[3][0]

.

.

.

.

Cache can store only 256 blocks & after sometime these blocks will be replaced without any hit.

⇓

no. of miss = no. of elements $= 2^{18}$

2 mins Summary

Topic Mapping Hardware

Topic Array Access With Cache

Topic Multilevel Cache

Happy Learning

THANK - YOU