CS & IT ENGINEERING

COMPUTER ORGANIZATION AND ARCHITECTURE

Cache Organization

Lecture No.- 10

By- Vishvadeep Gothi sir

Recap of Previous Lecture

- **Topic** Hardware Implementation of Mappings
- **Topic** Hit Latency
- **Topic** Mux & Comparators in Cache
- **Topic** Array Access with Cache

Topics to be Covered

Topic Array Access with Cache

Topic Multilevel Cache

Topic : Questions on Cache and Array

- Cache Size = 16 bytes
- Block size = 4 bytes
- Array in main memory int A[10], each element is 2 bytes

Topic : Questions on Cache and Array

- Cache Size = 32 bytes
- Block size = 4 bytes

→ no. of blocks in cache = $\frac{32B}{4B}$ = 8 blocks

- Array in main memory int A[20], each element is 2 bytes

array size = 20 elements = 20 * 2 = 40 B

blocks needed to store array = $\lceil \frac{40B}{4B} \rceil$ = 10 blocks

no. of elements per block = $\frac{4B}{2B}$ = 2

no. of overlapping blocks
= 10 − 8
= 2

if array is accessed 4 times, then total no. of hits & misses are?

	1st access	2nd access	3rd	4th
miss	10	2+2 = 4	4	4
hit	10	16	16	16

↑ 1 miss per block of array

Total hits = 58

total miss = 22

[NAT] GATE-PYQ

Ans = 56

#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?

no. of blocks in cache = 32

block size = 64B

array size = 50 × 50 = 2500 element

= 2500 Bytes

no. of blocks to store array = $\lceil \frac{2500B}{64B} \rceil$ = 40

	1st access	2nd access	Total
no. of miss	40	8 * 2 = 16	40+16 = 56

no. of overlapping blocks = 40 − 32 = 8

[MCQ]

#Q. Which of the following lines of the data cache will be replaced by new blocks in accessing the array for the second time?

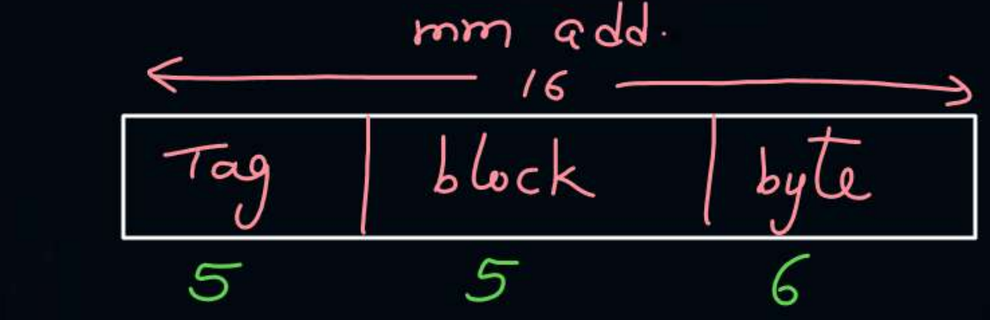
blocks

A ✓ line 4 to line 11

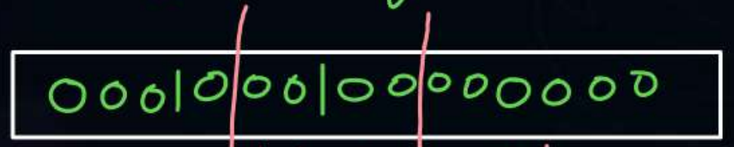
B line 4 to line 12

C line 0 to line 7

D line 0 to line 8

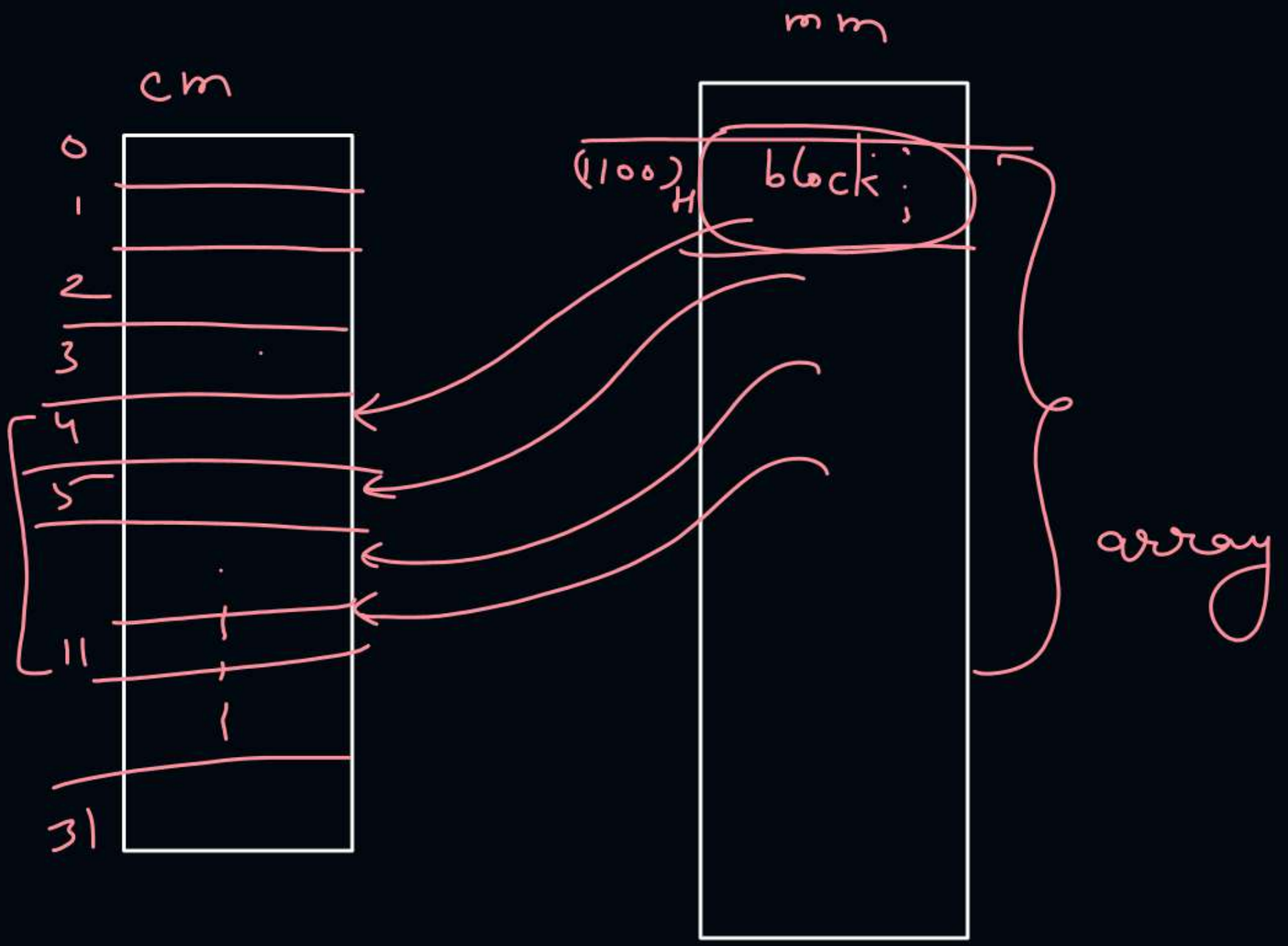


starting add. of array = $(1100)_{16}$



⇓

4 of cm


mm
cm
0
1
2
3
4
5
11
31
(100)H
block;
array

[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] ;
    }
}
```

row-wise access

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.

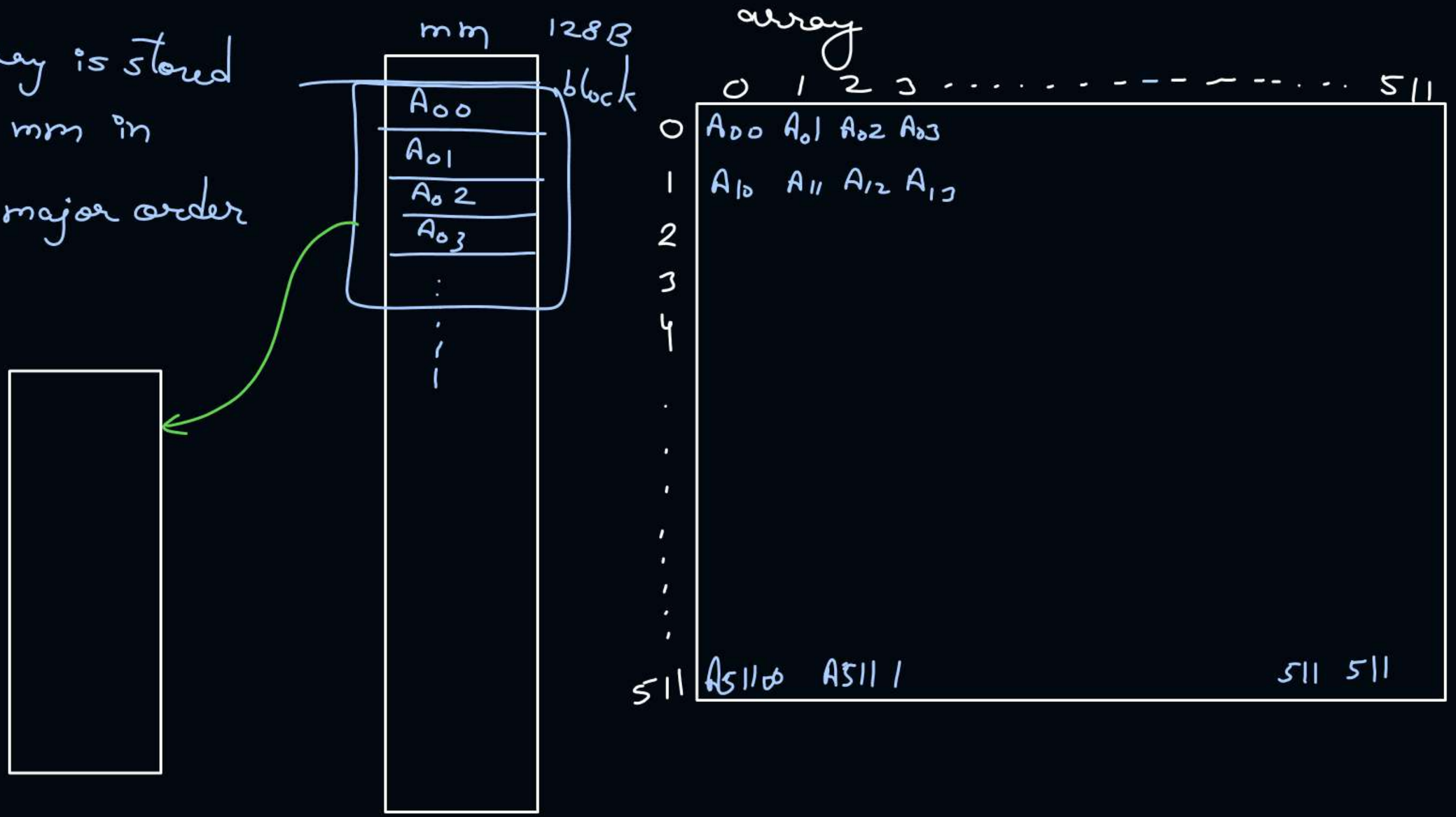
The value of M_1 is :

for P1, one miss per block → no. of miss for P1 (M1) = 2^{14}

array size = 512 × 512 = 2^{18} elements = $2^{18} * 8B = 2^{21}B$

no. of blocks needed to store array = $\frac{2^{21}B}{128B} = \frac{2^{21}}{2^7} = 2^{14}$

array is stored in mm in row major order



[NAT]

Ans = 2^{18}

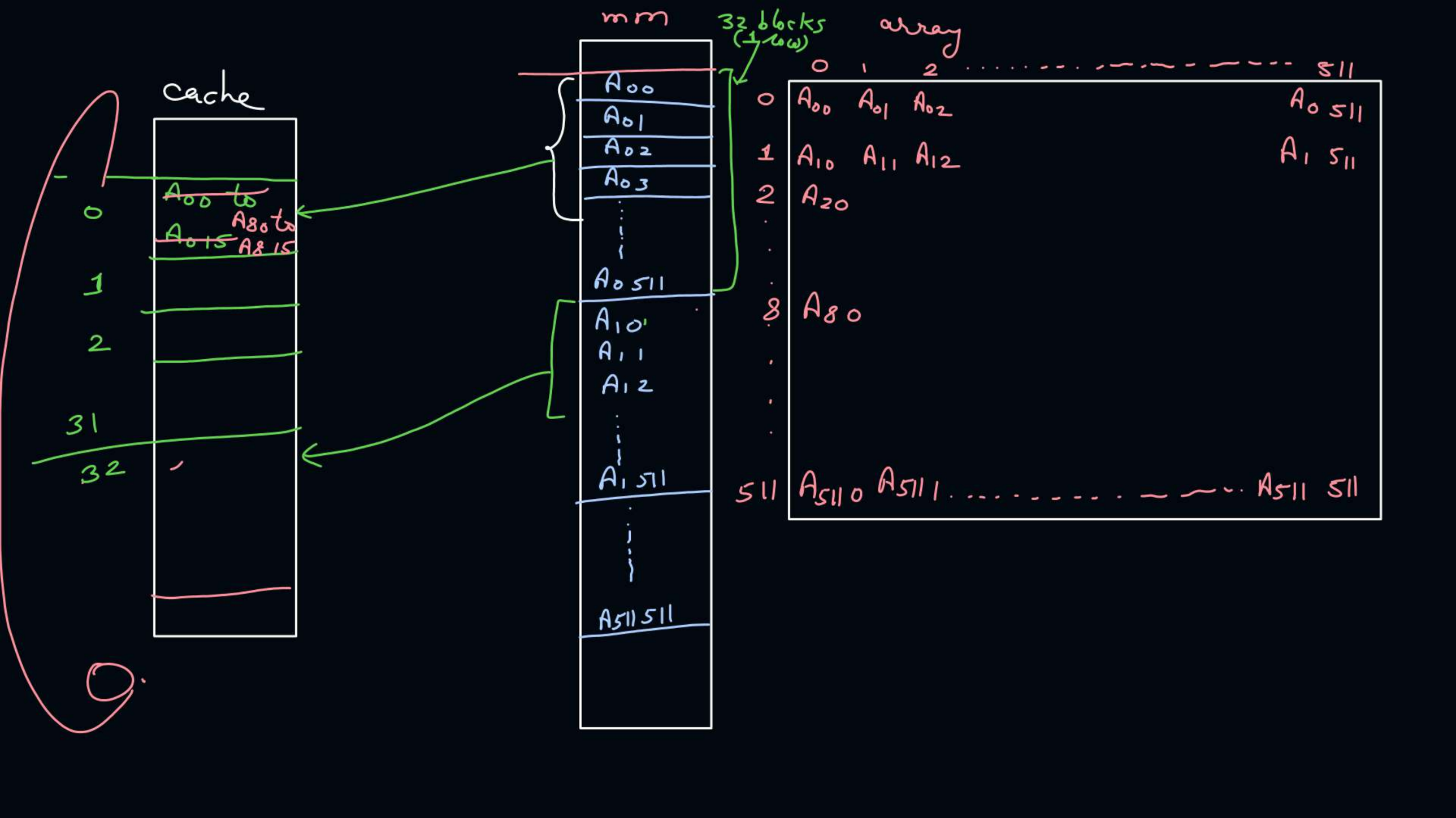
#Q. P1 and P2 are executed independently with the same initial state, namely, the array A is not in the cache and i, j, x are in registers. Let the number of cache misses experienced by P_1 be M_1 and that for P_2 be M_2.

The value of M_2 is :

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

no. of blocks needed to store one row = $\frac{512}{16} = 2^5 = 32$

no. of blocks in cm = $\frac{32\ KB}{128\ B} = 2^8 = 256$


mm
32 blocks (1 row)
array
cache
A00 to A015
A80 to A815
A00 A01 A02 ... A0511
A10 A11 A12 ... A1511
A20
A80
A5110 A5111 ... A5511
A00
A01
A02
A03
A0511
A10
A11
A12
A1511
A5110
A5111
A5511
0
1
2
31
32
8
511

CPU accesses elements		Cache	Block from mm to cm
CPU accesses elements	A[0][0]	Miss	A_{00} to $A_{0\,15} \Rightarrow x$
〃	A[1][0]	Miss	A_{10} to $A_{1\,15} \Rightarrow x+32$
〃	A[2][0]	〃	A_{20} to $A_{2\,15} \Rightarrow x+64$
〃	A[3][0]	〃	A_{30} to $A_{3\,15} \Rightarrow x+96$
⋮	⋮	⋮	⋮
〃	A[8][0]	〃	A_{80} to $A_{8\,15} \Rightarrow x+256$

replace block with elements A_{00} A_{015}

cache is very small & array is very big hence in column wise access, for new elements old blocks will be replaced without any hit.

⇓

hence 1 miss per element, cache will experience.

No. of miss for P2 (M2) = 2^{18}

Topic : Goal of Using Cache

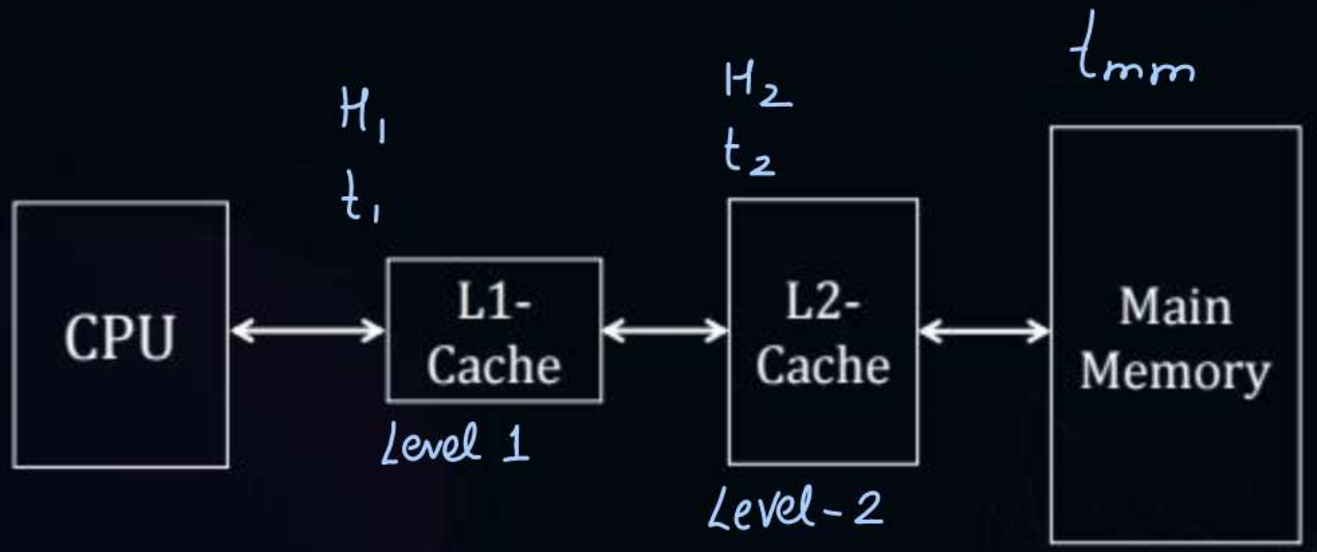
1. Minimize Access Time → Use small cache
2. Maximize Hit Rate → Use larger cache

→ use multilevel cache

3. Minimize Miss Penalty

↓

Topic : Multilevel Cache



Topic : Average Access Time Multilevel Cache

Simultaneous access :-

$$t_{avg} = H_1 t_1 + (1-H_1)\left[H_2 * t_2 + (1-H_2)\, t_{mm}\right]$$

or

$$= H_1 t_1 + (1-H_1) H_2\, t_2 + (1-H_1)(1-H_2)\, t_{mm}$$

Hierchical access:-

$$t_{avg} = H_1 * t_1 + (1 - H_1)\left[H_2 * (t_1 + t_2) + (1 - H_2)(t_1 + t_2 + t_{mm})\right]$$

or

$$= H_1 t_1 + (1 - H_1) H_2 (t_1 + t_2) + (1 - H_1)(1 - H_2)(t_1 + t_2 + t_{mm})$$

or

$$= t_1 + (1 - H_1)\left[t_2 + (1 - H_2) t_{mm}\right]$$

or

$$= t_1 + (1 - H_1) t_2 + (1 - H_1)(1 - H_2) t_{mm}$$

[NAT]

Ans = 22.75

#Q. Consider a 3-level memory hierarchy with L1 cache, L2 cache and a main memory. The hit ratios of L1 is 90% and of L2 is 95%. The access times of L1, L2 and main memory are 15ns, 60ns and 350ns respectively. The average memory access time is _____ns?

Use hierarchical access:-

$$T_{avg} = 15 + (1-0.9)\left[60 + (1-0.95)\,350\right]$$

$$= 15 + 0.1\left[60 + 0.05 * 350\right]$$

$$= \underline{\underline{22.75}}\ ns$$

2 mins Summary

Topic Array Access with Cache

Topic Multilevel Cache

Happy Learning

THANK - YOU