Recap of Previous Lecture
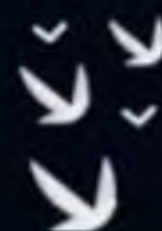
Topic — Cache Memory

Topic — Average Memory Access Time

Topics to be Covered

Topic Cache Write

Topic Write Through & Write Back Cache

Topic Write Allocate vs No Write Allocate

mm

cm

$$x = \cancel{8}\,/9$$

$$x = \cancel{8}\,/9$$

CPU

$$x = 9$$
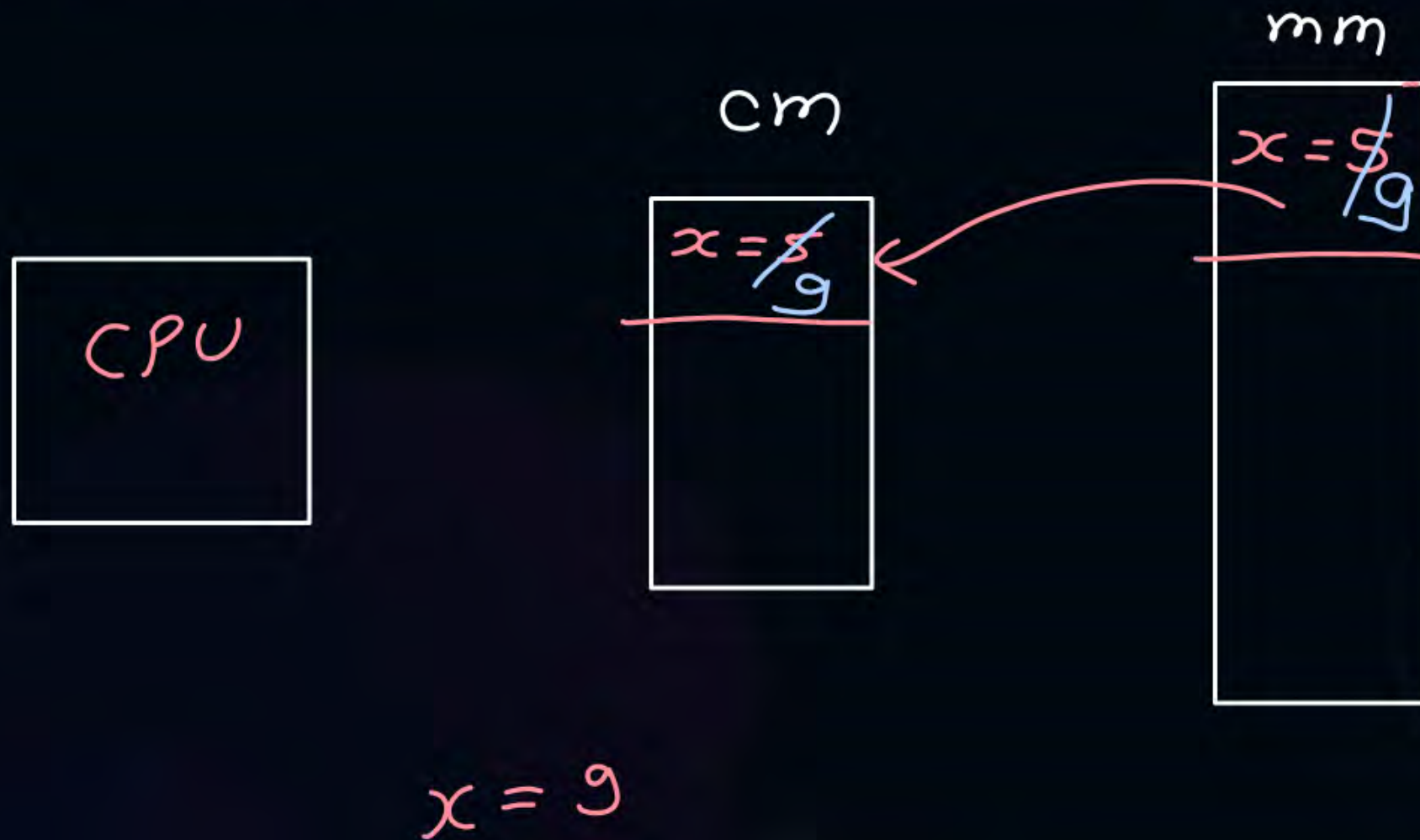
1. Write Through

2. Write Back

write in cm and mm are done simultaneously.

<u>Advantage</u> :-

→ Consistency for values in cm & mm.

<u>Disadvantage</u> :-

→ Time consuming because irrespective of hit or miss in cache, for write operation mm is accessed.

write operation performed in cm only and the cache block is written back to mm only when it is replaced.

<u>Adv:-</u>

$\longrightarrow$ Time saving as compared to write through

<u>Disadv:-</u>

$\longrightarrow$ Inconsistency for values in cm & mm.

write through:-

cm

mm

$i = 1\ 2\ 3$
$4\ 5$
$6\ 7$
$8\ 9$
10
$=$

$i = 1\ 2$
$3\ 4\ 5$
$6\ 7\ 8\ 9$
10
$=$

$for(i = 1; i < 10; i++)$

$\{$

$=$
$=$

$\}$

write back:-

cm

mm

$i = 1\ 2\ 3$
$4\ 5$
6
$7\ 8$
$9\ 10$

$i = 1\ 10$

Write Allocate: ⇒ Used with write back

The block is loaded on a write miss.

No Write Allocate: ⇒ Used with write through

The block is modified in the main memory and not loaded into the cache.

write through with no write allocate

Read

Hit in cache

CPU reads required content from cache

miss in cache

CPU reads req. content from mm and the missed block is copied from mm to cache.
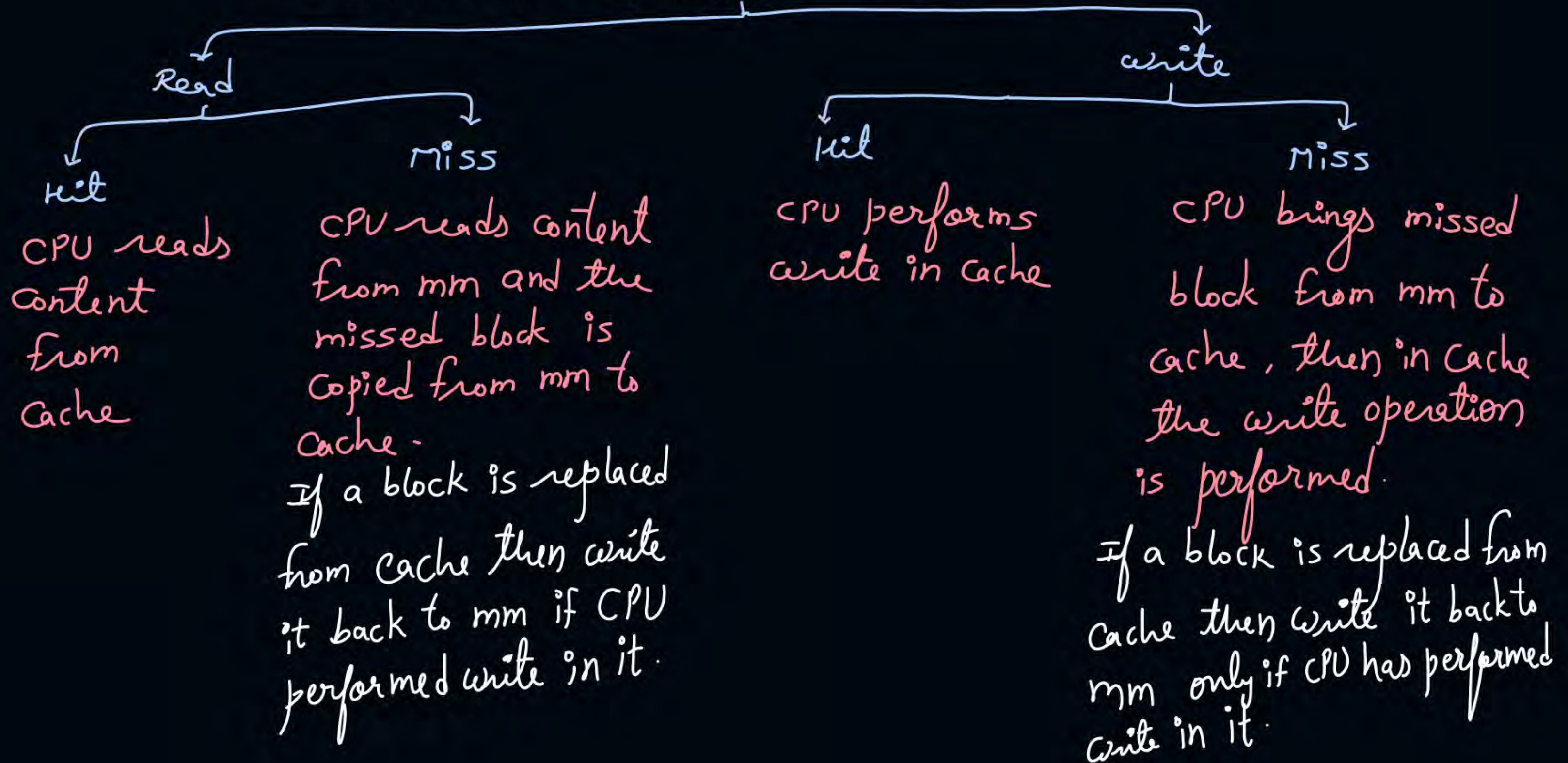If a block is replaced from cache then no need to write it back in mm.

write

Hit

CPU performs write in cache and mm simultaneously

Miss

CPU performs write in mm and do not bring missed block in cache

write back with write allocate

Read

Hit
CPU reads content from cache

Miss
CPU reads content from mm and the missed block is copied from mm to cache.

If a block is replaced from cache then write it back to mm if CPU performed write in it.

write

Hit
cpu performs write in cache

Miss
CPU brings missed block from mm to cache, then in cache the write operation is performed.

If a block is replaced from cache then write it back to mm only if CPU has performed write in it.

$$T_{avg\ read} = H * t_{cm} + (1-H)(t_{mm})$$

$$T_{avg\ write} = t_{mm}$$

actually $\max(t_{cm}, t_{mm})$

$$T_{avg} = \frac{\text{fraction of}}{\text{read operations}} * T_{avg\ read} + \frac{\text{fraction of write operations}}{} * T_{avg\ write}$$

Effective hit rate = Fraction of * hit rate
of write through cache   read operations

↓

when CPU accesses
only cache

$t_{cm} = 100ns \quad t_{mm} = 1000 ns$

$H = 90\%$

#Q. A system has a write through cache with access time of 100ns and hit ratio of 90%. The main memory access time is 1000ns. The 70% of memory references are for read operations.

1. Average memory access time for read operations only $= 0.90 * 100 + 0.1 * 1000$

$$= 190 ns$$

2. Average memory access time for write operations only $= 1000 \; ns$

3. Average memory access time for read-write operations both

$$\hookrightarrow = 0.7 * 190 + 0.3 * 1000$$

$$= 433 \; ns$$

4. Effective Hit ratio

$$= 0.7 * 0.9 = 0.63$$

#Q.   Size of data sent to main memory from CPU:

1. For write hit, when a write through cache is used? ⇒ 1 data size

2. For write miss, when a write through cache is used? ⇒ 1 data size

3. For write hit, when a write back cache is used? ⇒ Nothing

4. For write miss, when a write back cache is used? ⇒ Nothing

only when missed block
copied from mm to cache

#Q. Size of data sent from ~~main memory~~ to cache:

1. For write hit, when a <u>write through</u> cache is used?

$\left. \begin{array}{c} \\ \\ \end{array} \right\}$ nothing

← No write allocate

2. For write miss, when a write through cache is used?

3. For write hit, when a <u>write back</u> cache is used? ⇒ nothing

(write allocate)

4. For write miss, when a <u>write back</u> cache is used?

↘ 1 block

Tavg for write back :- (Hierarchical access)

$$T_{avg \ read \ or \ write} = H * t_{cm} + (1-H)(t_{cm} + t_{bt} + write \ back \ time)$$

write back time = $d * t_{bt}$

d = fraction of modified or dirty blocks replaced from cache.

$$Ans = 3.6625 \text{ ns}$$

#Q. The memory access time is 2 nanosecond for a read operation with a hit in cache, 10 nanoseconds for a read operation with a miss in cache, 4 nanoseconds for a write operation with a hit in cache and 15 nanoseconds for a write operation with a miss in cache. Execution of a sequence of instructions involves 100 memory read operations and 60 memory write operations. The cache hit-ratio is 0.9. The average memory access time (in nanoseconds) in executing the sequence of instructions is?

	Read	write
Hit	2	4
miss	10	15

$$T_{avg} \text{ Read} = 0.9 * 2 + 0.1 * 10 = 2.8 \text{ ns}$$

$$T_{avg} \text{ write} = 0.9 * 4 + 0.1 * 15 = 5.1 \text{ ns}$$

$$\text{fraction of read} = \frac{100}{160} = 0.625$$

$$\text{fraction of write} = \frac{60}{160} = 0.375$$

$$T_{avg} = \left(0.625 * 2.8\right) + \left(0.375 * 5.1\right)$$

$$= 3.6625 \text{ ns}$$

#Q. The memory access time is 1 nanosecond for a read operation with a hit in cache, 5 nanoseconds for a read operation with a miss in cache, 2 nanoseconds for a write operation with a hit in cache and 10 nanoseconds for a write operation with a miss in cache. Execution of a sequence of instructions involves 100 instruction fetch operations; 60 memory operand read operations and 40 memory operand write operations. The cache hit-ratio is 0.9. The average memory access time (in nanoseconds) in executing the sequence of instructions is?

H.W.

#Q.   Assume a miss rate of 2% for the instruction cache and of 4% for the data cache, a miss penalty of 100 cycles for all misses, and a frequency of 36% of loads and stores. If the CPI is 2 without memory stalls, determine how much faster the processor runs with a perfect cache that never misses.

2 mins Summary

Topic   Cache Write

Topic   Write Through & Write Back Cache

Topic   Write Allocate vs No Write Allocate