

COMPUTER SCIENCE

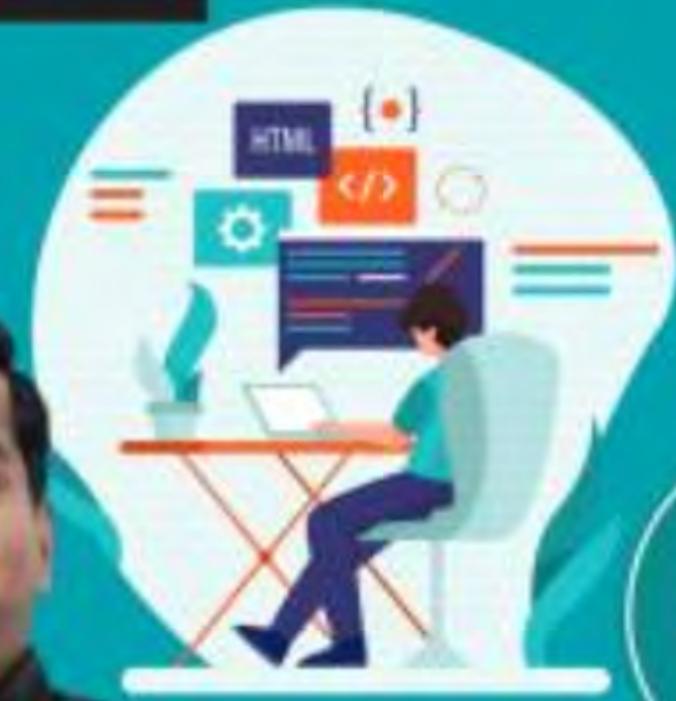
Computer Organization and Architecture

Secondary Memory & IO Interface

Lecture_01



Vijay Agarwal sir





A graphic of a construction barrier made of orange and white striped panels, with two yellow spherical bollards on top, positioned on the left side of the slide.

**TOPICS
TO BE
COVERED**

- o1 Updating Technique**

- o2 IO Organization**

Cache Memory

- Memory Org.
- Mapping Technique
 - Direct
 - set ASS.
 - Associative
- Replacement Algo
- Multi Level Cache
- Updating Technique

updating Technique :



updating Technique :

- CPU Perform Read \ominus Write operation On Cache Memory.
- When the Data [MM Block] is available in Cache then its known Read Hit & Write Hit.
- If the Data is Not available in the Cache then it is called Read Miss & Write Miss. then we have to perform Read allocate & Write Allocate.

updating Technique :

Read allocate: When there is a Read Miss in the Cache then transferring [Copying] that Data from Main Memory to Cache Memory is called Read allocate.

Write allocate: When there is write Miss in the Cache then transferring [Copying] that Data (MM Block) from Main Memory to Cache Memory is called Write Allocate.

⑧ What is the Read Miss, Write Miss, Read Hit if
Write Hit ?

T₁: MOV Y₁

On Byte of Bo

~~Defn~~

SV YL [0000];

remove
read

T₂: ADD δ_L , #47;

$\{ T_3 : \text{mov } [0000], \underline{x_L} : \leftarrow \text{memory write} \}$

Execution Sequence

$$\gamma_1 = 20$$

$$\Phi_2: \gamma_1 \leftarrow \gamma_1 + 47 \Rightarrow \gamma_1 = 20 + 47$$

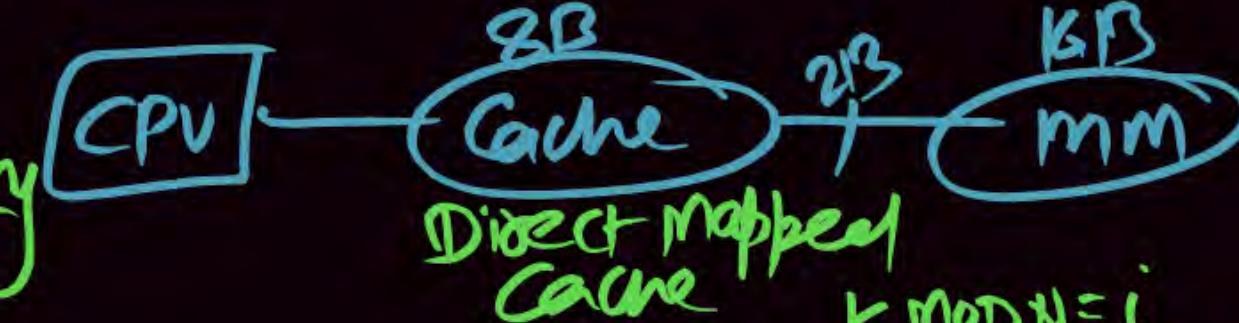
$$\Phi_3: \mathfrak{m}[\underline{0000}] \in \gamma_L$$

Bo

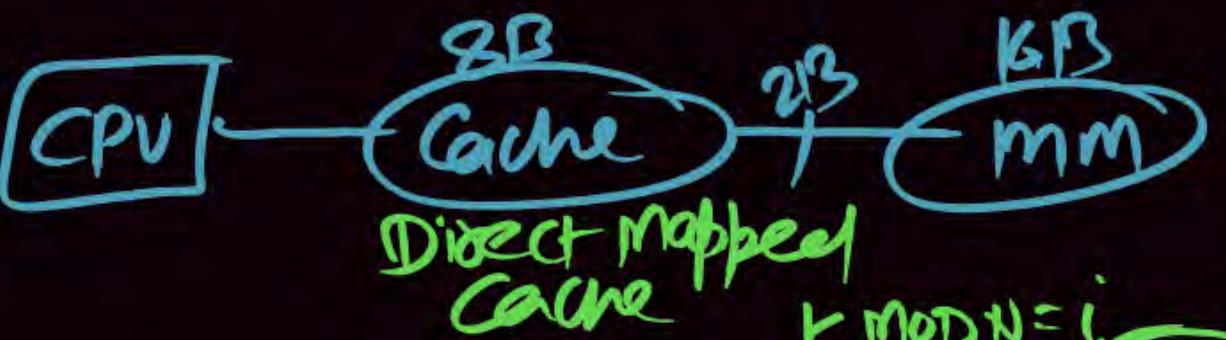
Cache initially Empty 01
→ Read Miss them 10
we perform Read allocate
(MM to CM) 11

→ write Hit
Cache write Hit

Initially Cache memory
is Empty



000	i	$B_0(2D)$	$x=20$
001	i	B_1	
010	i	B_2	
011	i	B_3	
100	i	B_4	
101	i	B_5	
110	i	B_6	
111	i	B_7	



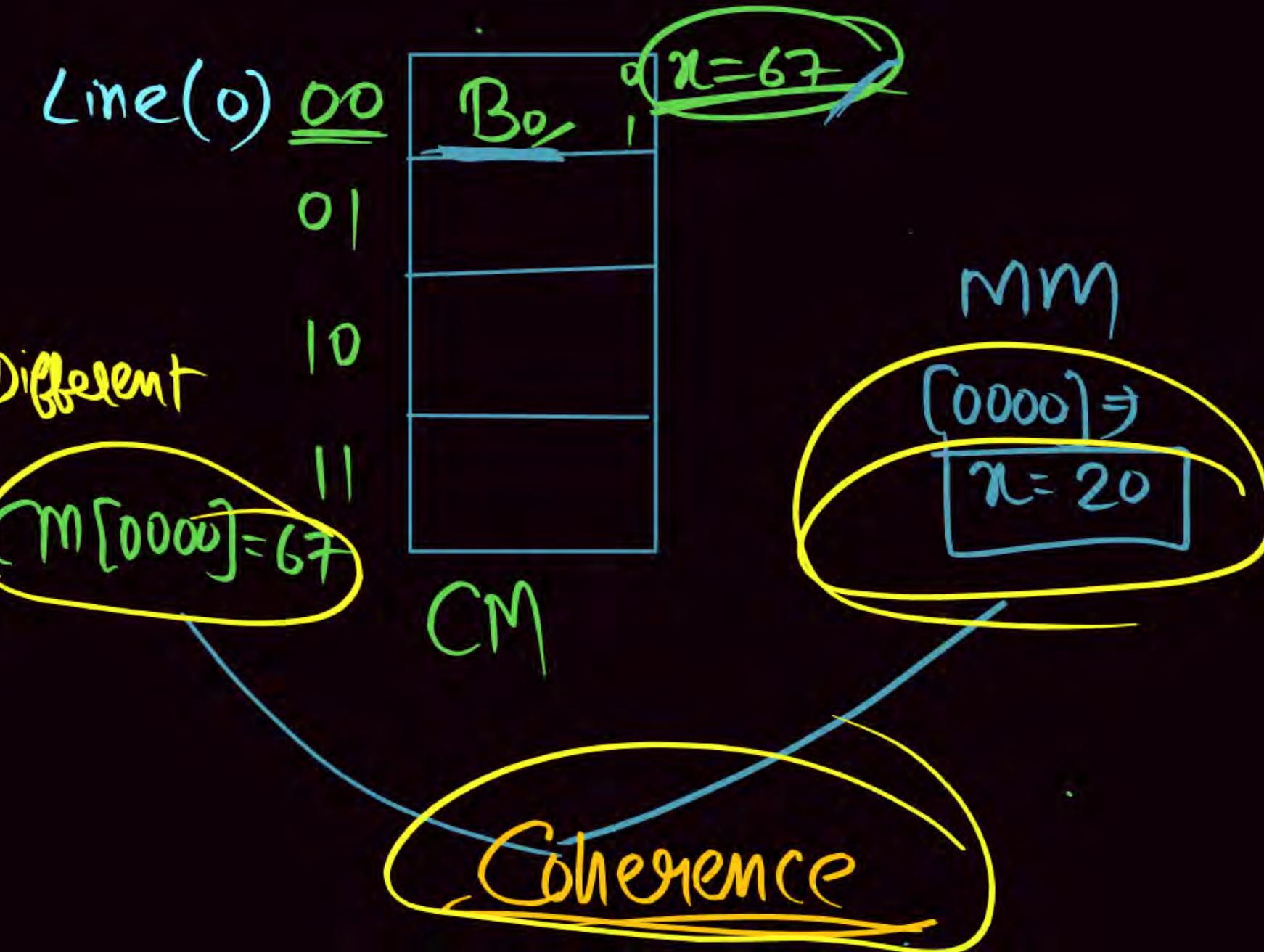
	$K \bmod N = i$	$0 \leq i < 2^3$
000	B_0	$i = 0$
001	B_1	$i = 1$
010	B_2	$i = 2$
011	B_3	$i = 3$
100	B_4	$i = 4$
101	B_5	$i = 5$
110	B_6	$i = 6$
111	B_7	$i = 7$

Coherence

Same address

Contain Different - Different

Value in MM & Cache
Memory

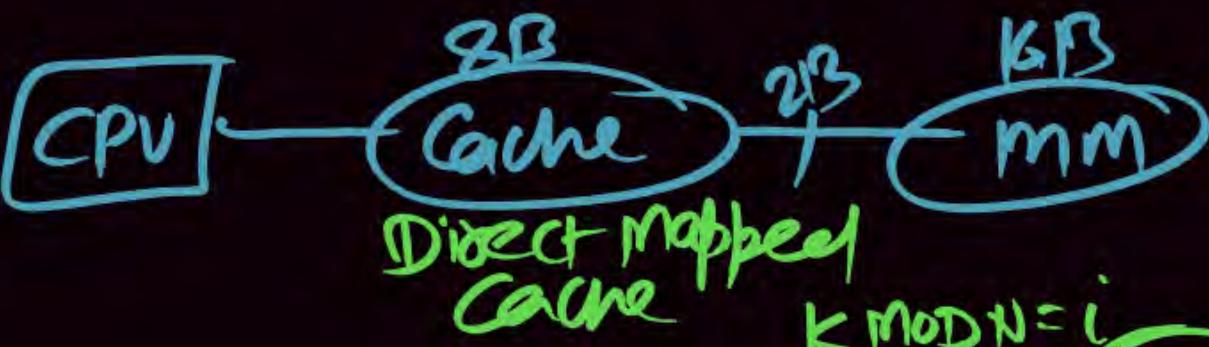


Q) Why Updating Technique is Required ?

(Ans) Assume in future there is a Cache Miss for any MM Block, which are Mapped into CM Line No '0' (in the previous example)

then this updated Data (of Block B₀) is Replaced by New Block
(All the update get lost)

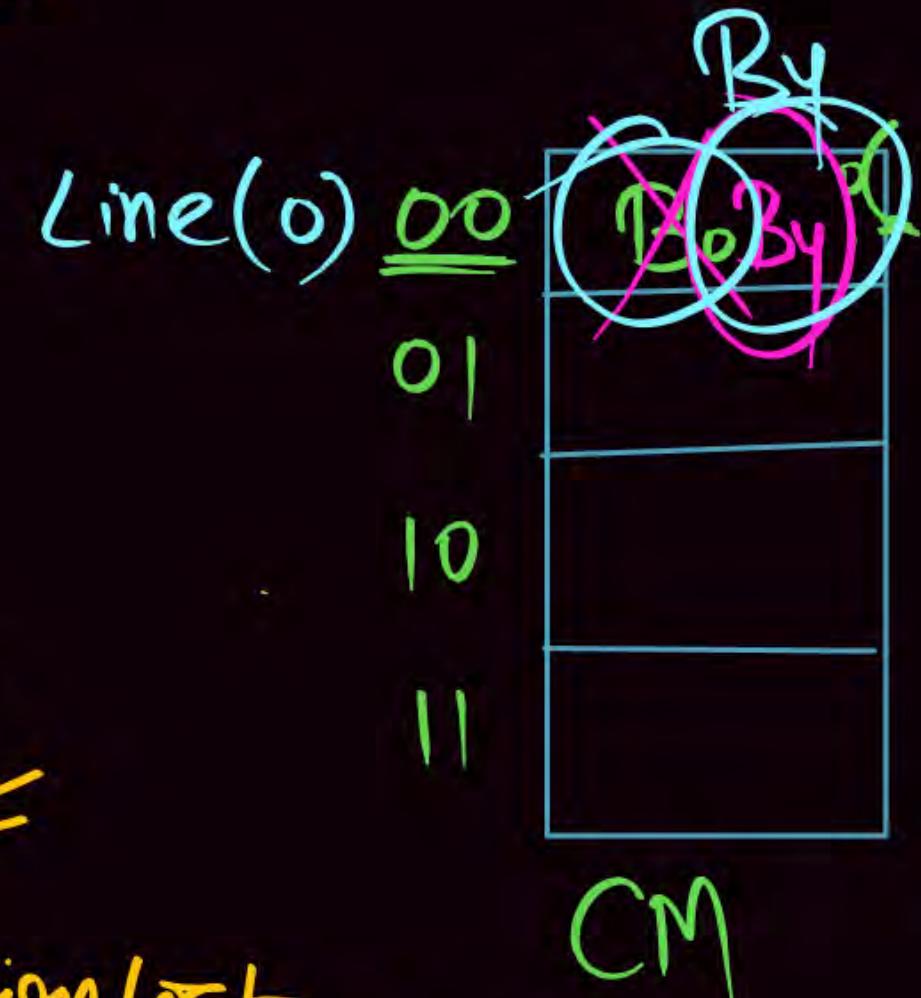
that means whatever work done by the CPU & load into the Cache get lost. Bz that block is Replaced by the New Main Memory Block. Now in the future if we Bring Block B₀ then we get the OLD Value.



Block B_0
Update lost

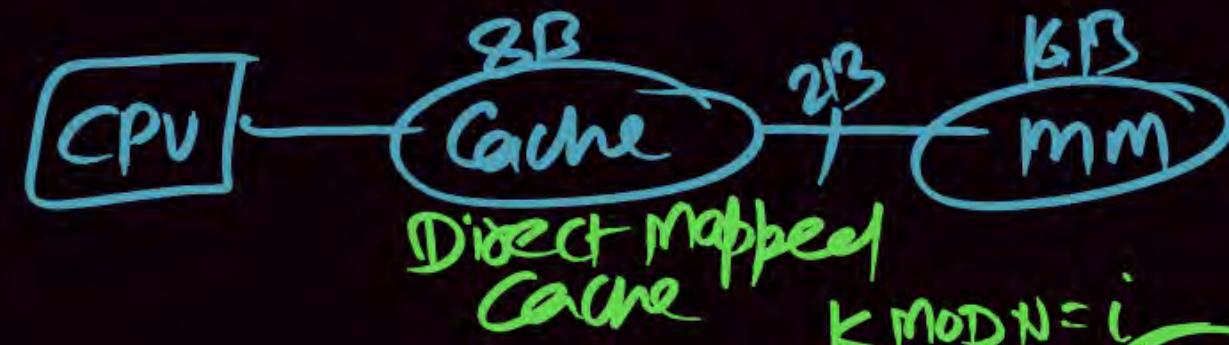
(D)

B_0 : Update lost
Due to B_4 Replacement



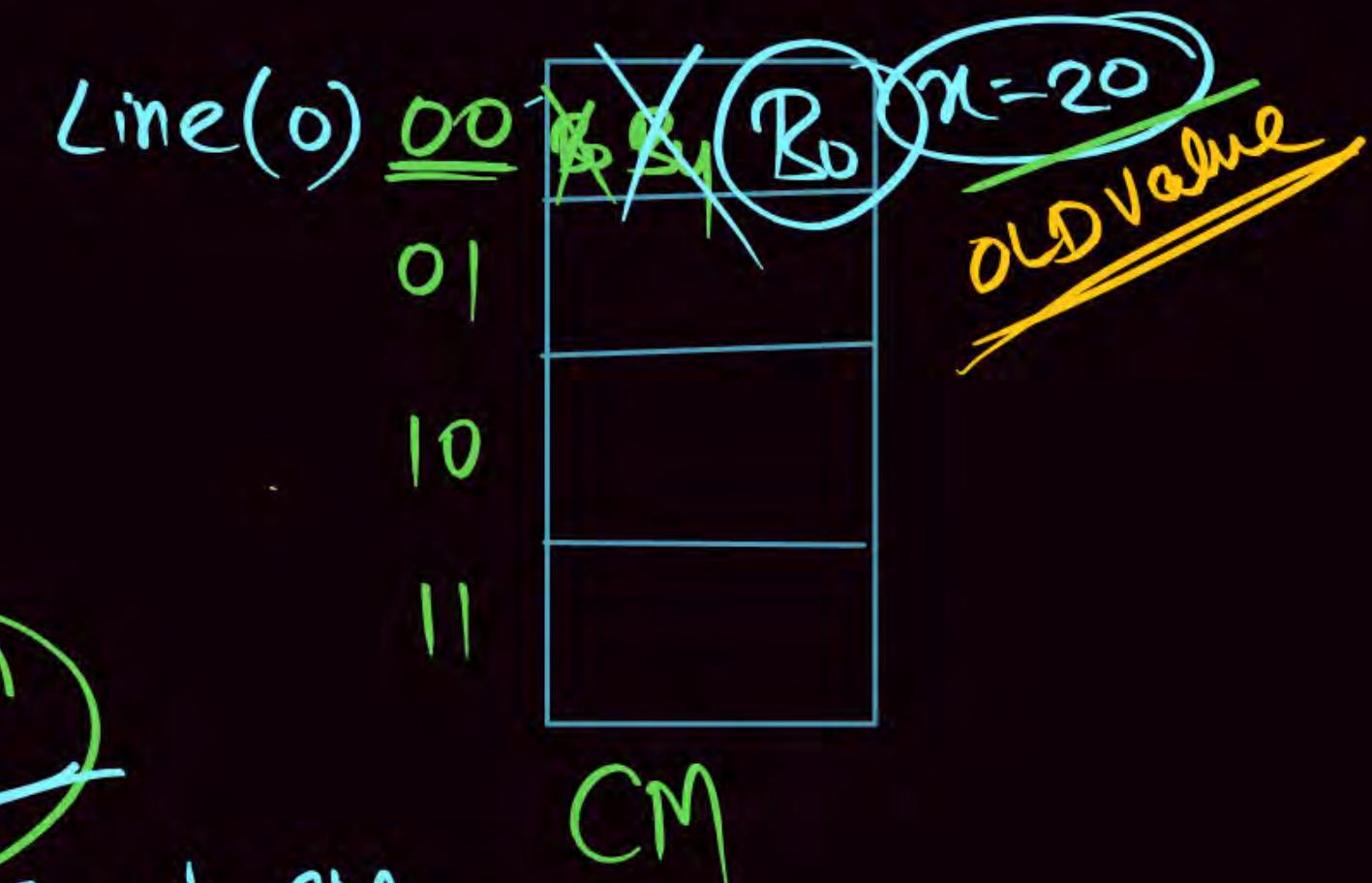
$K \bmod N = i$	B_i	$D_i = 2^3$	$J_i = 10$
000	B_0	$D_0 = 2^3$	$J_0 = 10$
001	B_1		
010	B_2		
011	B_3		
100	B_4		
101	B_5		
110	B_6		
111	B_7		

MM



~~Block B_0
Update or get lost~~

~~II~~



~~again Block B_0 come in CM
then getting the OLD value
so need of Updating Technique.~~

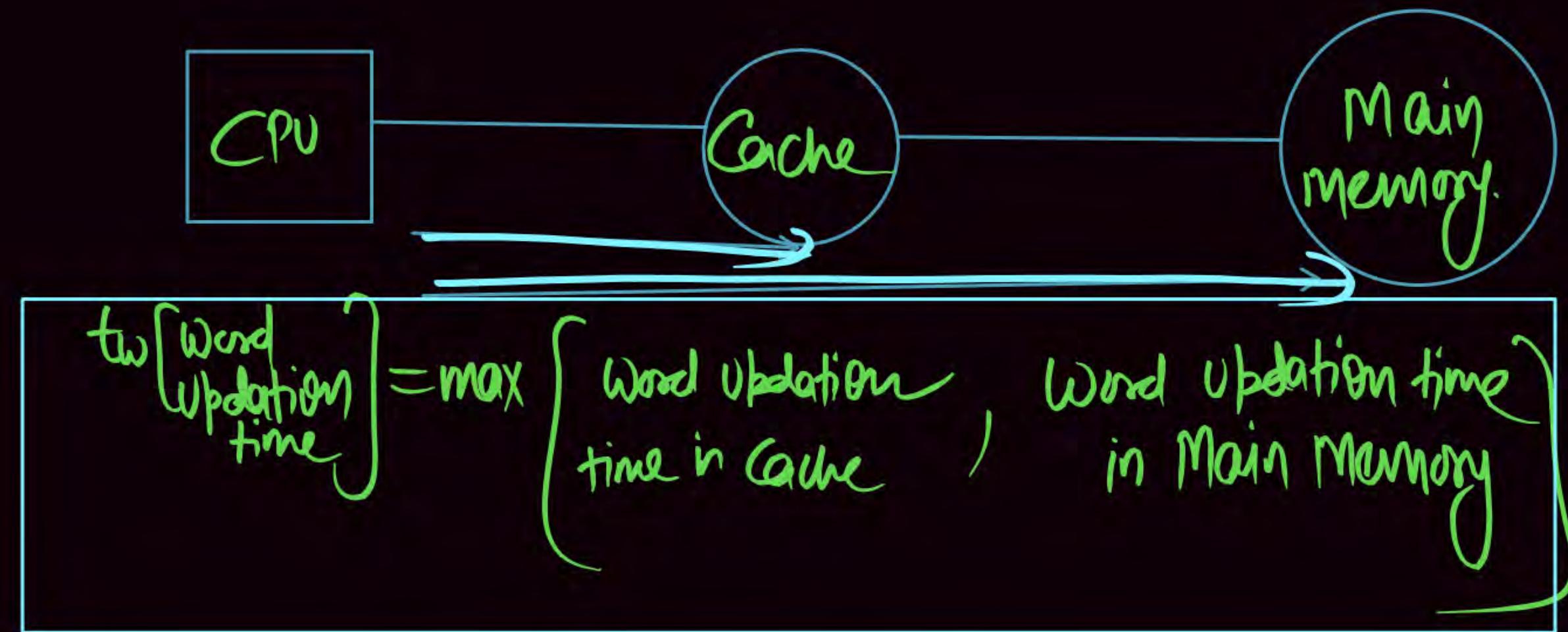
$k \bmod N = i$	$0x = 20$
000	1. $B_0 (20)$; $i = 0$
001	1. B_1
010	1. B_2
011	1. B_3
100	1. B_4
101	1. B_5
110	1. B_6
111	1. B_7

MM

updating Technique :

- ① Write Through (No Extra bit)
- ② Write Back (Extra bit)

① Write through : In the write through Both Cache Memory & Main Memory Update Simultaneously.
So No Problem of Coherence (No inconsistency)



Hierarchical

$$T_{avg\text{read}} = h_r t_c + (1-h_r) (t_m + t_c)$$

Read Hit Read Data Read miss Read Allocate Read Data from CM .

$$T_{avg\text{write}} = h_w t_w + (1-h_w) (t_m + t_w)$$

Write Hit Word Update Write Miss Write Allocate Word Update .

$$T_{avg_{WT}} = \frac{\% \text{ of Read operation}}{} \times T_{avg_{read}} + \frac{\% \text{ of write operation}}{} \times T_{avg_{write}}$$

$$N_{WT} > \frac{1}{T_{avg_{WT}}}$$

Dis Advantage in WT: If there is a frequent updation in Cache, then
100 updation(write)
~~100 MM~~ → 100 MM also Access

Simultaneous Access

$$\bar{t}_{avg,read} = h_0 t_C + (1-h_0) t_M$$

$$t_{avg,write} = t_W.$$

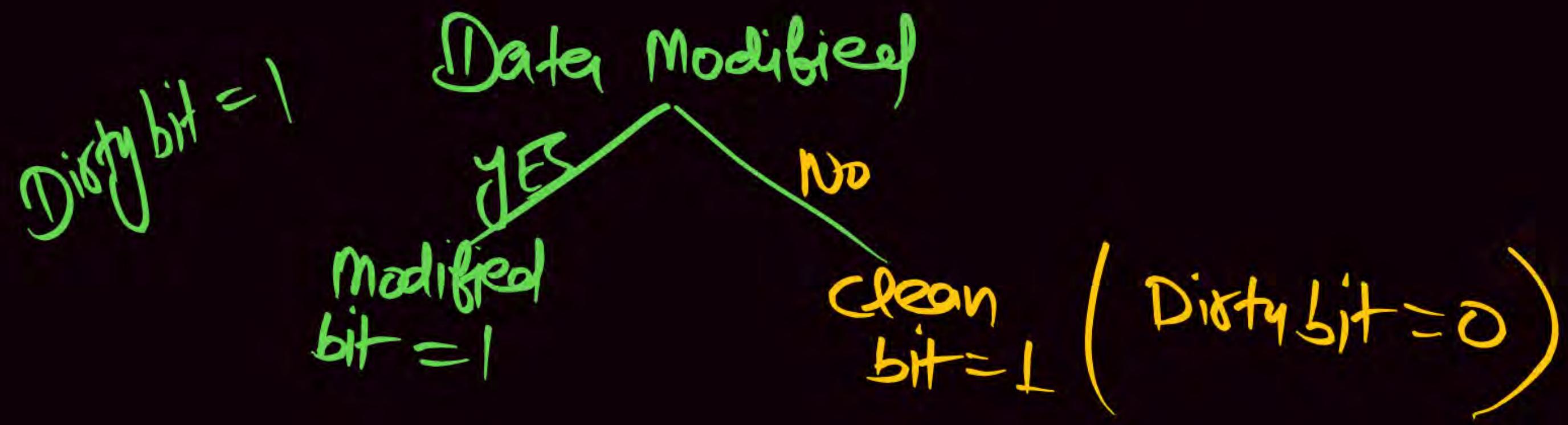
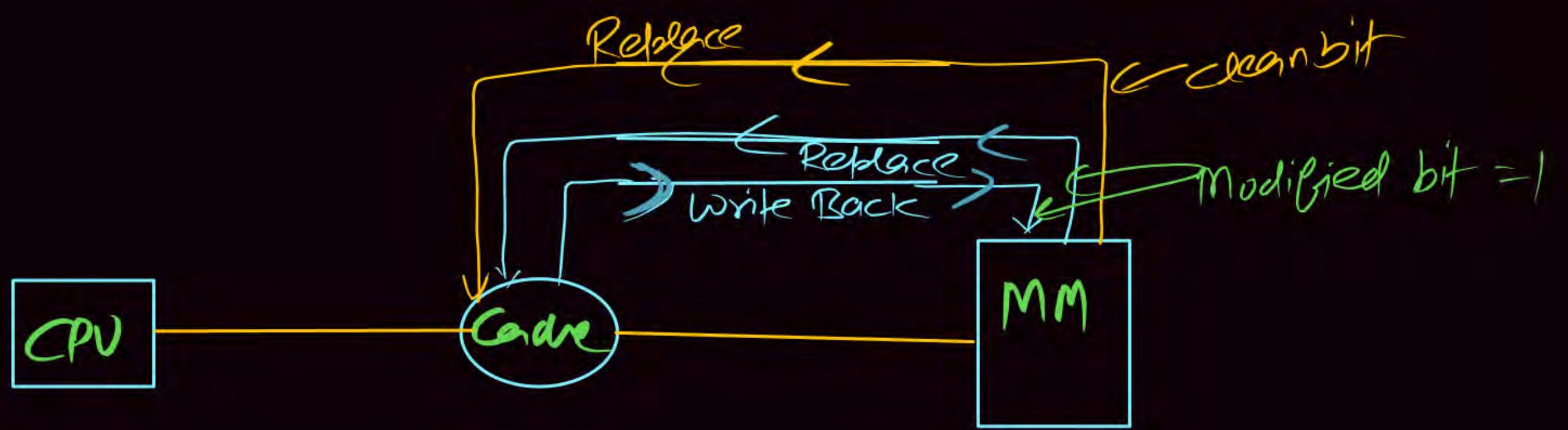
② Write Back Updating Technique: In the WB Updating Technique
Only Cache Memory Update,
Main Memory Update later (So in WB Cache Memory Update
& MM update later So Coherence is Present).

• But this Coherence does not Create Any Problem.

WHY? (why Coherence does not create Any Problem)

Because in WB, Each Cache Line Maintains Some Extra bits [like Valid|Invalid , Dirty bit (Modified bit)]. to Maintain the Status of CM Block .

- If the CM Block is Updated the Modified bit(Dirty bit) is set to 1. If Block is Not Updated then Clean bit is set [Dirty bit = 0].
- Before Replacing the CM Block we check the Status of the Block.
- If a Block is a Updated Block (Modified bit set) then firstly this Updated Data Write Back into Main Memory (Higher memory) then Replace the CM block with Any New (Request) Block.
- If Block is Not Updated Block then Directly Replace with New Main Memory (Requested) Block .



$$T_{avg\ read} = h_r t_c + (1-h_r) \left[\frac{1}{\% \text{ Data}} \underset{\substack{\downarrow \\ \text{modified}}}{{t_n + t_m + t_c}} + \frac{1}{\% \text{ Clean}} \underset{\substack{\downarrow \\ \text{bit}}}{{t_m + t_c}} \right]$$

$$T_{avg\ write} = h_w t_c + (1-h_w) \left[\frac{1}{\% \text{ Data}} \underset{\substack{\downarrow \\ \text{modified}}}{{t_y + t_m + t_c}} + \frac{1}{\% \text{ Clean}} \underset{\substack{\downarrow \\ \text{bit}}}{{t_m + t_c}} \right]$$

$$T_{avg\ WB} = f_r \times T_{avg\ read} + f_w \times T_{avg\ write}$$

$$\eta_{WB} = \frac{1}{T_{avg\ WB}}$$

f_r : % of Read operation
 f_w : freq(.) of work operation

updating Technique :

Q.

Let WB and WT be two set associative cache organizations that use LRU algorithm for cache block replacement. WB is a write back cache and WT is a write through cache. Which of the following statements is/are FALSE?

P
W

[2022: MSQ 1M]

- A ~~false~~ Each cache block in WB and WT has a dirty bit.
- B ~~false~~ Every write hit in WB leads to a data transfer from cache to main memory. [Before Replace (Modified bit Set)
then WB CM to MM then Replace with New MM Block]
- C ~~Eviction of a block from WT will not lead to data transfer from cache to main memory.~~ : True.
- D ~~false~~ A read miss in WB will never lead to eviction of a dirty block from WB. [Eviction Demand On Mapping Tech & Replacement Algo]

While Cache Accessing.

①

Physical Index Cache (MM Address is used in Mapping)

②

Virtual Index Cache (VM Address is used in Address Mapping)

(VI VT)

(↳ Virtual Index Virtual Tag.)

Till Now

LIL + GATE PyQ's Done

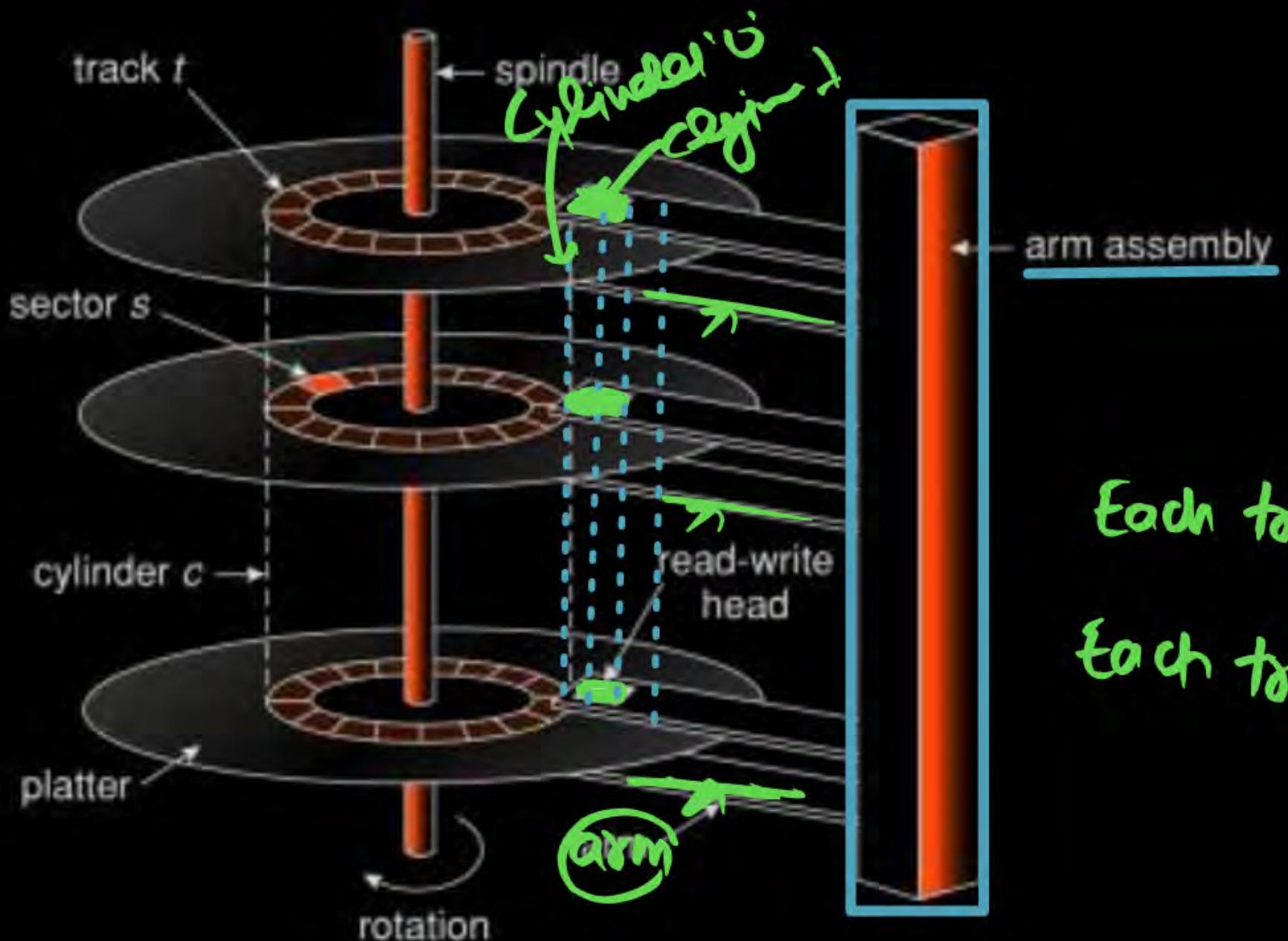


Secondary-memory

Moving - Head Disk Mechanism

: Disk

P
W



Same tracks will form a cylinder

Each track Number 0 \Rightarrow cylinder '0'
Each track Number 1 \Rightarrow cylinder '1'

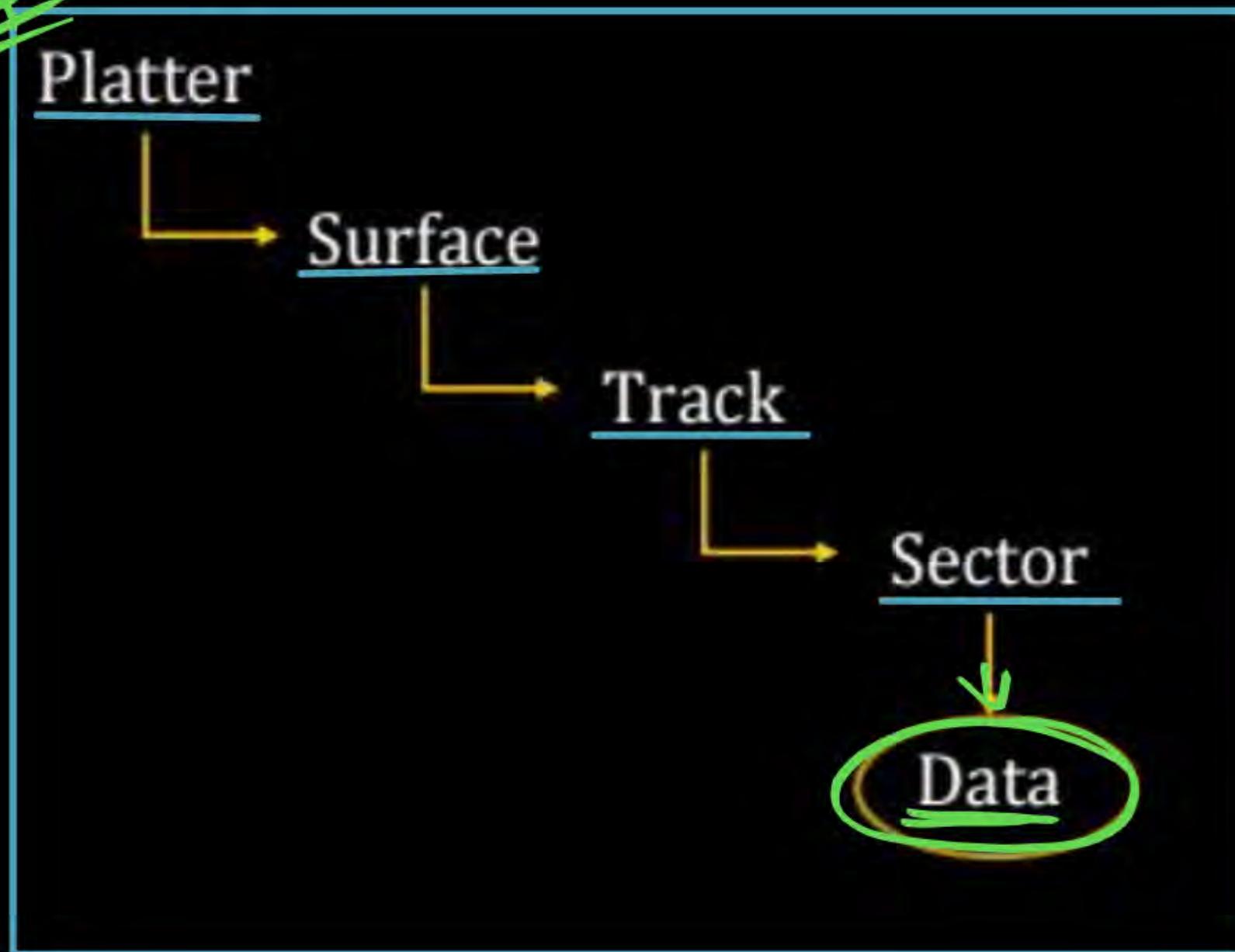
Magnetic disks provide the bulk of secondary storage for modern computer systems. Conceptually, disks are relatively simple.

Each disk platter has a flat circular shape, like a CD. Common platter diameters range from 1.8 to 3.5 inches.

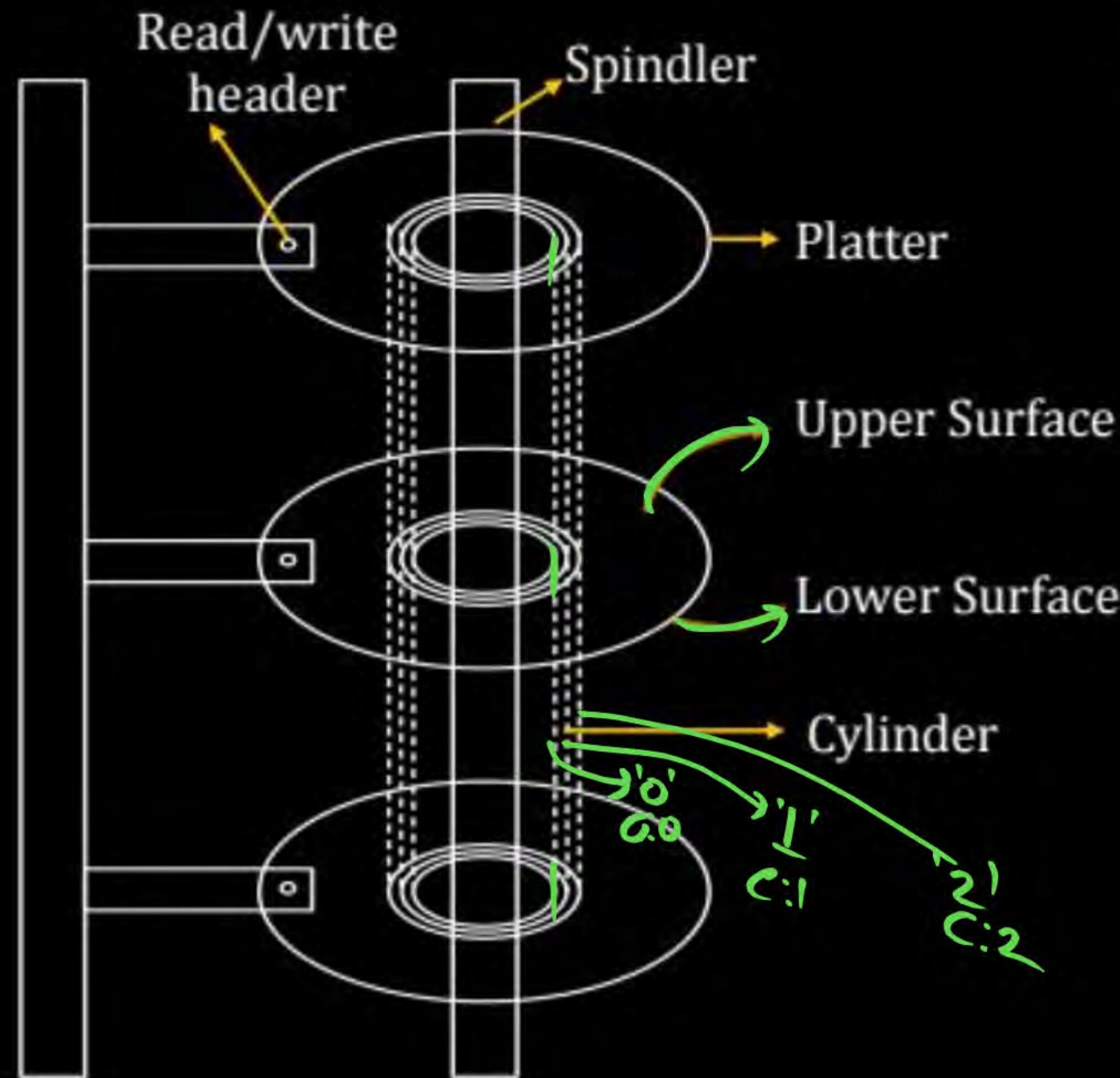
The two surfaces of a platter are covered with magnetic heads are attached to a disk arm that moves all the heads as a unit.

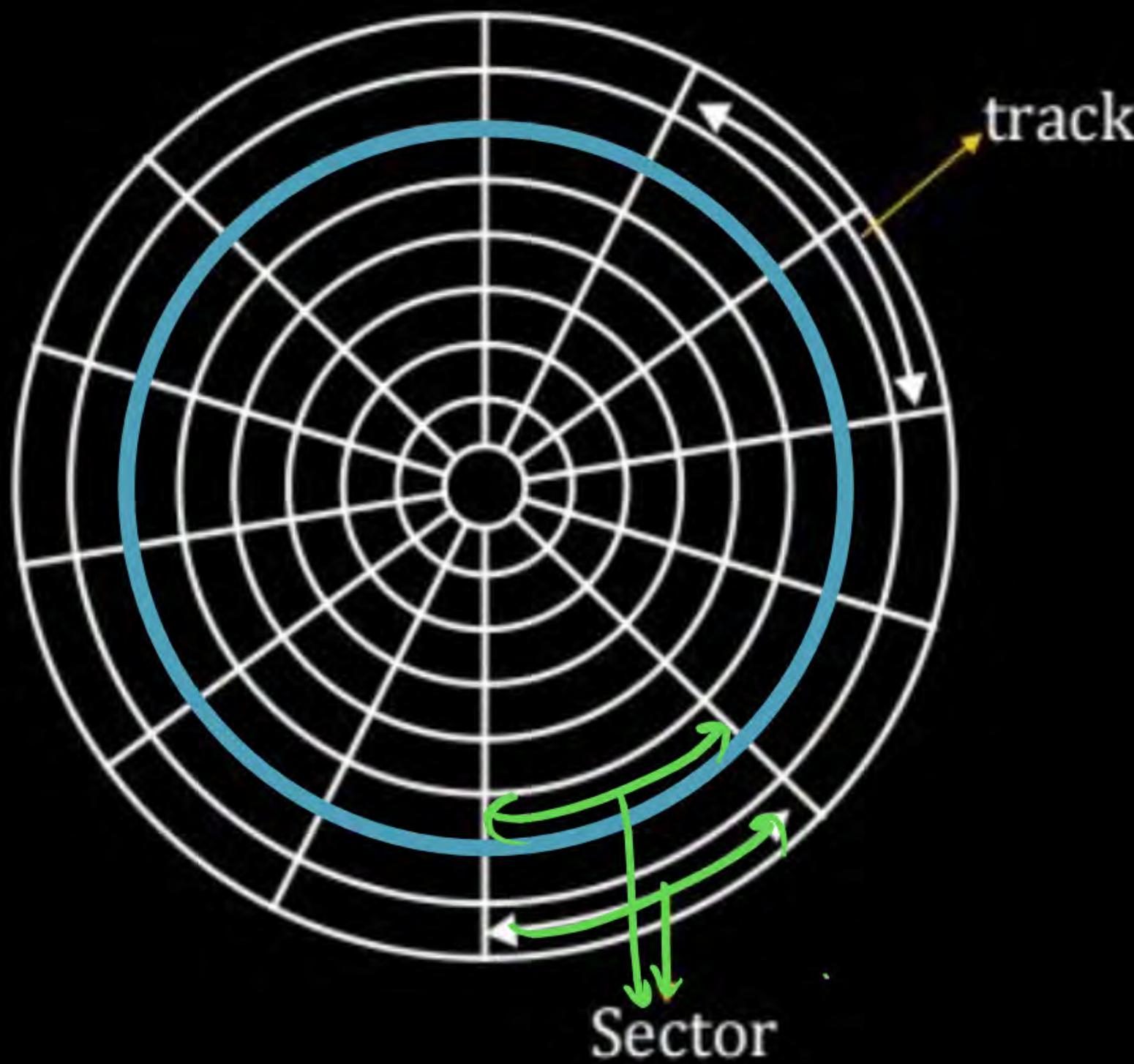
The surface of a platter logically divided into circular tracks, which are subdivided into sectors.

The set of tracks that are at one arm position makes up a cylinder. There may be thousand of concentric cylinders in a disk drive, and each track may contain hundreds of sectors. The storage capacity of common disk drives is measured in gigabytes.



Disk - structure





Header(or)
(pre-amble)



CRC
(Cyclic
Redundancy
check)

Sector
Or
Disk Block

Platter

Surface

Track

Sector



Disk

- ① Capacity
- ② Access time (S.T., R.L., D.T.T & Data transfer Rate)
- ③ Disk Addressing $\langle C, h, S \rangle$.

Q. Consider a disk which has 16 platter, each platter has two surface. Every surface has 1K Track is further Divided into 512 sector and every sector can store the 8KB Data, then calculate.

- (i) What is the capacity of Disk?
- (ii) How many bits are required to identify any particular sector of the Disk?

Solution (i) 1 platter - 2 surface

16 platter - 16×2 i.e. 32 surface

1 surface - 1K Track

32 surface - $32 \times 1K$ track \Rightarrow 32K Track

1 track - 512 sector

32k Track \Rightarrow $32K \times 512$ sector

Each sector capacity \Rightarrow 2KB

Total Disk capacity = $32K \times 512 \times 8KB$

$$= 2^5 \times 2^{10} \times 2^9 \times 2^{13} \Rightarrow 2^{37} B$$

Disk capacity = 128 GB

(ii) #bits required to represent sector in a disk

$$= 16 \times 2 \times 1K \times 512$$

$$= 2^4 \times 2^1 \times 2^{10} \times 2^9 \Rightarrow 2^{24} = 24 \text{ bits}$$

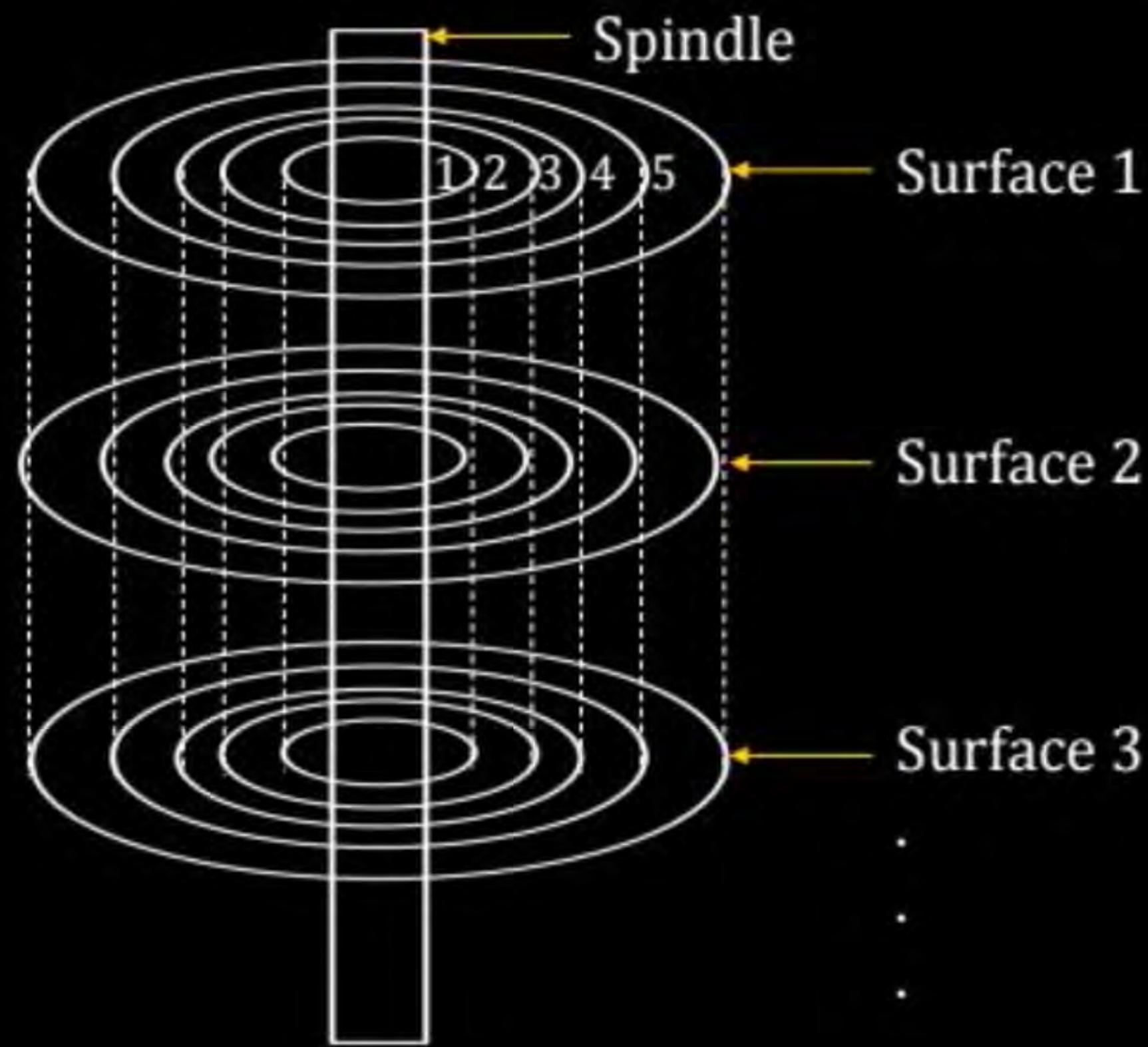
Q.

Consider a disk pack with 16 surfaces. 128 tracks per surface and 256 sectors per track. 512 bytes of data are stored in a bit serial manner in a sector. The capacity of the disk pack and the number of bits required to specify a particular sector in the disk are respectively

- (a) 256 Mbyte, 19 bits
- (b) 265 Mbyte, 28 bits
- (c) 512 Mbytes, 20 bits
- (d) 64 Gbyte, 28 bits

[GATE - 2007]

Same track number in all the surface will form a cylinder.



cylinder in the disk = # track, in the surface

Track capacity = # sector tracks * # Bytes / sector

Cylinder capacity = # surface in the disk * track capacity

Disk capacity = # cylinder in the disk * cylinder capacity

To access the data from the hard disk different adjustments are required in the hard disk so, the associative adjust latencies are -

- (1) Seek time
- (2) Rotational time
- (3) transfer time

Disk I/O operation

- Seek time
- Rotational Latency
- Transfer time
- Transfer rate

- The read / write header can never be outside the track it will be pointing to any particular track of the surface.
- read / write header will be move in forward & backward direction & disk in one direction (either clockwise or anti-clockwise)

- 1) **Seek time:** The amount of time taken to move the read / write header from its current position to the desired track is called as seek time.
- 2) **Rotating Latency:** The amount of time taken to relate the track when the read / write header comes to exact position (sector)
The Rotation Latency is considered as $= \frac{1}{2}$ Rotation time
- 3) **Transfer time:** The amount of time taken to transfer the required data.
- 4) **Transfer rate:** The number of bytes found for unique line is called as transfer rate of disk.

Q.

Consider a disk system, which has an average seek time of 30 ns and rotational rate of the disk is 360 RPM. Each track of the disk has 512 sector, each of the size 512 Byte, then calculate.

- (i) Average Seek time
- (ii) Average rotational latency
- (iii) data transfer time for 4 sector (continuous)?
- (iv) Data transfer rate?



(i) Seek time = 30 ns = 30×10^{-9} sec

(ii) Rotational Latency \Rightarrow 360 rotation = 60 second

$$\frac{1}{2} \text{ rotation} = \frac{1/2 \times 60}{360} = \frac{1}{12} \text{ sec}$$

$$R.L = 0.083 \text{ sec.}$$

(iii) Transfer time \Rightarrow In one rotation time, we can read the total size of the track & to read the required data how much time is required?

1RT = we can read total size of data

? = To read required data.

Rotation time = 1/8 second

Total size of 1 track = $2^9 \cdot 2^9 \text{ B} \Rightarrow 2^8 \cdot 2^{10} \text{ B} = 256 \text{ KB}$

Required data = 4 sector

$$= 2^2 \cdot 2^9 \text{ B} \Rightarrow 2^1 \cdot 2^{10} \text{ B} = 2 \text{ KB}$$

1 Track

256 KB \rightarrow 1/6 sec

1B = 256 KB \rightarrow 1/6 second

$$2 \text{ KN} \rightarrow \frac{\frac{1}{6} \times 2 \text{ KB}}{256 \text{ KB}} = 0.0013 \text{ sec}$$

(iv) Data transfer rate

In one rotation time \rightarrow we can transfer total size of the track.

In one sec \rightarrow how much data we will transfer

$$\frac{1}{6} \text{ sec} \rightarrow 256 \text{ KB}$$

$$1 \text{ second} \rightarrow \frac{256 \text{ KB} \times \text{sec} \times 6}{\text{sec}} = 1536 \text{ KBPS}$$

MCQ

An application loads 100 libraries at start-up. Loading each library requires exactly one disk access. The seek time of the disk to a random location is given as 10 ms. Rotational speed of disk is 6000 rpm. If all 100 libraries are loaded from random locations on the disk, how long does it take to load all libraries? (The time to transfer data from the disk block once the head has been positioned at the start of the block may be neglected.)

[GATE-2011-CS: 2M]

A 0.50s

B 1.50s

C 1.25s

D 1.00s

Consider a disk pack with a seek time of 4 milliseconds and rotational speed of 10000 rotations per minute (RPM). It has 600 sectors per track and each sector can store 512 bytes of data. Consider a file stored in the disk. The file contains 2000 sectors. Assume that every sector access necessitates a seek, And the average rotational latency for accessing each sector is half of the time for one complete rotation. The total time (in milliseconds) needed to read the entire file is ____.

[GATE-2015(Set-1)-CS: 2M]

NAT

Consider a typical disk that rotates at 15000 rotations per minute (RPM) and has a transfer rate of 50×10^6 bytes/sec. If the average seek time of the disk is twice the average rotational delay and the controller's transfer time is 10 times the disk transfer time, the average time (in milliseconds) to read or write a 512-byte sector of the disk is ____.

[GATE-2015(Set-2)-CS: 2M]

Q.

A certain moving arm disk storage, with one head, has the following specifications.

Number of tracks/recording surface = 200

Disk rotation speed = 2400 rpm

Track storage capacity = 62,500 bits

The average latency of this device is P msec and the data transfer rate for all track in 1 surface is Q bits/sec.

Write the value of P and Q

[GATE-: 2 Marks]

Q. If the disk is rotation at 3600 rpm, determine the effective data transfer rate which is defined as the number of bytes transferred per second between disk and memory. (Given size of track = 512 bytes)

[GATE-: 2 Marks]

**THANK
YOU!**

