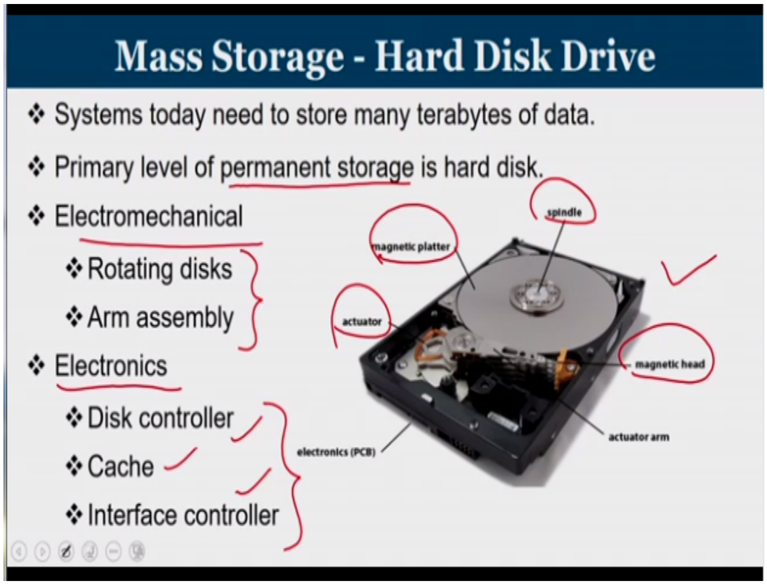
# Advanced Computer Architecture
**Prof. Dr. John Jose**
**Assistant Professor**
**Department of Computer Science and Engineering**
**Indian Institute of Technology-Guwahati**

## Lecture-27
## Secondary Storage Systems

Welcome to lecture number 19, in lecture 19 we have planned to discuss on secondary storage systems. Over the last 2, 3 weeks we had an overview of how cache memory works and how main memory works and both cache and main memory are volatile systems. We move onto the next phase which is the non volatile or permanent storage. So every system should have a permanent storage and depending on the need data and instruction are moved from the permanent storage into your main memory to your cache hierarchy and to processor.

**(Refer Slide Time: 01:15)**



Mass Storage - Hard Disk Drive

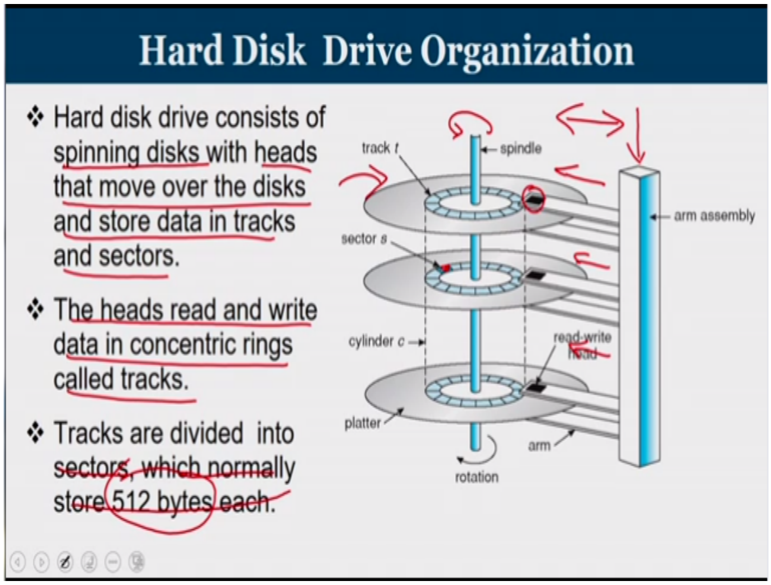
- Systems today need to store many terabytes of data.
- Primary level of permanent storage is hard disk.
- Electromechanical
  - Rotating disks
  - Arm assembly
- Electronics
  - Disk controller
  - Cache
  - Interface controller

Systems today need to store many terabytes of data and the primary level of permanent storage is your hard disk. And this is how hard disks are looking like and generally all our data centers also requires huge amount of storage. So storing of permanent data and data is increasing day by day it is getting multiplicated in exponential rate and then huge data we have and storing of them is a challenging task.

But in this course we are not going heavily into how the storage of a data center is but rather we try to understand how basically a hard disk work, what are the challenges involved in designing of a hard disk. And the same problems we can escalate it to these storage systems and raid system of data centers. Generally for permanent storage devices like hard disk, we have an electro mechanical component and then electronics component.

The electro mechanical component consist of rotating disk arm assembly like that. But you can see that you have a magnetic head, then you have a spindle that rotate then there is a platter plus actuator. And then you have the electronics component of it which consist of a controller plus interface and plus the disk cache.

**(Refer Slide Time: 02:26)**



Hard Disk Drive Organization

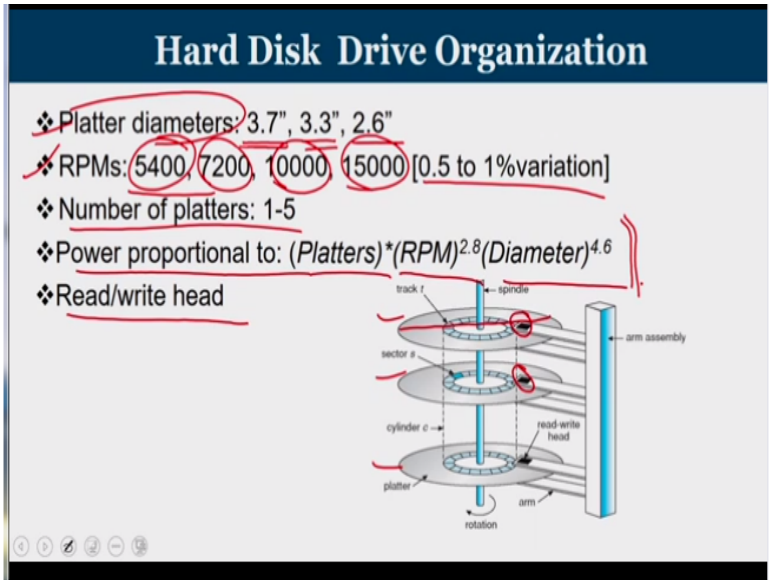
- Hard disk drive consists of spinning disks with heads that move over the disks and store data in tracks and sectors.
- The heads read and write data in concentric rings called tracks.
- Tracks are divided into sectors, which normally store 512 bytes each.

When you look at the hard disk drive organization, it basically consist of spinning disk with heads that move over the disk and store data in tracks and sectors. So this is called your arm assembly, this arm assembly can move in this direction. So as on when the arm assembly moves this one will go inside and outside and this spindle is going to rotate. So due to the rotation of these spindle you can this arm assembly or rather the head can touch any portion, it can touch any track.

So we divide into concentric tracks in this plate and then we have different sectors in each of this track and data is stored there using magnetic properties. So the arm assembly will move together, so if this going to move inside all other things will move inside the same time. So that is one kind of the movement then second one is spindle rotation such that the appropriate track will get under the read head.

The heads read and write data in concentric ring called tracks and tracks are divided into sectors and which can generally stored 512 bytes of data. So this is basically your sector and sector consist of blocks and these block can store 512 bytes of data.

**(Refer Slide Time: 03:47)**



Hard Disk Drive Organization

- ❖ Platter diameters: 3.7", 3.3", 2.6"
- ❖ RPMs: 5400, 7200, 10000, 15000 [0.5 to 1%variation]
- ❖ Number of platters: 1-5
- ❖ Power proportional to: (Platters)*(RPM)$^{2.8}$(Diameter)$^{4.6}$
- ❖ Read/write head

Generally the diameter of your platters that is a surfaces, the cylindrical surface, the circular surface is of 3.7 inch, 3.3 inch and 2.6 inch. And this plates are been rotated with the help of a spindle and the number of revolutions per minute is roughly 5400 and ranging of 7200, 10000, 15000. So we call it as rpm speed and there can be some 1% of variation also it may not be exactly possible for us to design a circuitry which will have 15000 rpm in 1 minute.

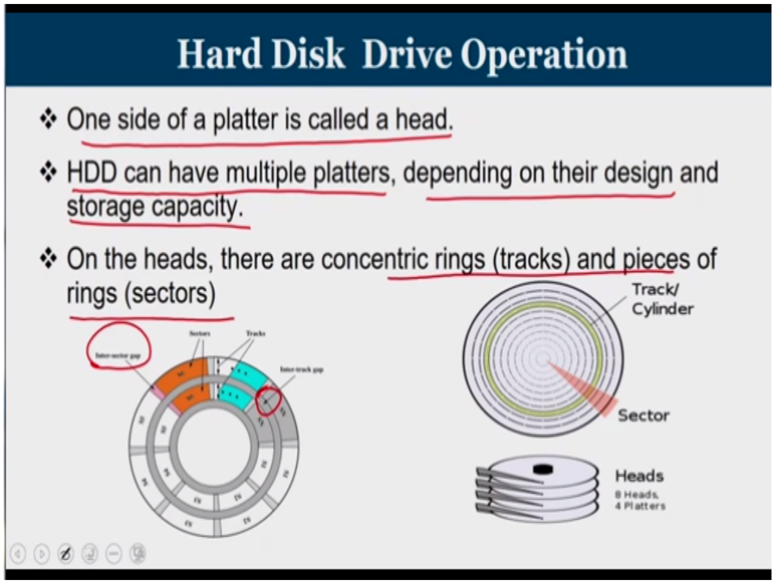
And your number of platters also vary from 1 to 5 and the power is proportional to platters into rpm ratio 2.8 into diameter rise to 4.6.

$$\text{power} \propto \text{rpm ratio} * \text{diameter}$$

We are not going for the derivation of this those were interested can look into the textbooks and material from the internet. And then we have a read and write head that is been associated with this. So generally if you look at this the platter diameter what I mentioned is this is called a diameter of the platter.

And then at what rate this spindle is going to rotate and the rotations per minute the rpm is been given with 5400, 7200, 10000 and all and we have different platters. So in this case we have platters 1, 2, 3 like that different platters are there and then we have the read write heads, these are the read, write heads.

**(Refer Slide Time: 05:20)**

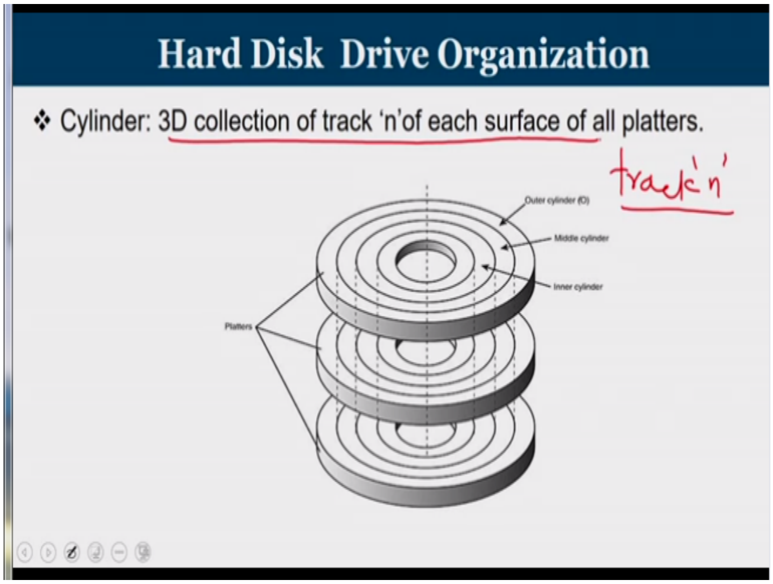


## Hard Disk Drive Operation

- One side of a platter is called a head.
- HDD can have multiple platters, depending on their design and storage capacity.
- On the heads, there are concentric rings (tracks) and pieces of rings (sectors)

Now once side of the platter is typically known as a head and hard disk drive can have multiple platters depending on the design and storage capacity. Now on the heads if you look at there are concentric rings or tracks and there are pieces of ring that we known as sectors. So it is very clear in the diagram what do you mean by tracks you have different tracks are there and then for each track there is an inter sector gap between each sectors there is an inter sector gap and between each track we have the inter track gap also.

**(Refer Slide Time: 05:51)**



## Hard Disk Drive Organization

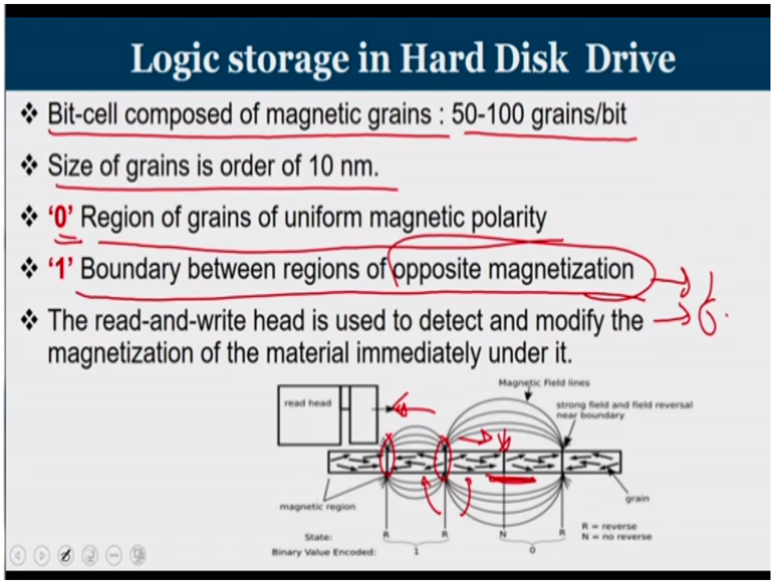
- Cylinder: 3D collection of track 'n'of each surface of all platters.

track 'n'

Now when we come to the cylinder concept if you take of track n of all the platters then that is what is known as cylinder concepts. So it is a 3Dcollection of track n of each of the surfaces, so basically this will give us an understanding about. Let us say you are going to write something on track 3 of 1 surface, then the remaining portion of the data it is not stored in the adjacent track, it is stored on the same track of a different platter.

Such that without head movement I can read and write from the other platters, so the set of all tracks of the same number of all platters put in together that is called a cylinder concept and cylinder is a 3 dimensional concept.

**(Refer Slide Time: 06:34)**



## Logic storage in Hard Disk Drive

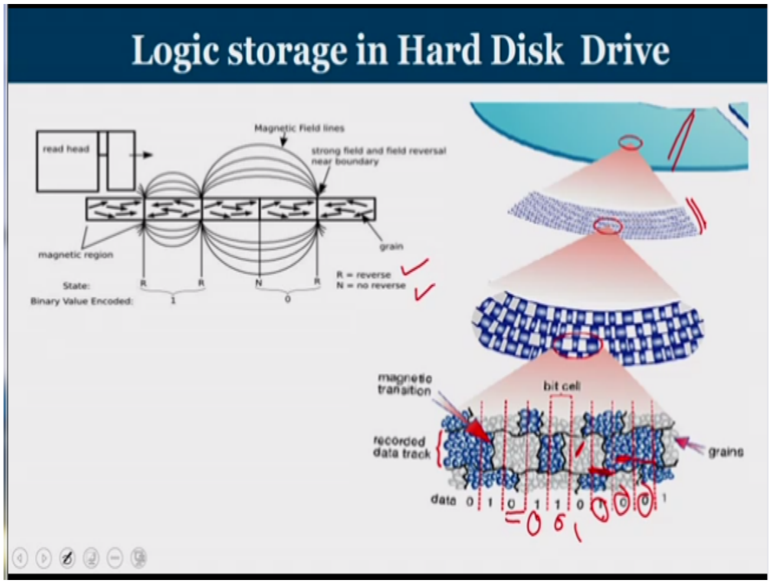
- Bit-cell composed of magnetic grains : 50-100 grains/bit
- Size of grains is order of 10 nm.
- '0' Region of grains of uniform magnetic polarity
- '1' Boundary between regions of opposite magnetization
- The read-and-write head is used to detect and modify the magnetization of the material immediately under it.

Now let us try to understand how are you going to save your data or what is logic 1 and what is logic 0 as far as a bit cell inside the hard disk is concerned. So the bit cell composed of magnetic grains. And typically 50 to 100 grains are used to represent the bit and size of the grains is in the order of 10 to 20 nanometers. And 0 means the region of grains of uniform magnetic polarity, so when all the grains within a region is of uniform magnetic polarity we call it as 0.

If the boundary between regions if they have opposite magnetization then we call it as 1. So the read and write head is use to detect and modify the magnetization of the material immediately under it. So here if you look into here you can see that this is a boundary of 1 region and these boundaries are points wherein either side of the boundary you have magnetization in different.

So here the magnetization in this direction whereas in this it is different, so when there is a difference that is because of opposite magnetization we call it as 1. Now if you look into this case you can find that they are in the same alignment, so this junction will basically represent as 0. So 0 means they are of uniform magnetic polarity and 1 means they are opposite magnetic polarity.

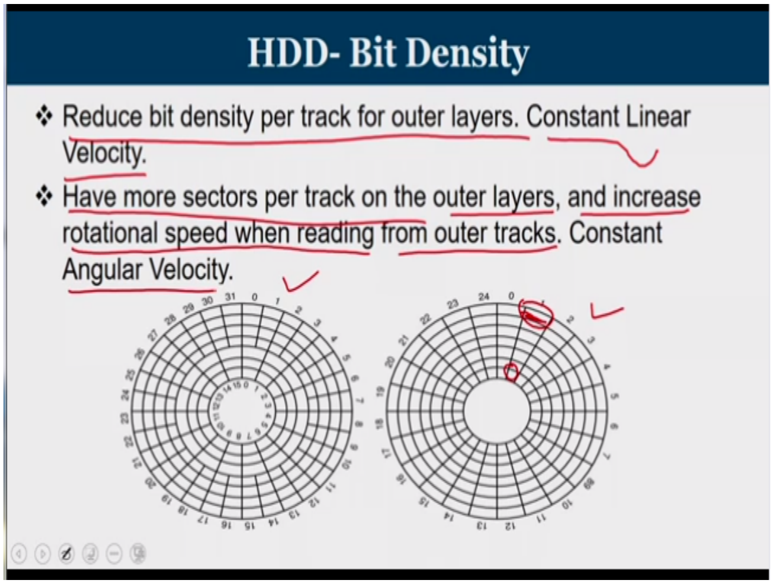
**(Refer Slide Time: 07:59)**



Logic storage in Hard Disk Drive

So this is how a zoomed version of you have seen this is a your surface or your platter and then you have many such tracks are available. And if you zoom into each track you can see magnetic material and wherever there is similar one then that is called as 0. And when there is a variation that happens then that is known as 1. So here it is no variation, so it is 0 here there is variation, so it is 1, here there is no variation, so it is 0, no variation that is 0. So if you look at that wherever there is a reversal that happens in the magnetic polarity that represents 1, when there is no reversal then that represents 0.

**(Refer Slide Time: 08:40)**



## HDD- Bit Density

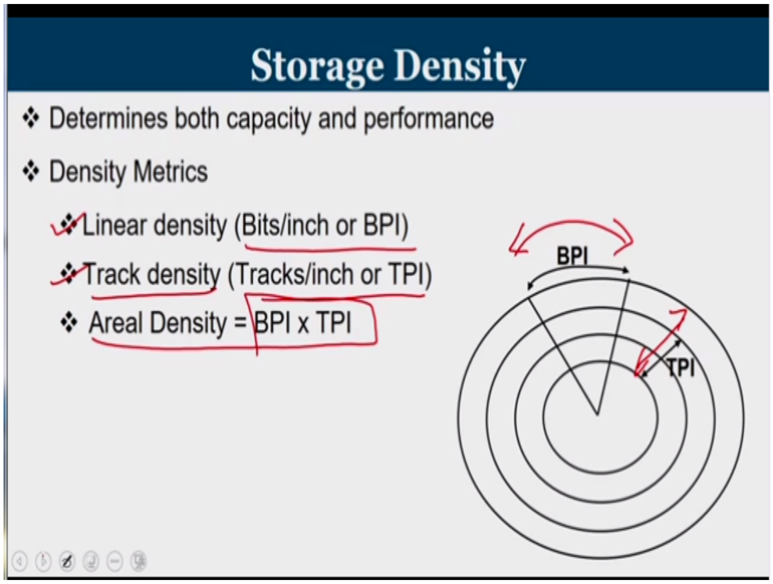
- Reduce bit density per track for outer layers. Constant Linear Velocity.
- Have more sectors per track on the outer layers, and increase rotational speed when reading from outer tracks. Constant Angular Velocity.

Now coming to the density we know that we have concentric tracks and the area in which you can store information in a track will be more in the outer tracks and the same sector will have only less area in the inner tracks. I can have 2 design, 1 that is shown on the left side where when you go to the outer tracks we can have more sectors whereas if you go to the right side design both in the inner track and in the outer track, the number of sectors per track is also same.

So we can reduce bit density per track for outer layers that is one way that is called constant linear velocity of movement that happens. Or we can have more sectors per track on the outer layers that is what you see on the left hand side diagram. And because of that we have to increase rotational speeds when reading from the outer track, that is basically constant angular velocity.

So we can go for both the approach, in the first approach that is shown on the left hand side because there are lot of area available in the outer tracks are as far as 1 sector is concern, I could further divide a sector into 2 equal subsectors. So when you read from the outer tracks you need to make the rotations more faster.

**(Refer Slide Time: 09:58)**

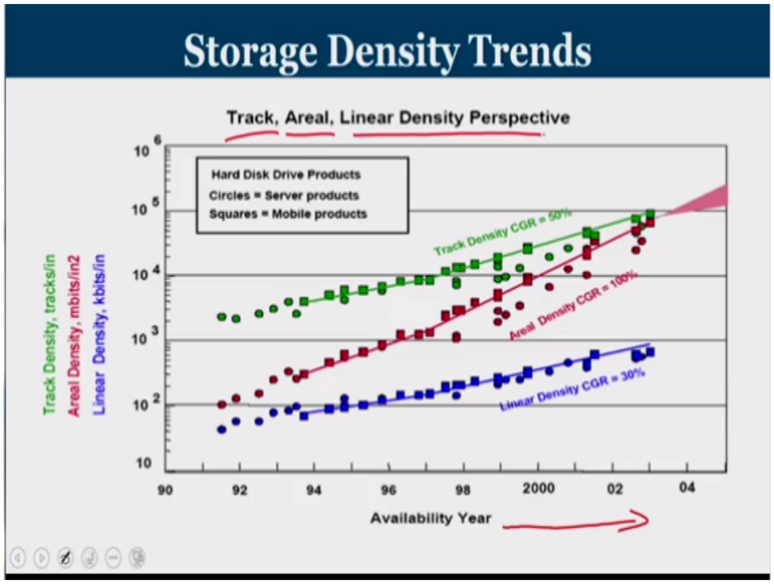


## Storage Density

- ❖ Determines both capacity and performance
- ❖ Density Metrics
  - ❖ Linear density (Bits/inch or BPI)
  - ❖ Track density (Tracks/inch or TPI)
  - ❖ Areal Density = BPI x TPI

Now coming to storage density we have something called bits per inch, number of bits that I can store per inch and the second parameter is called tracks per inch. So determines both capacity and performance and density metrics is defined by linear density that is called bits per inch or BPI. Track density that is called tracks per inch or TPI and what is the aerial density it is basically your BPI into TPI. So these are all different terminologies that are use to express the overall storage of the hard disk.

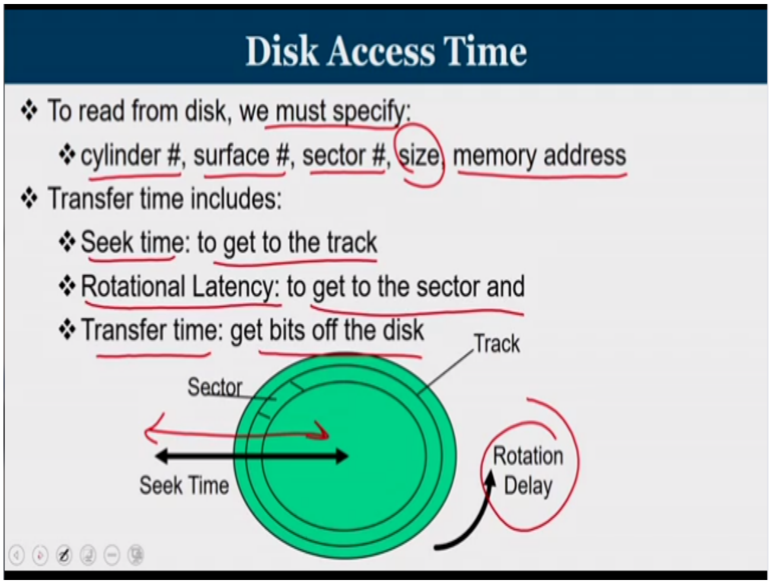
**(Refer Slide Time: 10:27)**



## Storage Density Trends

And these are all numbers which will tell you over the years whatever the hard disk that has come to the market at what is the appropriate track areal linear density perspective.

**(Refer Slide Time: 10:40)**



## Disk Access Time

- ❖ To read from disk, we must specify:
  - ❖ cylinder #, surface #, sector #, size, memory address
- ❖ Transfer time includes:
  - ❖ Seek time: to get to the track
  - ❖ Rotational Latency: to get to the sector and
  - ❖ Transfer time: get bits off the disk

Now having understood the basic concept of how a hard disk looks like, now we need to know how reading and writing happens. So you will get an address and this address has to be spitted up into cylinder number, platter number, sector number and track number. So an address let us say 200 we have to tell that this location 200 means it is cylinder number 30, platter number 2, sector number 5.

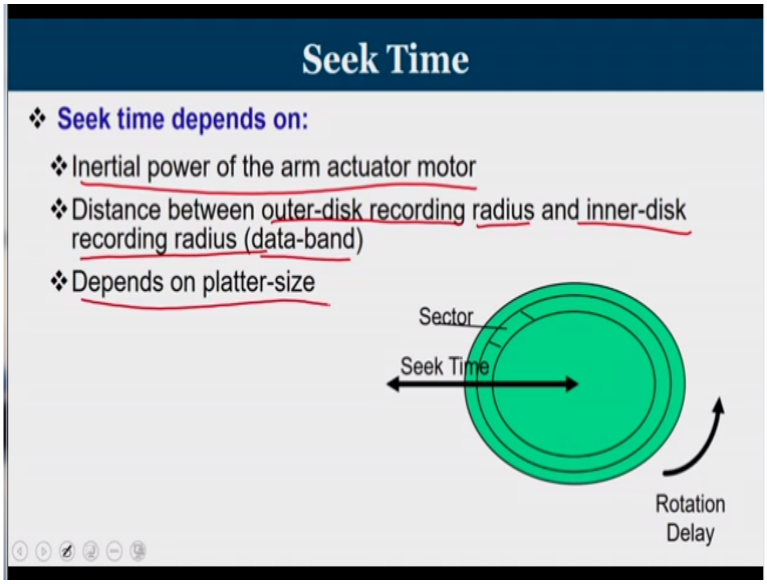
So that will be actually your address, so there is a mapping that is involved, given an address that is coming into the hard disk. The hard disk controller has to divide into smaller granularity in terms of in which cylinder this address is located and within the cylinder on which surface. And once you get a surface what is a track number and the sector number. Now once you get it how much time it will take, sometimes if the current head is in 1 cylinder and the new request is coming to a different cylinder.

Then the head has to move that means the arm assembly has to either go in or out and that is what is called seek time. And after reaching the corresponding cylinder then in order to place the corresponding sector accurately we need to have the rotation also adjusted and that is what is known as rotational latency. So it is a seek time and rotational latency that governs how much time it requires for the transfer of data from it is current head position.

So head movement is actually required that depends on what is the location in which head is there in the current context and what is the new request coming and what are the parameters associated with the new request. So to read from a disk we must specify cylinder number, surface number, sector number and what is the size of the data and what is the address that is we are talking about.

So any movement that is happening here that is called rotational delay and any movement of the arm assembly inside or outside that is what is known as seek time. So basically your transfer time includes seek time means to get into the appropriate track and rotational latency to get into the sector and transfer time to get the bits of the disk.

**(Refer Slide Time: 12:43)**

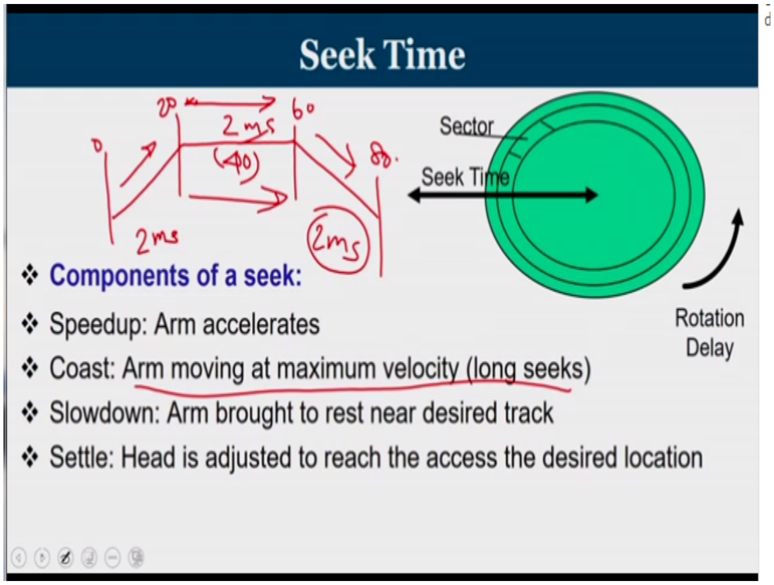


Seek Time

- Seek time depends on:
  - Inertial power of the arm actuator motor
  - Distance between outer-disk recording radius and inner-disk recording radius (data-band)
  - Depends on platter-size

Your seek time depends upon there is an inertial power of the arm actuator motor, so initially your arm is in 0 speed from it is trying to read. Now if the arm wanted to move from that particular cylinder to a different cylinder then it has a slowly start from 0 speed that is called inertial power and distance between outer disk recording radius and inner recording radius, that is called the data band. So how much is the distance between the inner most track and outer most track and what is the size of the platter.

**(Refer Slide Time: 13:16)**



## Seek Time

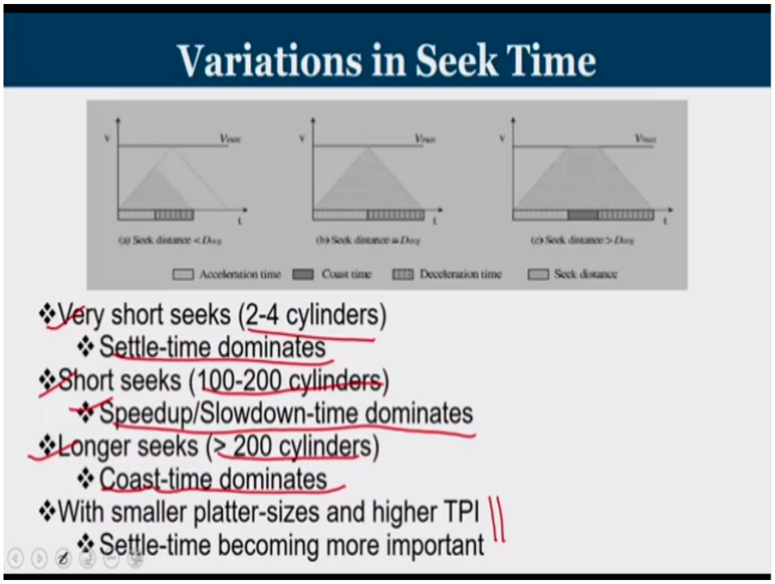
- Components of a seek:
- Speedup: Arm accelerates
- Coast: Arm moving at maximum velocity (long seeks)
- Slowdown: Arm brought to rest near desired track
- Settle: Head is adjusted to reach the access the desired location

Now we will see what are the components of a general seek, first is there is a increase in the speed and then your speed is constant for some tracks and then you are going to come down. So there is a speed up wherein your arm is going to accelerate and then there is cost the arm is moving at maximum velocity that is basically for long seeks. And then we have a slow down your arm brought to rest near the desired track.

So think of a case that it requires let us say 2 millisecond to move from track number 0 to track number 20. And it requires 2 millisecond to slow down from track number 60 to track number 80 and the movement from 20 to 60 will be at a higher rate. And the whole thing can be completed maybe in another 2 milliseconds. So here this 40 tracks I can cover in 20 millisecond whereas initial 20 tracks I will take the same 2 millisecond and the last 20 tracks also I can cover in 2 millisecond.

So there is an acceleration side there is a cost which is the uniform velocity, the arm moving at maximum velocity and then there is a slow down. The arm is brought to rest near the desired track and then the head is adjusted to reach the access to the corresponding desired location.

**(Refer Slide Time: 14:42)**



Variations in Seek Time

(a) Seek distance < $D_{avg}$ (b) Seek distance = $D_{avg}$ (c) Seek distance > $D_{avg}$

- ❖Very short seeks (2-4 cylinders)
  - ❖Settle-time dominates
- ❖Short seeks (100-200 cylinders)
  - ❖Speedup/Slowdown-time dominates
- ❖Longer seeks (> 200 cylinders)
  - ❖Coast-time dominates
- ❖With smaller platter-sizes and higher TPI
  - ❖Settle-time becoming more important

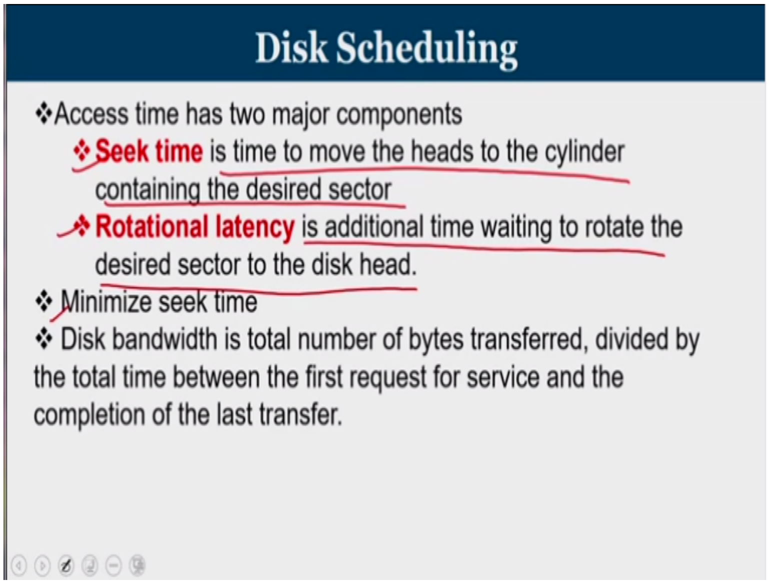
So variation in seek time, sometimes when you have a very short seek that means the current position and the desired location is very close to each other that is within 2 to 4 cylinders it is settle time that is going to dominate. For short seeks coming up to 100 to 200 cylinders the speed up and slow down time is going to dominate. And for longer seek that is more than 200 cylinders you have a speed up then you have longer amount of cost time and then you are going to slow down, So in that case of cost time dominates.

With smaller platter sizes and higher tracks per inch settle time become more important. Because of tracks are been closely associated, so in order to locate the corresponding cylinder number with high precision your settle time will take little bit of more time. So the whole concept of these acceleration cost and deceleration and settling can be defined as the case of an elevator.

So consider the case that you have a 100 storey building wherein you are going to start from the ground floor all the way to the 100th floor. Now it may take 4 or 5 seconds initially to gather the speed let us say you took 3 seconds to reach from floor 0 to floor 5. From floor 5 all the way up to floor number 95 you maybe going at a maximum speed and the last 5 floors you maybe again coming down.

So moving from floor number 5 all the way to floor number 95 at a constant speed that is what is known as cost. So it is very much important to understand the concepts and a correlation between these terms.

**(Refer Slide Time: 16:23)**



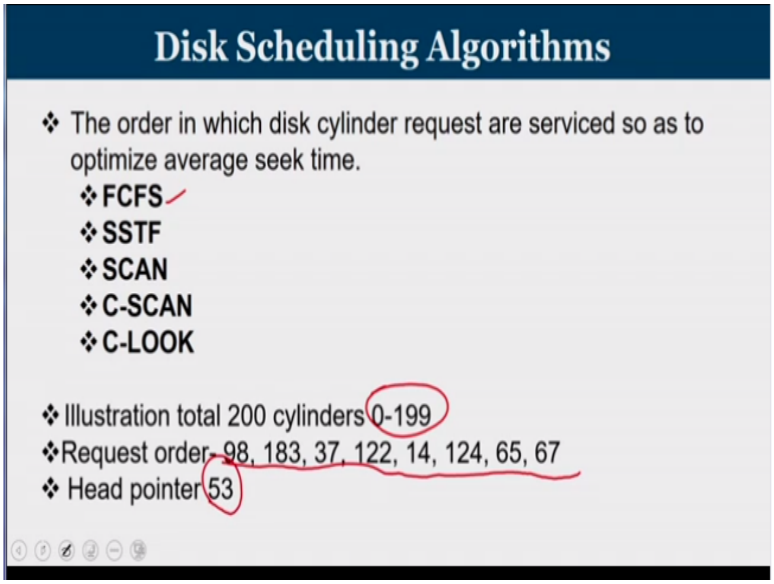
## Disk Scheduling

- Access time has two major components
  - **Seek time** is time to move the heads to the cylinder containing the desired sector
  - **Rotational latency** is additional time waiting to rotate the desired sector to the disk head.
- Minimize seek time
- Disk bandwidth is total number of bytes transferred, divided by the total time between the first request for service and the completion of the last transfer.

Now coming to another aspect which is called disk scheduling, so the access time has 2 major components that we discuss one is called a seek time it is a time to move the head to the cylinder containing the desired sector. And second one is called rotational latency it is a additional time waiting to rotate the desired sector to the disk and we are going to minimize the seek time that is our job.

So the disk bandwidth is total number of bytes transferred divided by total time between the first request for service and the completion of the last transfer.

**(Refer Slide Time: 16:55)**



## Disk Scheduling Algorithms

- ❖ The order in which disk cylinder request are serviced so as to optimize average seek time.
  - ❖ FCFS
  - ❖ SSTF
  - ❖ SCAN
  - ❖ C-SCAN
  - ❖ C-LOOK
- ❖ Illustration total 200 cylinders 0-199
- ❖ Request order- 98, 183, 37, 122, 14, 124, 65, 67
- ❖ Head pointer 53

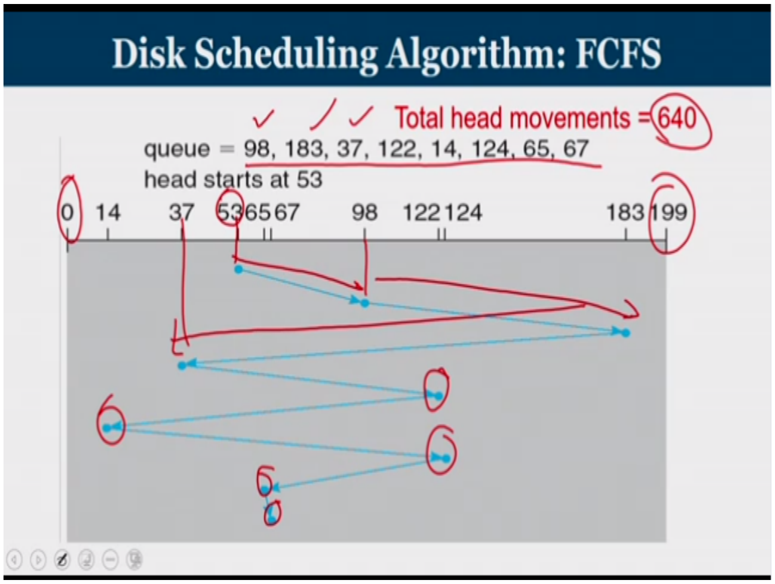
Let us now try to work with disk scheduling algorithms, now what is the context of disk scheduling algorithms. Your main memory is going to give you lot of requests to the disk scheduler such that it wants the data to be copied from the hard disk all the way to main memory. The request can be from processor side, request can be from DMA controllers, from I/O processors like that whatever it is at the end the hard disk controller have a set of request.

And this request has to be converted to cylinder numbers, platter numbers, sector numbers. Now once you have many request which one will I pick that is called disk scheduling. Like a previously mentioned DRAM scheduling means you got lot of request which one to schedule similarly disk scheduling means there are lot of request that is coming from that I have to pick one to service after servicing that go to the next one.

So once you have a pool of request that is been waiting depending upon different quality metrics we have different kind of choosing of these request. Let us try to understand each one of them one by one. So the first one is FCFS scheduling, for the rest of the illustration you just imagine that you are going to talk with a hard disk system which consist of 200 cylinders 0 to 199 and the order in which the request came is 98, 183, 37, 122, 14, 124, 65 and 67.

These are the cylinder numbers of the request that has come and the request of come in this order meaning 98 the request to cylinder number 98 is come first followed by 183 followed by 37 like that. So all of these are now currently inside the queue and assume that the head is at 53.

**(Refer Slide Time: 18:40)**



## Disk Scheduling Algorithm: FCFS

Total head movements = 640

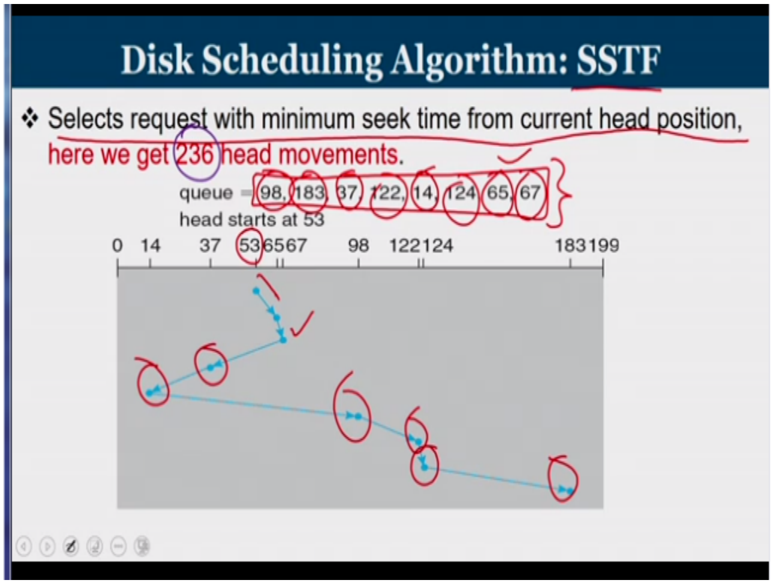
queue = 98, 183, 37, 122, 14, 124, 65, 67

head starts at 53

Now if you go to FCFS scheduling first come first out scheduling, you know that this is the queue currently my head is at 53. Now if you look at what is a very next one 98, so from 53 the head has to move to 98. Now you will see that very next one that is there in the queue is 183, so from 98 the head will move to 183. Then it goes to 37, so it come back so it is 37 and then it is 122, so it goes all the way to 122 then it goes to 14 then it goes to 124, it goes to 65 and then 67.

So whatever you see these are the cylinder numbers ranging from 0 all the way up to 199 and the way in which the head is moving. So to satisfy all this request in the given context it requires 640 head movements. So if the seek time is proportional to the number of head movements then we have 640 head movements that is there. So if you look at overall the summary of first come first out scheduling is it is not going to give you the best of performance, it is morely like zigzag movement.

**(Refer Slide Time: 19:52)**



## Disk Scheduling Algorithm: SSTF

- Selects request with minimum seek time from current head position, here we get 236 head movements.

queue = 98, 183, 37, 122, 14, 124, 65, 67

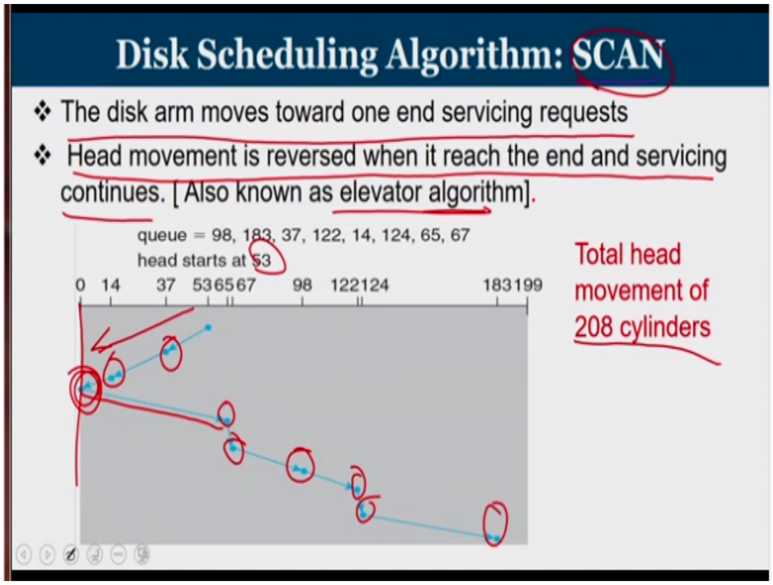
head starts at 53

Let us now try to see another algorithm which is called shortest seek time first, so here it selects request with minimum seek time from the current head position. So you have these are the request that are already there in the queue and we know that currently my head is at 53. Now look into this queue find out which is the most closest to 53 and we know that 65 is the most closest.

So even though 65 as come a little late or 98 has come very early I am not going to process 98, I am going to move my head from 53 all the way to 65. Now once you are in 65 you see that the rest of the request 67 is the closest, so I move to 67. Now if you look at there is 124 that is there, so you need to find out how many number of head movements are required to move from 67 to 124 then there is something called 14 there is 37.

So 37 seems to be the closest one to 67, so the next request is to 37 and from 37, 14 is the closest one, so move to 14. And then you move to 98 and then from there to 122, 124 that is what is been shown and at the end where which in 183. So in this case I have little lesser number of head movements only 236 movements are there because at every time I am trying to look into who is my nearest point with respect to the current head location.

**(Refer Slide Time: 21:21)**



## Disk Scheduling Algorithm: SCAN

- The disk arm moves toward one end servicing requests
- Head movement is reversed when it reach the end and servicing continues. [Also known as elevator algorithm].

Total head movement of 208 cylinders

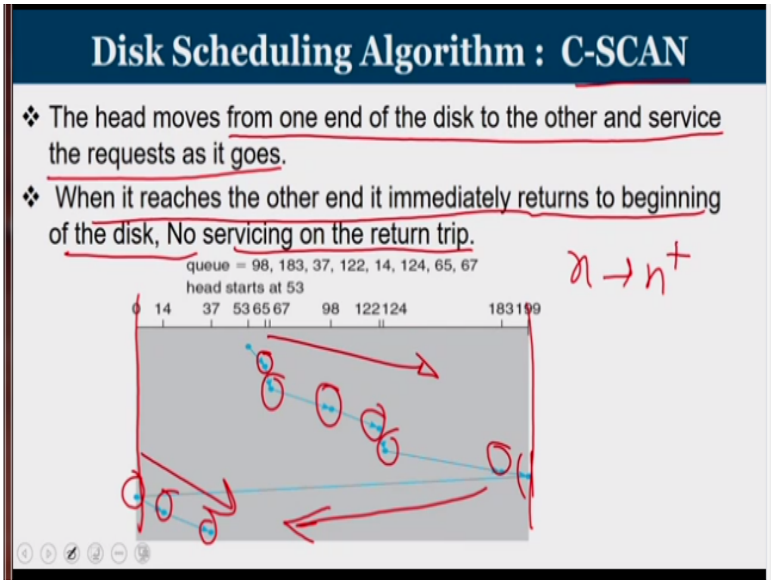
Moving further it is called scan algorithm, so you know that in the first algorithm FCFS scheduling as well as SS shortest seek time fist scheduling. The head has to move forward and backward depending upon which is the closest one and to understand what do you mean by the head going a U-turn it has to come down to 0 speed and then take a reverse direction and that is going to consume lot of power, it is a slow process as well.

So can you reduce the number of U-turns, can you travel in one direction try to satisfy all the request in that direction and then take a reverse direction and try to satisfy them. That is exactly what this scan algorithm is, so disk scheduling algorithm using scan the disk arm moves towards one end serving the request. Head movement is reversed when it reaches the end and then serving continues this is also known as elevator algorithm.

So you know that the head is at 53, let us imagine that from 53 I am moving to inner most tracks that is going towards lower cylinder numbers. So from 53 start servicing in this direction, in the meantime service 37, service 14 with that we are done but the head will move and hit this end. And then it takes a reverse direction when it takes a reverse direction there is nobody from 0 all the way to 53 which is to be serviced.

Because everything between 0 and 53 is already service, so post 53 we have 65, 67 then 98, 122, 124 and 183. So there is only this is the only U-turn that you have and the total number of head movements is reduced to 208.

**(Refer Slide Time: 23:06)**



## Disk Scheduling Algorithm : C-SCAN

- The head moves from one end of the disk to the other and service the requests as it goes.
- When it reaches the other end it immediately returns to beginning of the disk, No servicing on the return trip.

queue = 98, 183, 37, 122, 14, 124, 65, 67

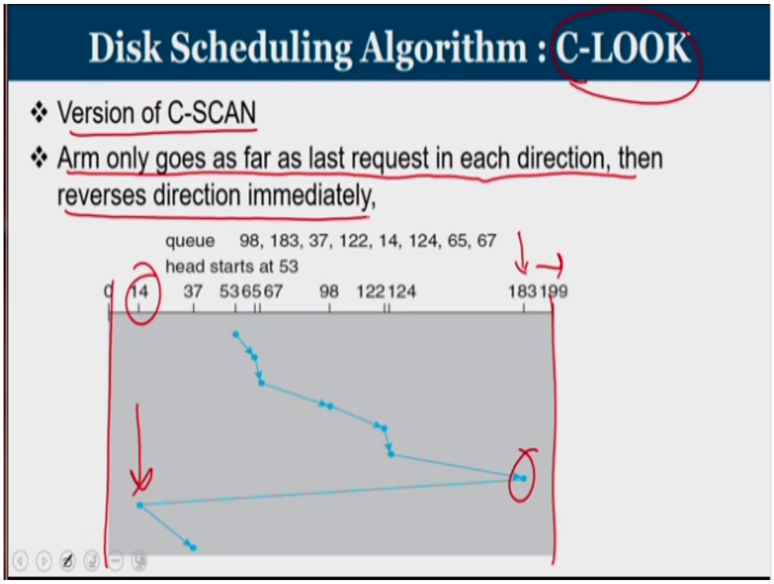
head starts at 53

$n \rightarrow n^+$

The other one is called circular scan the head moves from one end to disk to another and service request as it goes when it reaches the other end it immediately returns to the beginning of the disk, no servicing on the return trip. So basically when it moves from lower cylinder numbers to higher cylinder numbers means n to n+ during that time only I am going to service. So I am starting from 53 going to 65 and 67 they are larger numbers I will service 65 and 67, 98, 122, 124, 183.

Then I am going and hitting this end, once I hit the end you take a full retrace come back to 0 that means on return trip I am not servicing, I have always service when I moving from inner tracks to outer tracks. So this 14 and 37 will get serviced during this stage, so here you have 2 U-turns at basically happens but during this direction movement where is no servicing, so I can move in full cost.

**(Refer Slide Time: 24:10)**



## Disk Scheduling Algorithm : C-LOOK

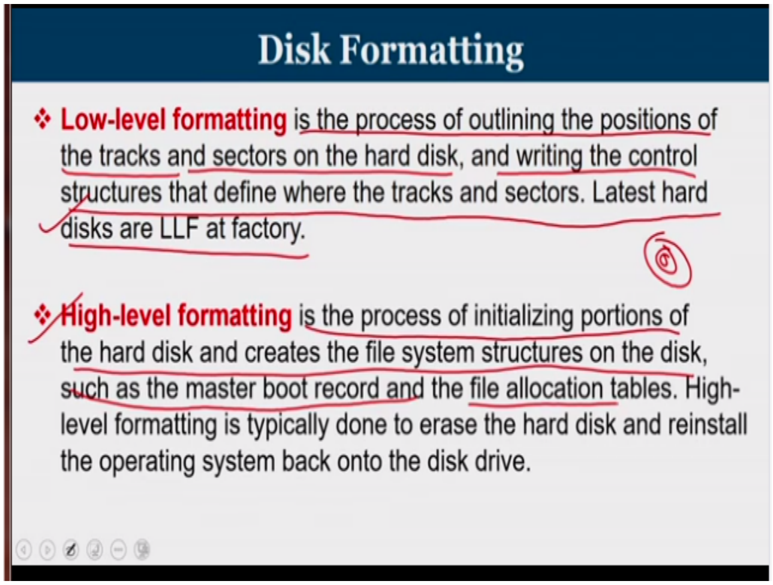
- ❖ Version of C-SCAN
- ❖ Arm only goes as far as last request in each direction, then reverses direction immediately,

queue 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53

One of the drawback in this case is you need to touch the extremes, a slight modification of that is called the C-look algorithm where once at every stage you see whether there are anymore request that is there in the same direction. See if you look at the graph it is clear rather than moving from 183 all the way to 199 and stop at 183 and come back to the other extreme 14. So I am not going to touch on the extremes just move only to locations where the servicing is required.

So this is called a solidified version of scan which is known as C-look, arm only goes as far as the last request in each direction, then it reverses the direction immediately. So we learned about FCFS scheduling first come first serve it has lot of zigzag movements and then it is shortest seek time first. Then third it is a scan algorithm and then we have the C-scan that is called circular scan and then which is called the C-look algorithm.

**(Refer Slide Time: 25:14)**



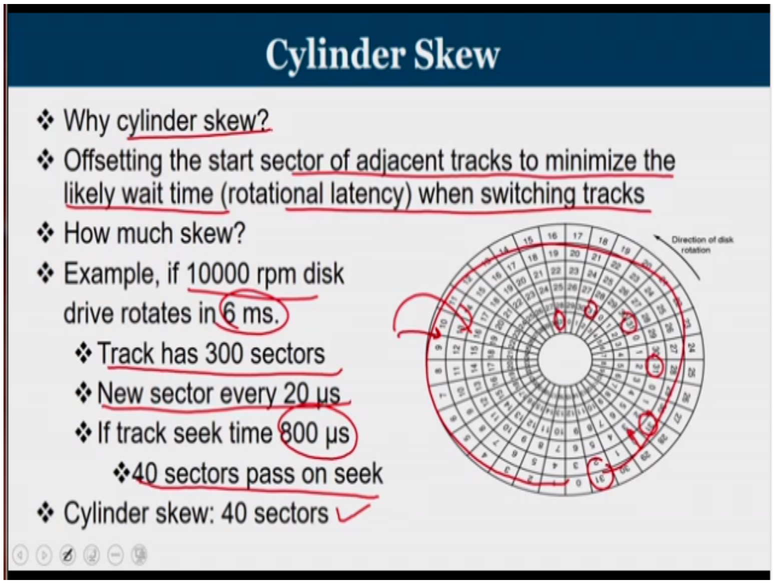
## Disk Formatting

- **Low-level formatting** is the process of outlining the positions of the tracks and sectors on the hard disk, and writing the control structures that define where the tracks and sectors. Latest hard disks are LLF at factory.
- **High-level formatting** is the process of initializing portions of the hard disk and creates the file system structures on the disk, such as the master boot record and the file allocation tables. High-level formatting is typically done to erase the hard disk and reinstall the operating system back onto the disk drive.

Now coming to formatting concepts in the disk there are 2 types of formatting as far as disk is concerned. The first is called low level formatting, it is the process of outlining the positions of the tracks and sectors on the hard disk. And writing the control structures that define where tracks and sectors start and end and these are typically done at the factory level. So the process of creating this tracks and sectors and that is been done only once generally at the factory.

And what we do in terms of a general formatting that we talk in our colloquial language is called high level formatting. It is a process of initializing portions of the hard disk and create the file system structures on the hard disk. Such that the master boot record and the file allocation tables are all kept, high level formatting is typically done to erase the hard disk and reinstall operating system back on to the disk drive.

**(Refer Slide Time: 26:08)**



## Cylinder Skew

- Why cylinder skew?
- Offsetting the start sector of adjacent tracks to minimize the likely wait time (rotational latency) when switching tracks
- How much skew?
- Example, if 10000 rpm disk drive rotates in 6 ms.
  - Track has 300 sectors
  - New sector every 20 µs
  - If track seek time 800 µs
    - 40 sectors pass on seek
- Cylinder skew: 40 sectors

Next topic is cylinder skew, offsetting this start sector of an adjacent track to minimize likely waiting time of rotation latency when you switch across tracks. So think of a case that we have to continuously read the data let us say my data reading is getting over on 31. So I am reading from 0 all the way up to 31st track, now imagine I have only one platter so I have to switch to the next cylinder, that means the inner cylinder.

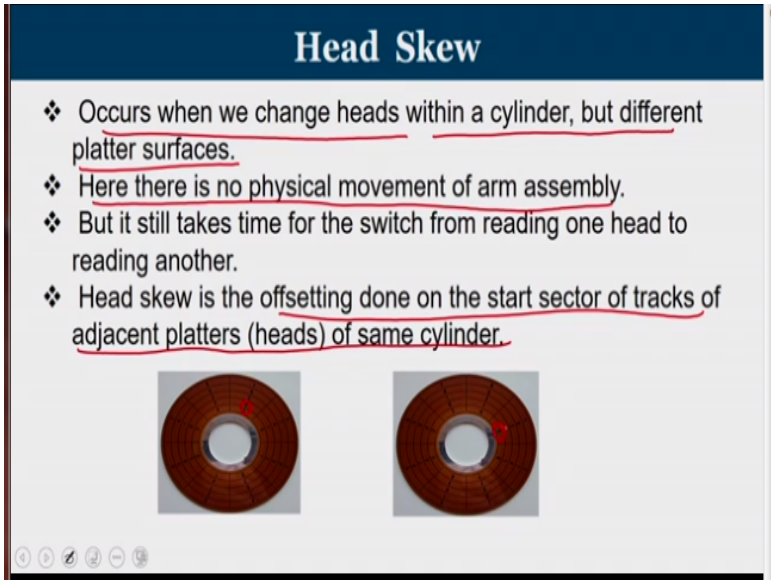
So it takes some time for the head to get itself detached from the outer track and then have a seek time to reach the next track. During that time also the whole platter is rotating, once a platter is rotating by the time it touchdown**.** So now our purpose is to start reading from 0, sector 0 of the next track by the time we reach sector 0 of the next track due to the rotational latency. It may not be 0 that is locating, there need to be a circular shift, so the 31 is here.

So if you move from inner most tracks, you can see that 31 is not in the exactly same sector. So there is a small delay, so while the rotation happens when you detach yourself from 1 track take off and come and land up in the next track. Because of the rotation we may have to align the sector numbers of adjacent tracks accordingly. This adjustment is known as cylinder skew, let us work out a problem what is been given in the slide.

Consider the case that we were talking about a 10000 rpm disk it rotates in 6 millisecond, when you have 10000 rotations per minute then it takes 6 millisecond for 1 second. Assume that your track has 300 sectors and if you take 6 millisecond to cover up all these 300 sectors then every new sector you reach in 20 microseconds. If the seek time for 1 track means moving from 1 track into the next is 800 microseconds that means during 800 microseconds 40 sectors are passed during the seek time.

So the cylinders skew is 40 sectors, that means adjacent cylinders sector numbers has to have a shift of 40 and that is what is known as cylinder skew.

**(Refer Slide Time: 28:54)**



Head Skew

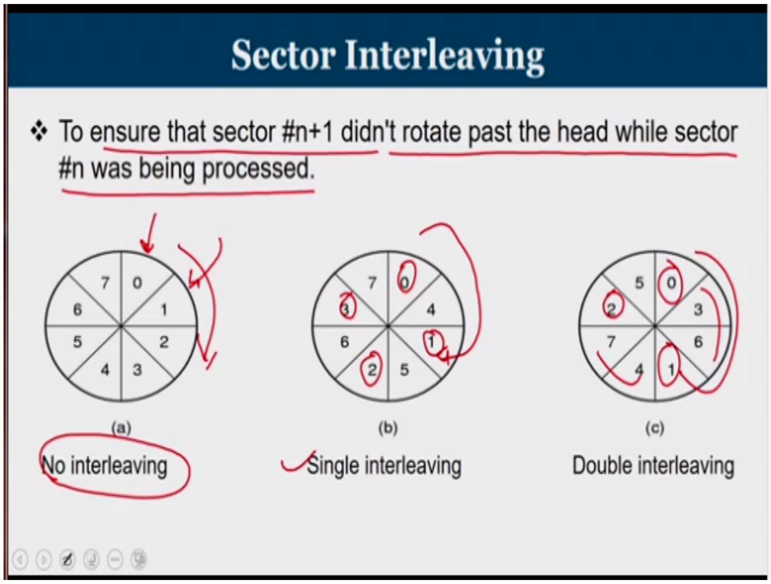
- Occurs when we change heads within a cylinder, but different platter surfaces.
- Here there is no physical movement of arm assembly.
- But it still takes time for the switch from reading one head to reading another.
- Head skew is the offsetting done on the start sector of tracks of adjacent platters (heads) of same cylinder.

There is another thing that is called head skew, it occurs when we change heads within a cylinder but different platter surfaces. Here there is no physical movement of arm assembly, so what we have is consider that you have 2 or 3 surfaces. Let us say my head is touching here, now the head wanted to read from the next platter of the same cylinder. So there is a switch removing from 1 surface and going and touching the next surface.

During the time also because of the rotation there is movement, so when you start from sector number n in 1 of the cylinder and moving and touching the sector number n of the same cylinder in a different platter. During that switching time since because of the rotation that sector number n should not be kept exactly at the same location, there should be a shift like what we have seen in cylinder skew.

So cylinder skew happens when there is a movement from one cylinder to another and head skew happens when there is a data that is to be copied which was through 1 head now. Now I am moving across platters, so head skew is the offsetting done on the start sector of tracks of adjacent platters of the same cylinder. So here you can see 5 is here whereas in this case 5 is going to be there.

**(Refer Slide Time: 30:22)**



## Sector Interleaving

- To ensure that sector #n+1 didn't rotate past the head while sector #n was being processed.

(a) No interleaving

(b) Single interleaving

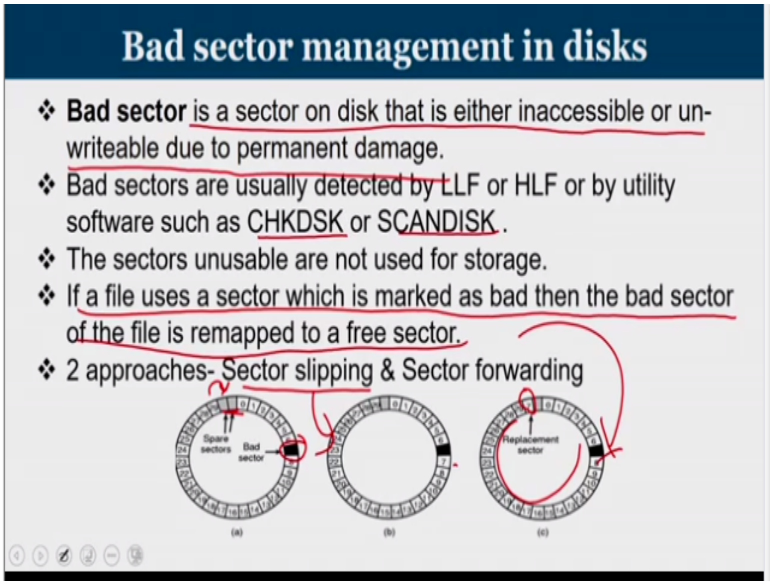
(c) Double interleaving

Now the other one is called sector interleaving, to ensure that sector number n + 1 did not rotate past the head while sector n was being processed. So you just look at this case your reading from sector number 0 and then because of this rotation by the time I process contents of 0. And then go and read 1 maybe 1 is not there it maybe 2, so this is called no interleaving process.

No interleaving process means my reading should be in such a way that by the time I complete reading and trying to go and read from the next one the next sector should be ready. So here I read 0 but by the time I process and come down the sector is actually 1 that is kept. So if there is an interleaving like this then it is known as single interleaving, I can have double interleaving also were 0, 1 and 2, the adjacent ones are been separated by 2.

So the sector interleaving ensure that one particular sector did not rotate past the head while the previous sector is been processed.

**(Refer Slide Time: 31:32)**



## Bad sector management in disks

- **Bad sector** is a sector on disk that is either inaccessible or un-writeable due to permanent damage.
- Bad sectors are usually detected by LLF or HLF or by utility software such as CHKDSK or SCANDISK .
- The sectors unusable are not used for storage.
- If a file uses a sector which is marked as bad then the bad sector of the file is remapped to a free sector.
- 2 approaches- Sector slipping & Sector forwarding

And we know that these are all magnetic equipments and we are supplying voltages in order to read and write over it. And that can be possibility that some of these sectors will be damage, so once you have certain sectors damage we call them as bad sectors. There are utility softwares which will help us to find out which are the bad sectors. Now how will be address bad sectors just because of 1 or 2 sectors of a hard disk is bad, we need not discard it.

Let us imagine there are 32 sectors inside a track but we make only 30 sectors available for the user to work on and 2 sectors are kept as buffer. So if there is a damage in any of the sectors I could potentially use this reserved 2 sectors to store the data that has suppose to store in the damage sector. There are 2 ways of doing it we will learn that, so bad sector management, bad sector is a sector on disk that is either inaccessible or un-writable due to permanent damage.

Bad sectors are usually detected by low level formatting or high level formatting or utility software such as check disks and scan disk. The sectors unusable are not used for storage, if a file uses a sector which is marked as bad then the bad sector of the file is remapped to a free sector. There are 2 approaches 1 is called sector slipping, so imagine that your sector number 7 is faulty and you have 2 extra sector in the same track they are called spare sectors.

If I copy this 29th sector into one of the empty sector such that everything will get copied. So that previously this was 8, so 7 will occupy the place of 8, 8 will occupy the place of 9. So n will occupy the place of n + 1 and the last one will occupy one of the free sector, that is technique is known as sector slipping. And here it is known as sector forwarding rather than all movement of others just 7 alone I keep in one of the empty sector and all other sectors are not disturbed then that is known as sector forwarding.

So by this mechanism we make sure that even if there is a bad block in any of the track the reserve sectors which has known as spare sectors are been used in order to store this contents. Basically 2 techniques slipping and forwarding, so that completes your quick summary of secondary storage system. So with this we are coming to the end of this storage aspect there are few tutorial sessions also which will help us to get more clarity in this concept.

A quick summary of what we learn today, apart from your conventional on-chip storage like caches and volatile main memory we need to have permanent storage also. And we are using magnetic technique based hard disk in order to permanently store data. We have seen how logically ones and zeros are been represented by same polarity and reverse polarity of magnetic materials at the (()) (34:49) zones.

We have seen about some of the organizational concept of what is arm assembly what are the subdivision of storages in terms of cylinders, platters and sectors. We have seen how reading and writing is been done and then familiarization of 2 important terms called seek time which itself will has an acceleration a cost and a slow down plus settle time. And then what is rotational latency.

Then we learned about scheduling concepts if multiple requests are coming how can you reduce seek time. There are different kind of scheduling algorithms which has it is own pros and cons and then we have few concepts on cylinder skew, head skew and sector interleaving. and we concluded our discussion with how will you manage bad sectors, thank you.