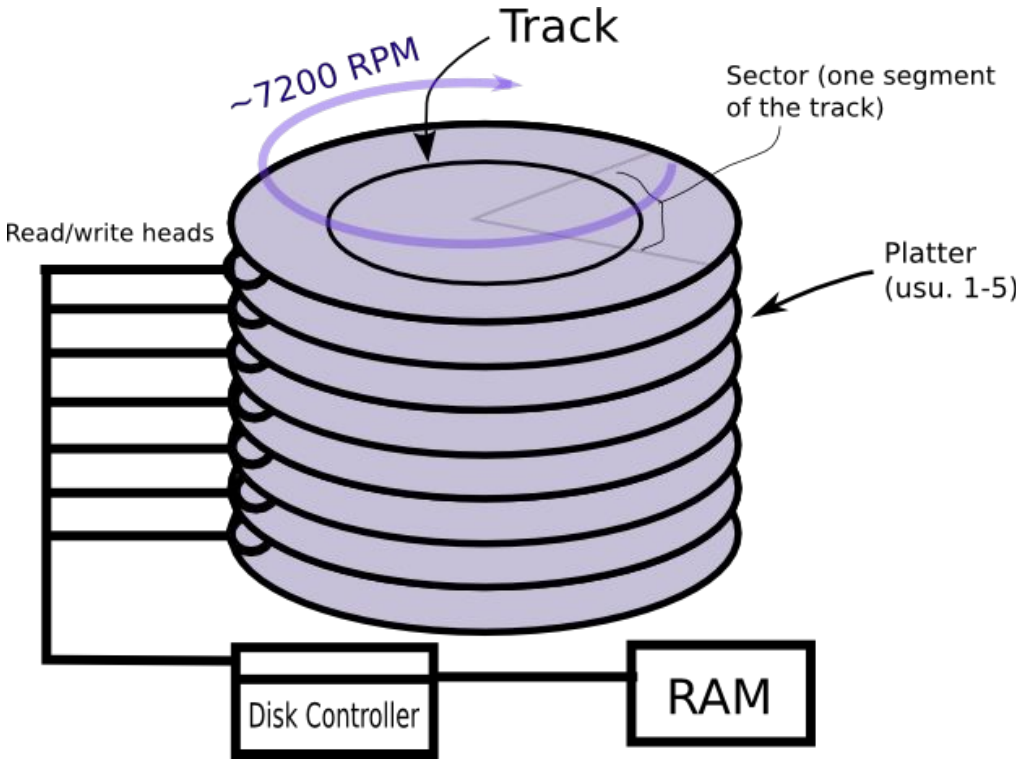
Magnetic Disk Details

Module 19

Reading

- While I have not dedicated a lecture to it, please make sure you read Chapter 5.2 and 5.3 carefully!
- This module is based on Chapter 5.4
- This module is about magnetic hard drives, not SSDs!

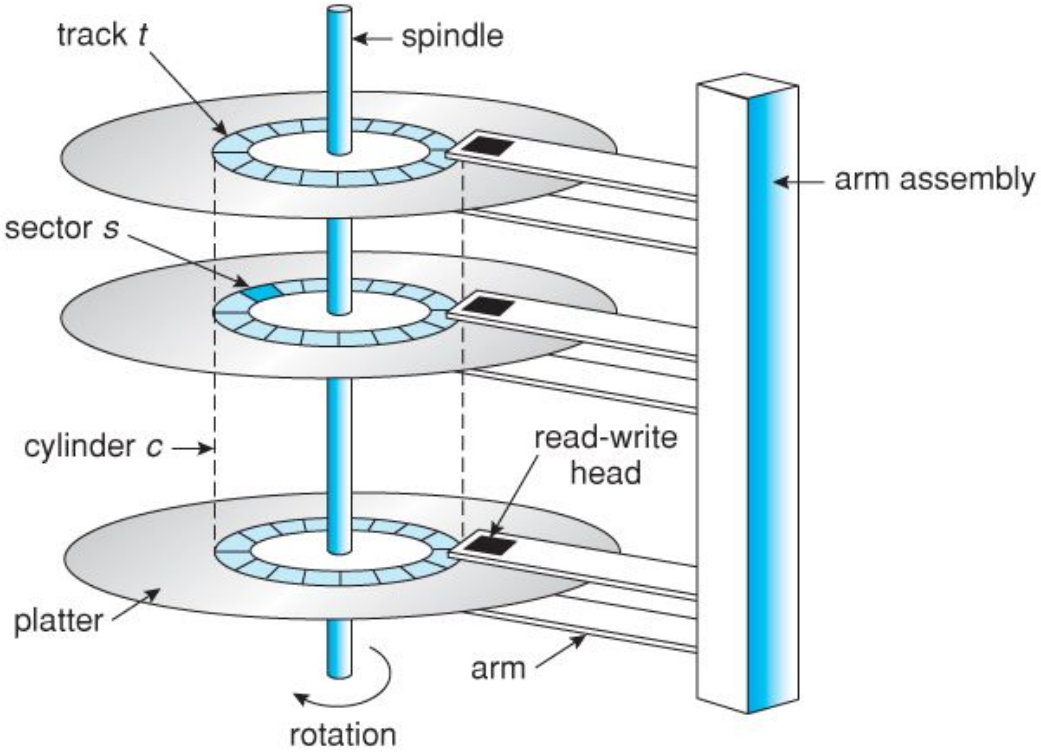
Hard Drive - Components



The Controller

- Common controller types are:
  - IDE - Integrated Drive Electronics
  - SATA - Serial ATA
- The controller is a microcontroller/small processor for doing:
  - Read/Write
  - Caching
  - Bad-Block remapping

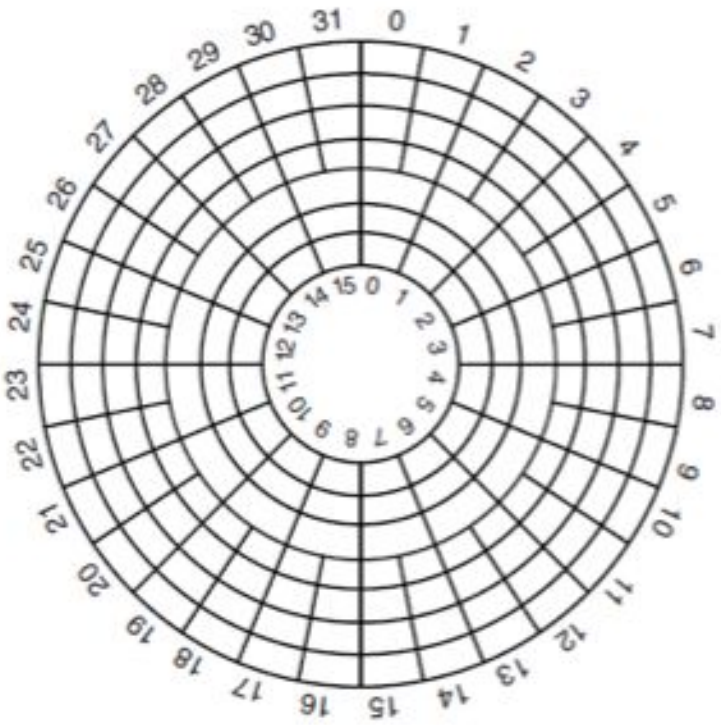
The physical device

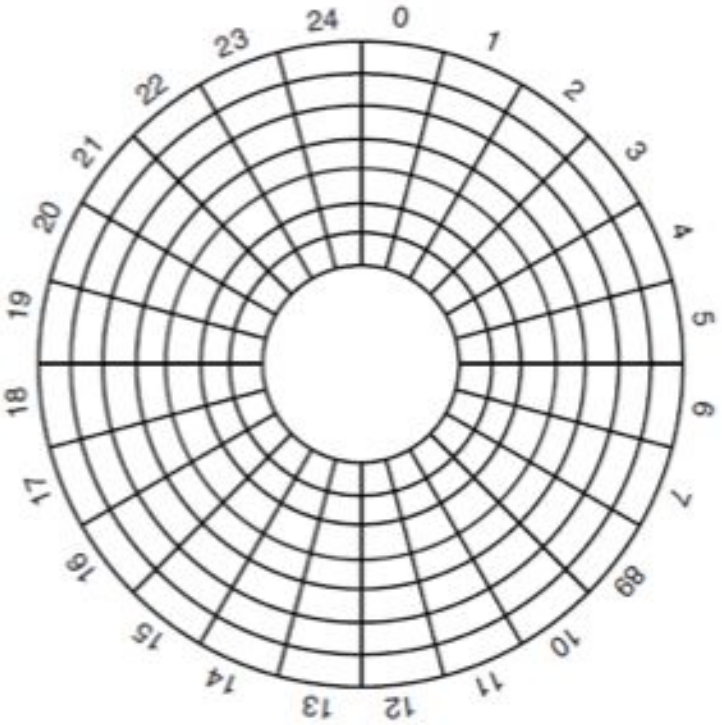


Sector 1 on Platter 1 is adjacent to Sector 1 on Platter 2, because they can be read at the same time!

Part of the job of the controller is to map block # to a track, sector, cylinder

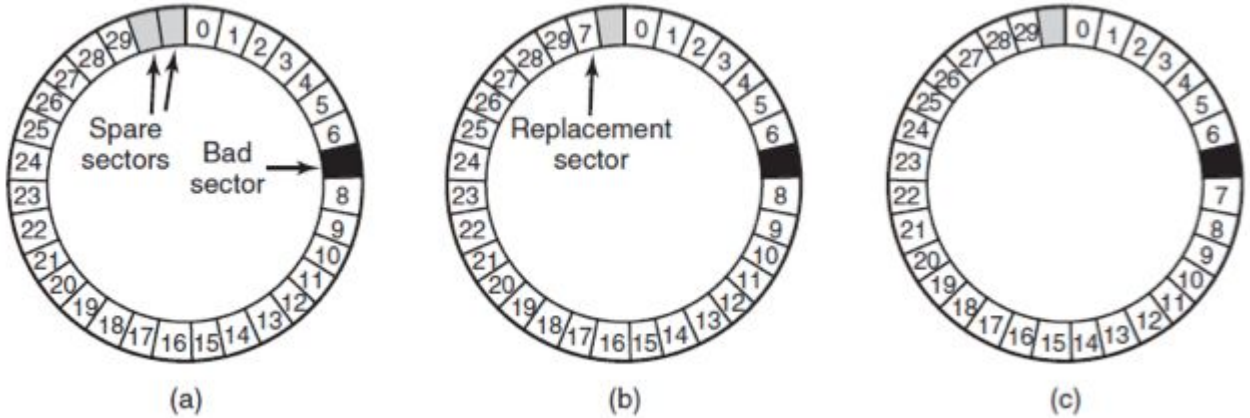
Virtual vs Physical Geometry





Bad Blocks

- We've mentioned that bad blocks can be detected by examining ECC in block footer
- A controller will reserve sectors on the same track to replace a bad sector when it arises.
- These mappings are held/performed internally to the disk controller



We can either re-assign (b) or shift (c)

Bad Blocks, Latency, and Redundancy

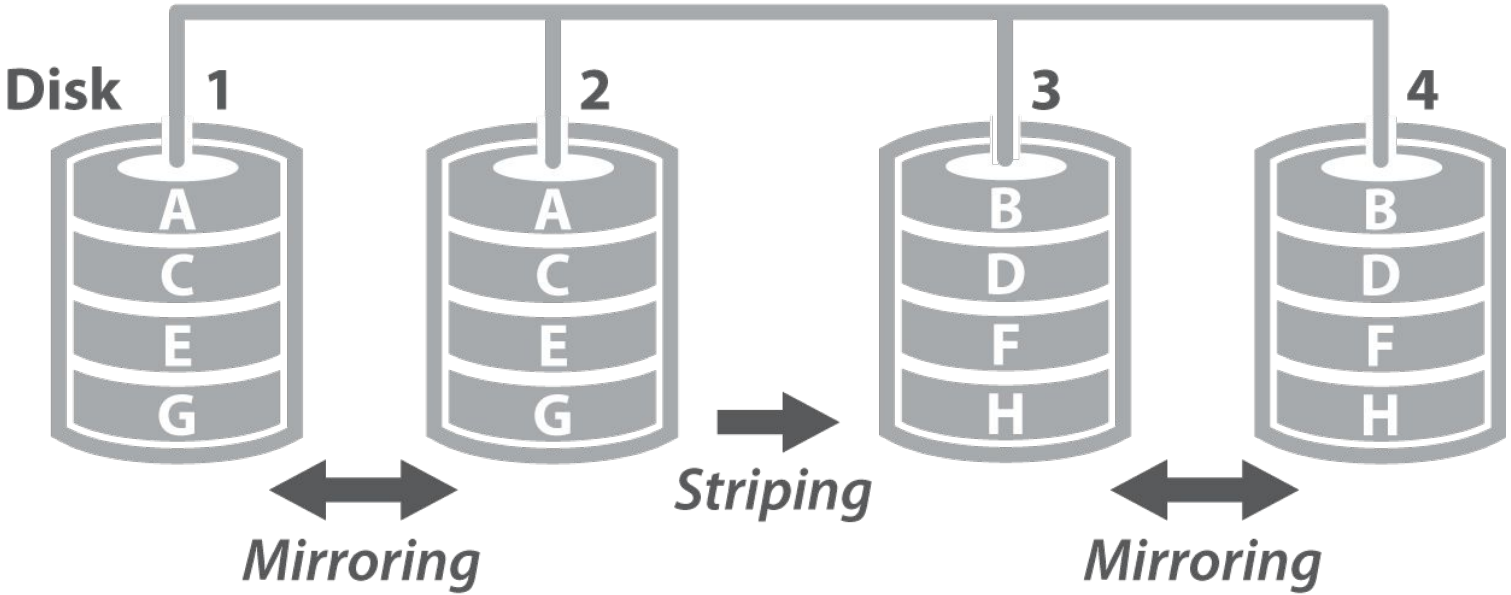
- We’ve got two essential problems with hard drives:
  - Sometimes blocks go bad
  - Writing a sector takes too much time

- **Solution:** RAID - Redundant Array of Inexpensive Disks

RAID

- Controller is far more sophisticated
  - Interface to CPU/OS is a single disk
  - Behind the controller, there are **many disks**.
- Two objectives:
  - Striping: Allow OS to issue X read/writes, which can execute in **parallel**.
  - Mirroring: Duplicate each byte on the disk on some other disk
- These objectives are met independently

RAID



Requires a minimum of four drives

Scheduling Read/Writes

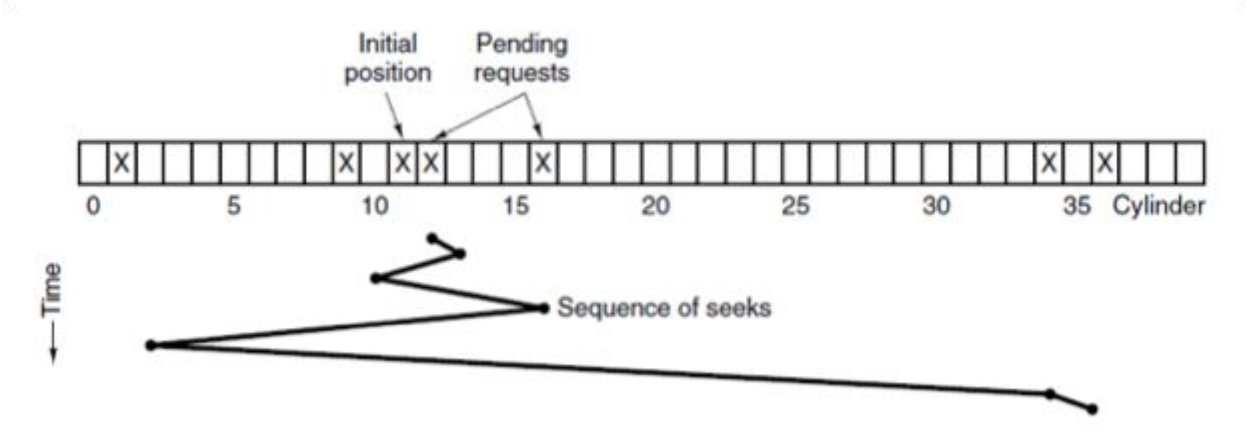
- Recall that every read/write maps to a block
- Every block maps to a platter, cylinder, sector
  - To move to a different cylinder (track), the disk head must move. This time is called **seek time**.
  - To move to a different sector, we must wait until it rotates underneath the disk head. This is called **rotational delay.**
  - There is no latency accessing different platters - they can be read simultaneously (same cylinder)
  - To read/write, we also incur **transfer time**

Scheduling Read/Write

- Transfer time is fixed
- Rotational time is variable, but uncontrollable
  - Rotational time is capped at the time it takes the disk to make a 360 degree spin
  - The disk never stops rotating, and it never turns around!
- The disk arm movement (seek time) can be controlled however - and we could be smart…
  - Rather than move the disk read/write head all over the place, we could **re-order** the requests.

Naive Approach: FCFS

As requests come to the controller, blocks are mapped to cylinders

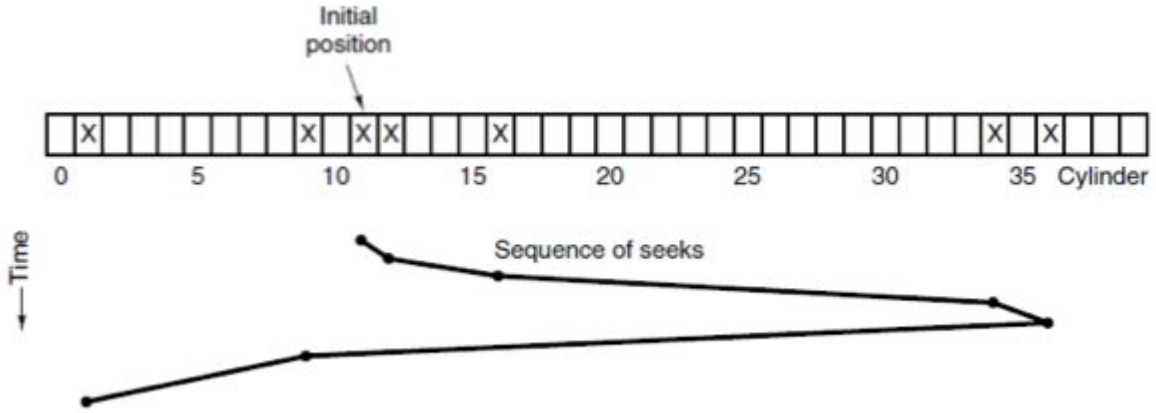


Head moves at fixed rate

Turning around also costs time (and wear)

Avoid turning around...

We can use the “elevator” algorithm - serving pending requests along the way



This will improve overall efficiency because the disk head does not turnaround as often

This is often called the **scan** algorithm as well

Fairness

- Scan algorithm has a flaw:
- Assume head is moving in positive direction
  - We pass cylinder 2
  - We won't return to cylinder 2 until we cover all cylinders above 2 **twice**.
  - The likelihood of requests pending in this location is very high now - assuming a random distribution of requests

Circular Scan

- A final optimization the controller will make is to **skip** back to cylinder 0 once it’s reached the largest cylinder

- This evens out the distribution of pending operations quite well.

Up next

- We'll examine how hardware clocks work, and how software can utilize them.

- We'll examine how the OS talks to UI devices such as keyboards and mice.