Iowa State University

Department of Electrical and Computer Engineering

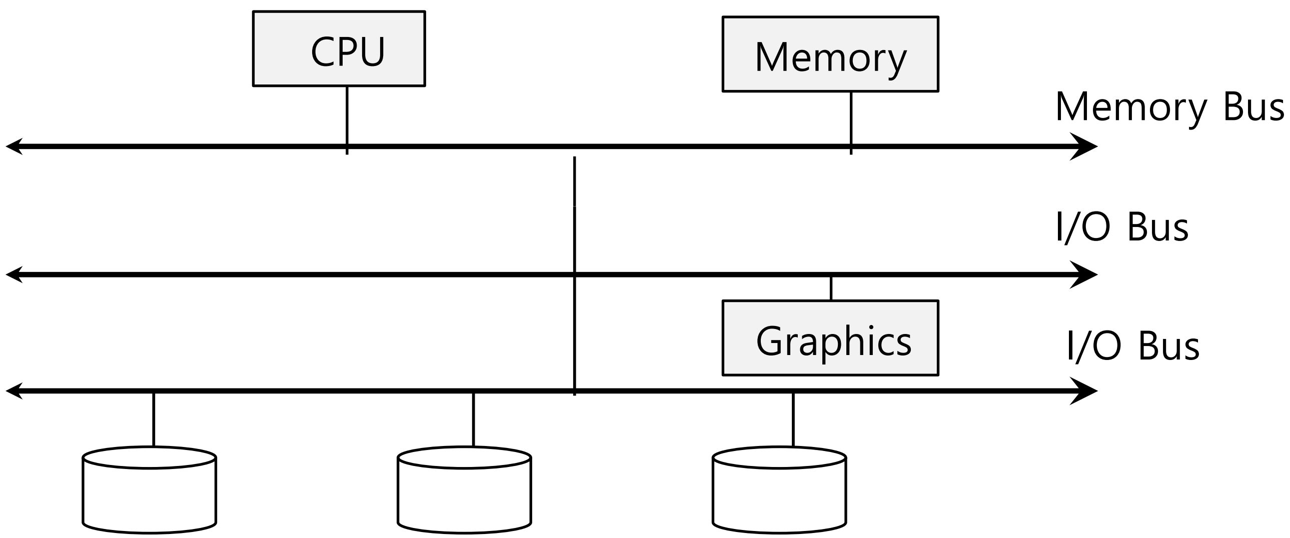
Lecture 27: Hard Disk Drives

Agenda

- Recap
- Hard Disk Drives (HDDs)
  - Pre HDD era
  - HDD Internals & I/O Time
  - I/O Scheduling

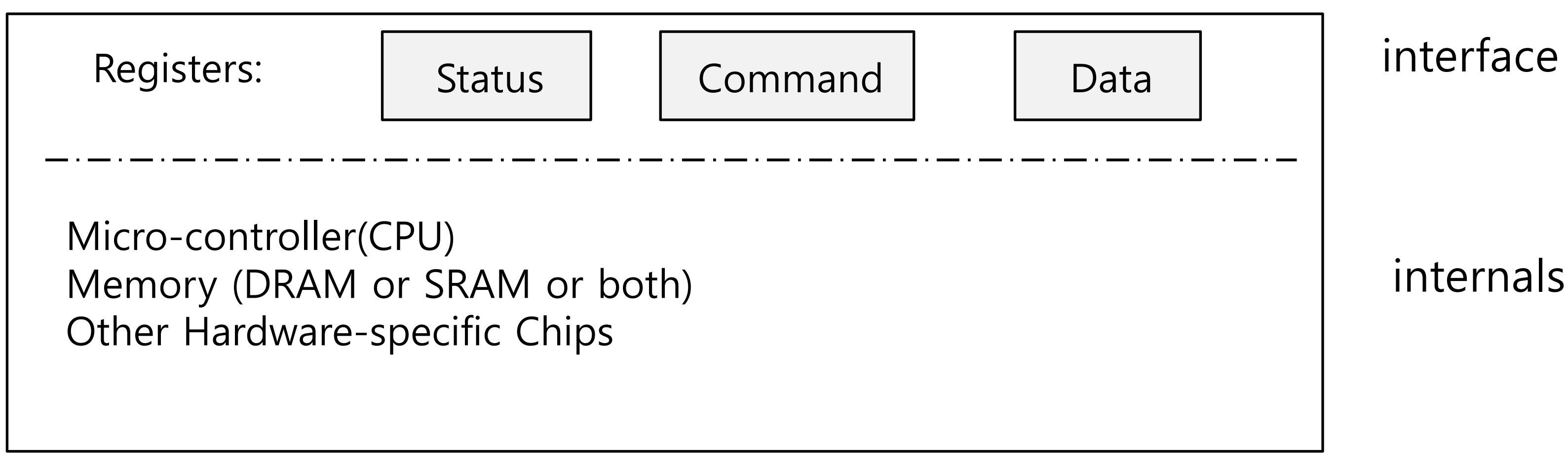
Recap

- I/O devices: critical for a computer to interact with users, the environment, and other systems
- Two basic types
  - Block devices
  - Character devices
- I/O Bus



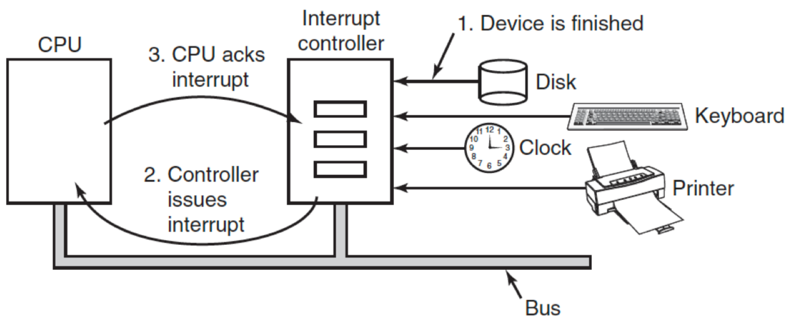
Recap

- Basic structure of I/O devices
  - **Hardware interface** allows the system software to control its operation
    - **status register**
    - **command register**
    - **data register**
  - **Internals:** implementation details invisible to host



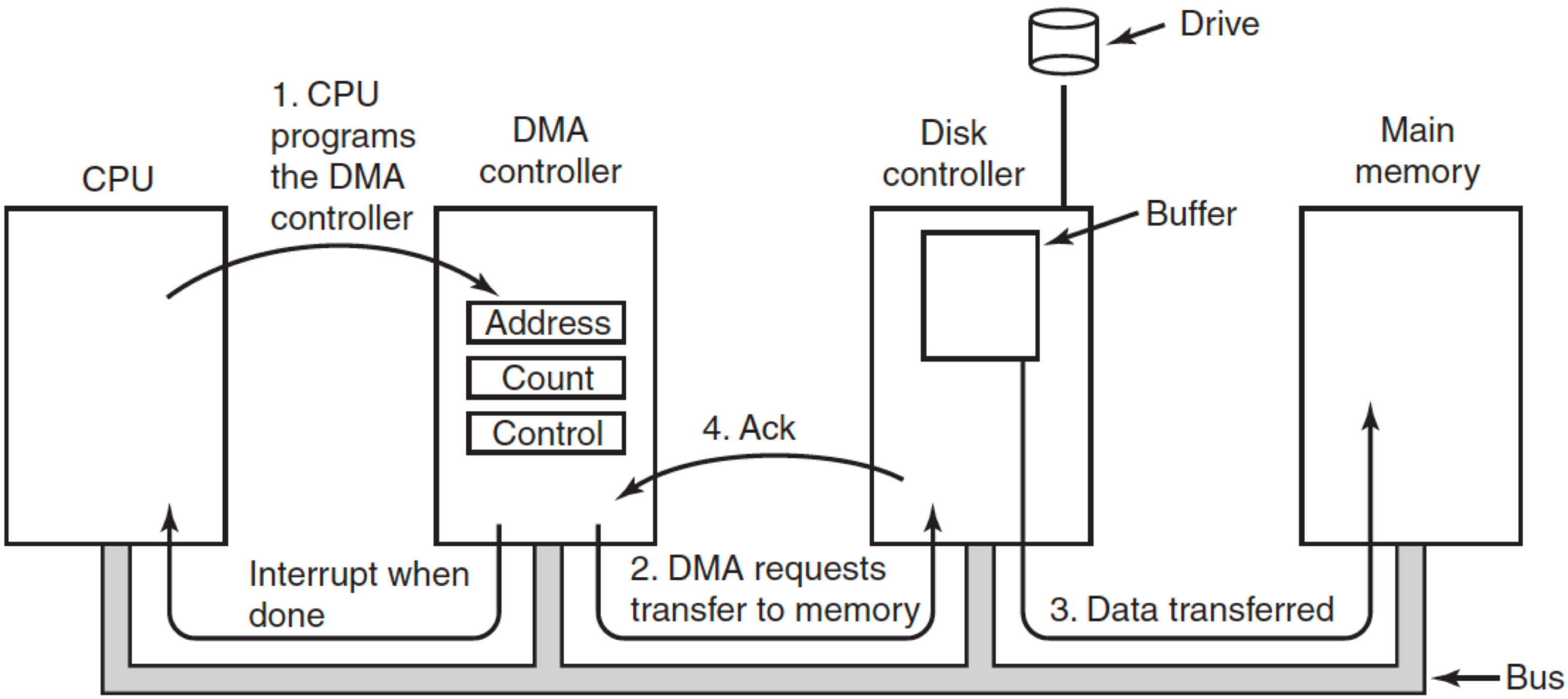
Recap

- programmed I/O
  - OS reads/writes data from/to device registers through specific I/O instructions
    - e.g., `in` and `out` instructions on x86
- memory-mapped I/O
  - The OS uses regular memory instructions to access the device
    - `load` (to read) or `store` (to write)
- Polling V.S. Interrupt



Recap

- Direct Memory Access (DMA)
  - transfer data between memory & I/O device without involving CPU



Agenda

- ~~Recap~~
- Hard Disk Drives (HDDs)
  - Pre HDD era
  - HDD Internals & I/O Time
  - I/O Scheduling

How to Store Data in Computers

- 1940s – 1970s: Punch Cards
  - computers had no memory or storage media built in
  - record programs and program data
  - card readers allow data to be reloaded into computer

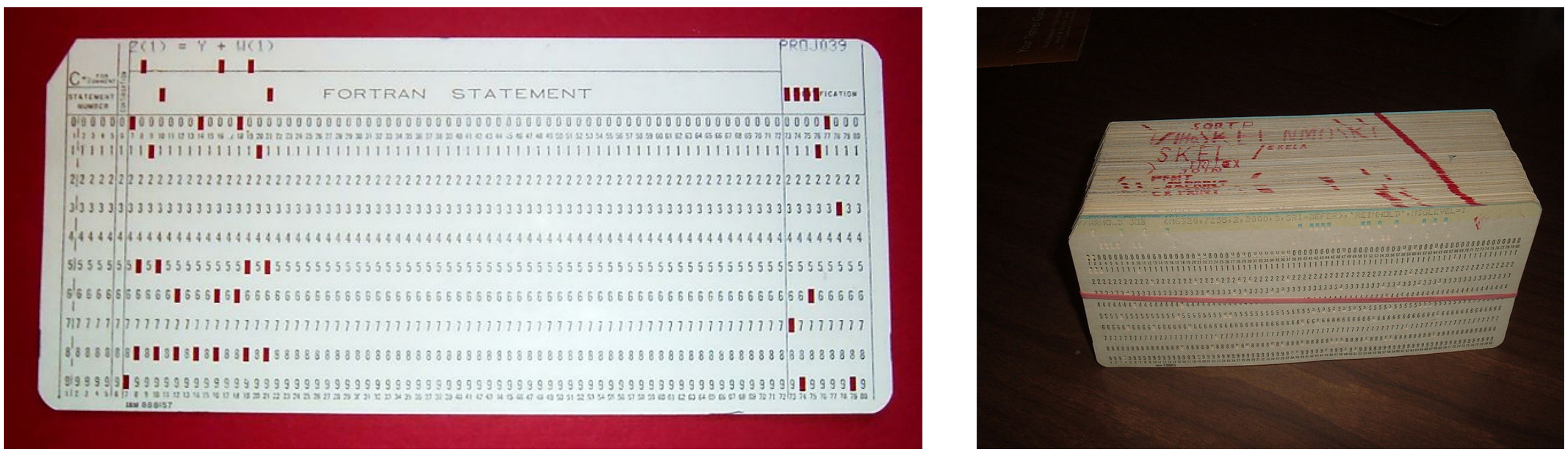


Fig. A punched card and a program

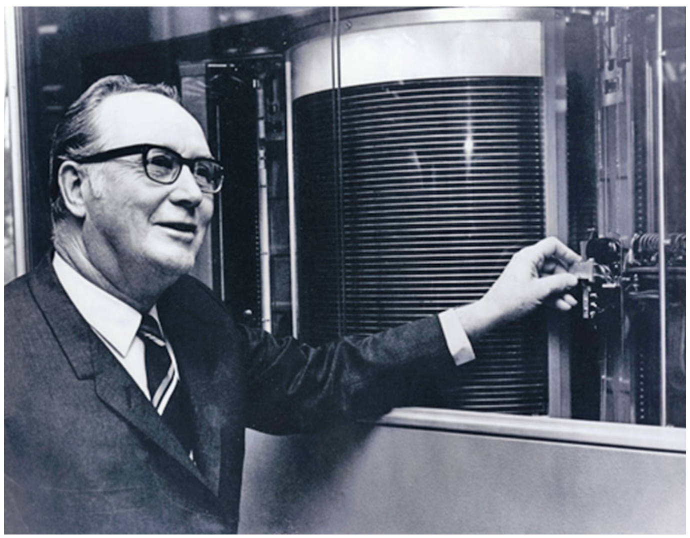
How to Store Data in Computers

- 1950s - Present: Tapes
  - paper tape -> magnetic tape
  - still in use for backup/archive

Fig. Paper tape and magnetic tape

How to Store Data in Computers

- 1956 - Present: Hard Disk Drives (HDDs)
  - initially used as memory: a random-access device, hold runtime data, while tapes were used for persistent storage



- 5 MB
- 50 platters each 24” in diameter

Fig. Rey Johnson & the 1st Hard Disk Drive

How to Store Data in Computers

- 1956 - Present: Hard Disk Drives (HDDs)
  - initially used as memory: a random-access device, hold runtime data, while tapes were used for persistent storage

- IBM RAMDAC computer with the IBM Model 350 disk storage system

How to Store Data in Computers

- 1956 - Present: Hard Disk Drives (HDDs)
  - HDD today
    - >10TB
      - Shingled magnetic recording (**SMR**)
    - < 0.85 inch
    - widely used as persistent storage, with DRAM as memory

How to Store Data in Computers

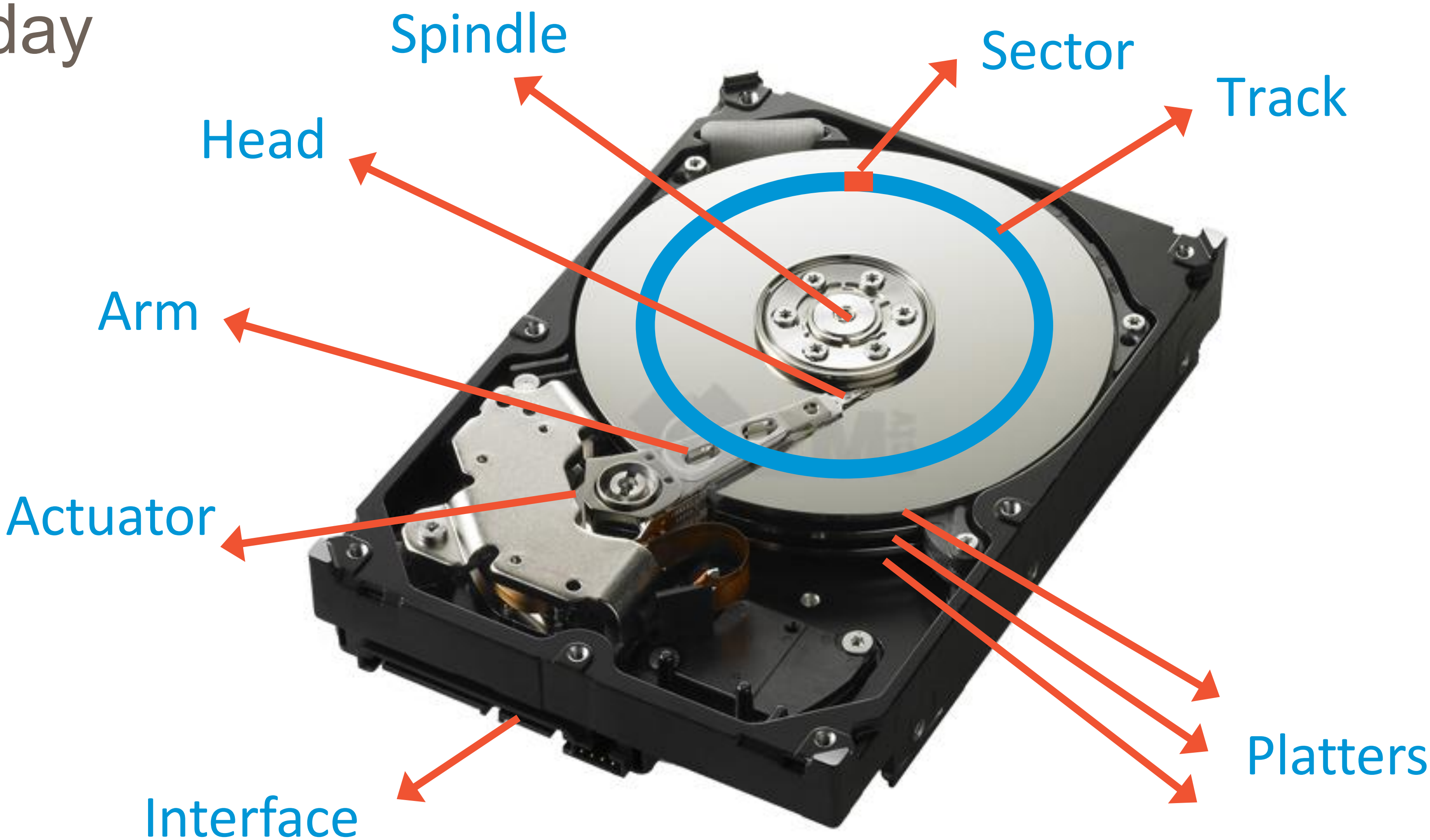
- 1956 - Present: Hard Disk Drives (HDDs)
  - HDD future
    - Heat-Assisted Magnetic Recording (HAMR)
      - potentially scale to dozens of or >100 TBs per drive

HDD Internals

- Main form of persistent data storage in computers today

HDD Internals

- Main form of persistent data storage in computers today



HDD Internals

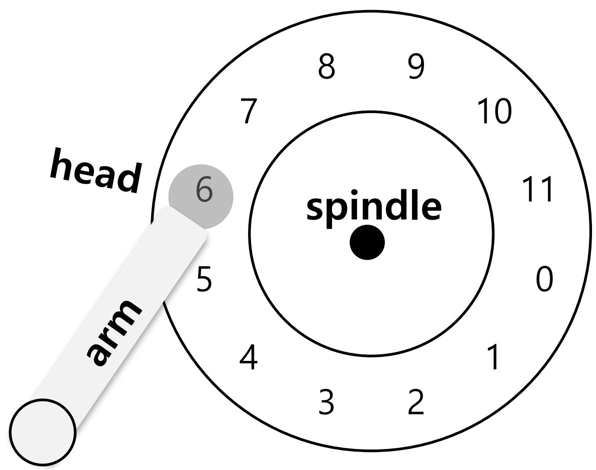
- **Sector**
  - the minimum, atomic access unit of HDD
    - traditionally 512 bytes
- **Track**
  - Concentric circles of sectors
- **Platter**
  - A circular hard plate
    - Aluminum coated with a thin magnetic layer
    - Data is stored persistently by inducing magnetic changes to it.
  - Each platter has 2 sides, each of which is called a **surface**

HDD Internals

- **Spindle**
  - Spindle is connected to a motor that spins the platters around.
  - The rate of rotations is measured in **RPM** (Rotations Per Minute).
    - Typical modern values : 7,200 RPM to 15,000 RPM.
      - E.g., 10000 RPM : A single rotation takes about 6 ms.

HDD Internals

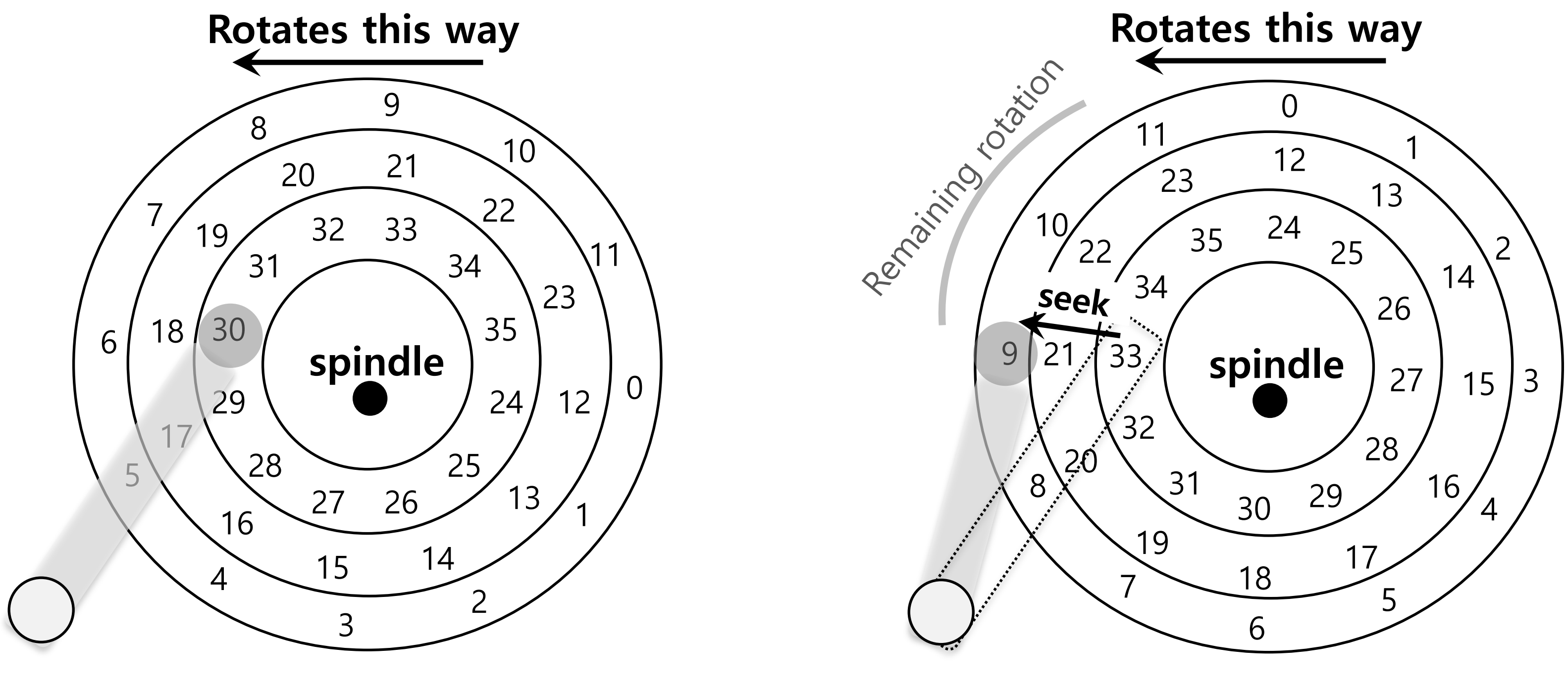
- **Disk head**
  - One head per surface of the drive
  - Perform the reading and writing on surface
  - Attached to a single disk arm, which moves across the surface.



A Single Track Plus A Head

HDD I/O Time: Seek

- **Seek**: Move the disk arm to the correct track
  - **Seek time**: Time to move head to the track containing the desired sector.
  - One of the most costly disk operations.



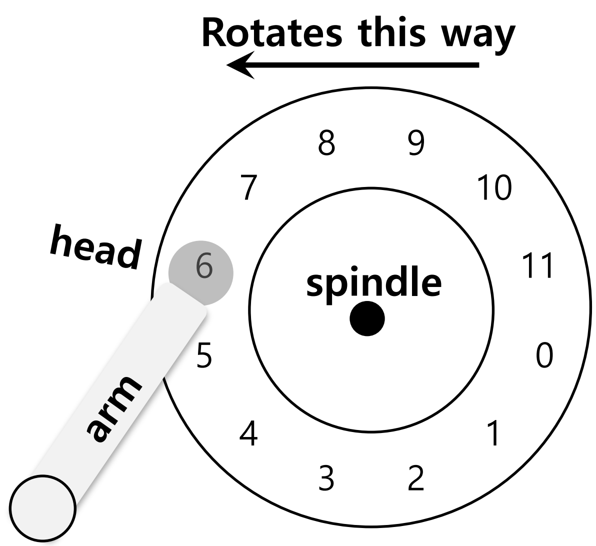
Three Tracks with A Head

HDD I/O Time: Seek

- Phases of Seek
  - Acceleration → Coasting → Deceleration → Settling
    - **Acceleration**: The disk arm gets moving.
    - **Coasting**: The arm is moving at full speed.
    - **Deceleration**: The arm slows down.
    - **Settling**: The head is *carefully positioned* over the correct track.
      - The settling time is often quite significant, e.g., 0.5 to 2ms.

HDD I/O Time: Rotational Delay

- Time for the desired sector to rotate to the head
  - e.g., full rotational delay is `R` and we start at `sector 6`
    - Read sector 0: Rotational delay = $\frac{R}{2}$
    - Read sector 5: Rotational delay = `R-1` (worst case.)



A Single Track Plus A Head

HDD I/O Time: Transfer

- The final phase of HDD I/O
  - When the head is on the right sector of the right track
  - Data is either *read from* or *written* to the surface.
- Complete HDD I/O time
  - three parts
    - **Seek**
    - Waiting for the **rotational delay**
    - **Transfer**
  - I/O time ($T_{I/O}$): $T_{I/O} = T_{seek} + T_{rotation} + T_{transfer}$
  - I/O rate ($R_{I/O}$):

$$R_{I/O} = \frac{Size_{Transfer}}{T_{I/O}}$$

Cache

- "Track Buffer"
  - Hold data read from or written to the surface
  - Allow the drive to quickly respond to requests.
  - Small amount of on-drive memory (e.g., 8 or 16 MB)
- Write on cache
  - **Writeback** (Immediate reporting)
    - Acknowledge a write has completed when it has **placed the data in on-drive memory**
  - **Write through**
    - Acknowledge a write has completed after the write has **actually been written to surface**.

Real Examples

- Specs:

	Cheetah 15K.5	**Barracuda**
Capacity	300 GB	1 TB
RPM	15,000	7,200
Average Seek	4 ms	9 ms
Max Transfer	125 MB/s	105 MB/s
Platters	4	4
Cache	16 MB	16/32 MB
Connects Via	SCSI	SATA

Real Examples

- Favor sequential I/O

		Cheetah 15K.5	Barracuda
T_{seek}		4 ms	9 ms
$T_{rotation}$		2 ms	4.2 ms
Random	$T_{transfer}$	30 us	38 us
	$T_{I/O}$	6 ms	13.2 ms
	$R_{I/O}$	0.66 MB/s	0.31 MB/s
Sequential	$T_{transfer}$	800 ms	950 ms
	$T_{I/O}$	806 ms	963.2 ms
	$R_{I/O}$	125 MB/s	105 MB/s

I/O Scheduling

- OS maintains a queue of I/O requests, and decides which I/O request to send to disk next
  - use heuristics to re-order I/O and (hopefully) minimize I/O time
    - e.g., order I/O requests based on logical block addresses (LBA)
- I/O merging
  - Reduce the number of request sent to the disk and lowers overhead
    - e.g., read blocks 33, then 8, then 34:
      - The scheduler may merge the request for blocks 33 and 34 *into a single two-block request*.

Agenda

- ~~Recap~~
- ~~Hard Disk Drives (HDDs)~~
  - ~~Pre HDD era~~
  - ~~HDD Internals & I/O Time~~
  - ~~I/O Scheduling~~

Questions?

*acknowledgement: slides include content from “Modern Operating Systems” by A. Tanenbaum, “Operating Systems Concepts” by A. Silberschatz etc., “Operating Systems: Three Easy Pieces” by R. Arpaci-Dusseau etc., and anonymous pictures from internet.