UNM SCHOOL of ENGINEERING
Department of Computer Science

Lecture #17
The Memory Hierarchy (Chap. 6)

Prof. Soraya Abad-Mota, PhD

Topics

- Chapter 6: The Memory Hierarchy
  - Storage technologies and trends (there is an article from 2019 on canvas on some developments)
  - Locality of reference
  - Caching in the memory hierarchy

The Memory Hierarchy for Performance Improvement

- Program performance Optimization continues beyond the bounds (covered on Lecture **#1**6**)** if we are aware of the memory hierarchy and its uses.
- Sec. 5.12 presents an introduction to the subtleties of the performance of load and store operations, assuming that all data are held in cache.
- Chapter 6: *The Memory Hierarchy* goes into detail about: how caches work, performance characteristics and how to make the best use of caches.

Memory and Storage Bottlenecks

- Many performance bottlenecks come from getting data to and from the processor
- Multiple levels of storage in the computer optimize the access and transfer of data:
  - On-chip (register) data storage
  - DRAM memory
  - Different forms of secondary storage (disk, SSD)
- Need to understand them to find out how to optimize for them

Random-Access Memory (RAM)

- Key features
  - RAM is traditionally packaged as a chip.
  - Basic storage unit is normally a cell (one bit per cell).
  - Multiple RAM chips form a memory.

- RAM comes in two varieties:
  - SRAM (Static RAM) (faster & more expensive, use +power)
  - DRAM (Dynamic RAM)

SRAM vs DRAM Summary

	Trans. per bit	Access time	Needs refresh?	Needs EDC?	Cost	Applications
SRAM	4 or 6	1X	No	Maybe	1000x	Cache memories
DRAM	1	10X	Yes	Yes	1X	Main memories, frame buffers

Textbook pp. 582-586, has many physical details about these memories.
(EDC stands for error detection and correction)

Nonvolatile Memories (pp. 586-589)

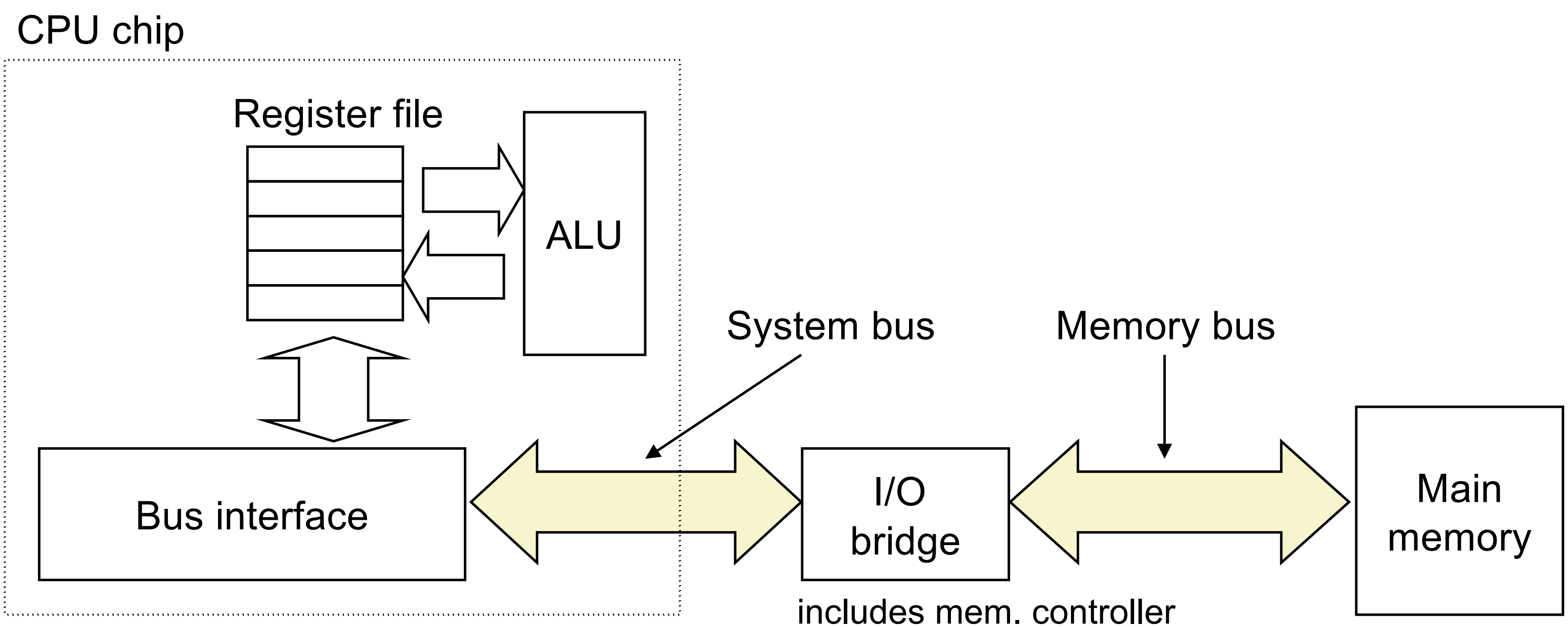
- DRAM and SRAM are volatile memories
  - Lose information if powered off.
- Nonvolatile memories retain value even if powered off
  - Read-only memory (ROM): programmed during production
  - Programmable ROM (PROM): can be programmed once
  - Eraseable PROM (EPROM): can be bulk erased (UV, X-Ray)
  - Electrically eraseable PROM (EEPROM): electronic erase capability
  - Flash memory: EEPROMs. with partial (block-level) erase capability
    - Wears out after about 100,000 erasings
    - Form of disk-drive based on flash mem (later)

Nonvolatile Memories (cont.)

- Uses for Nonvolatile Memories
  - Programs stored in ROM (BIOS, controllers for disks, network cards, graphics accelerators, security subsystems, etc.) are called *firmware*
  - *Solid state disks (SSD)* (replace rotating disks in thumb drives, smart phones, mp3 players, tablets, laptops, etc.)
  - Disk caches

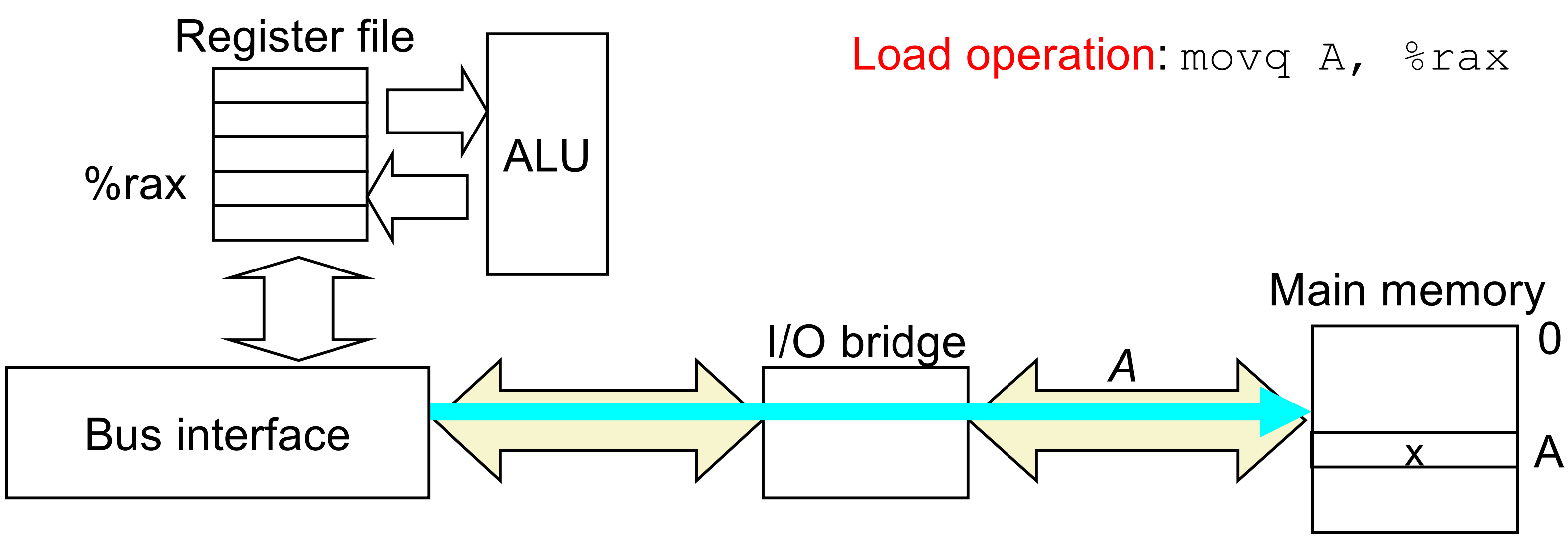
Traditional Bus Structure Connecting CPU and Memory (Accessing main memory, 587-589)

- A bus is a collection of parallel wires that carry address, data, and control signals.
- Buses are typically shared by multiple devices.
- bus transactions: series of steps to transfer CPU-memory



Memory Read Transaction (1)

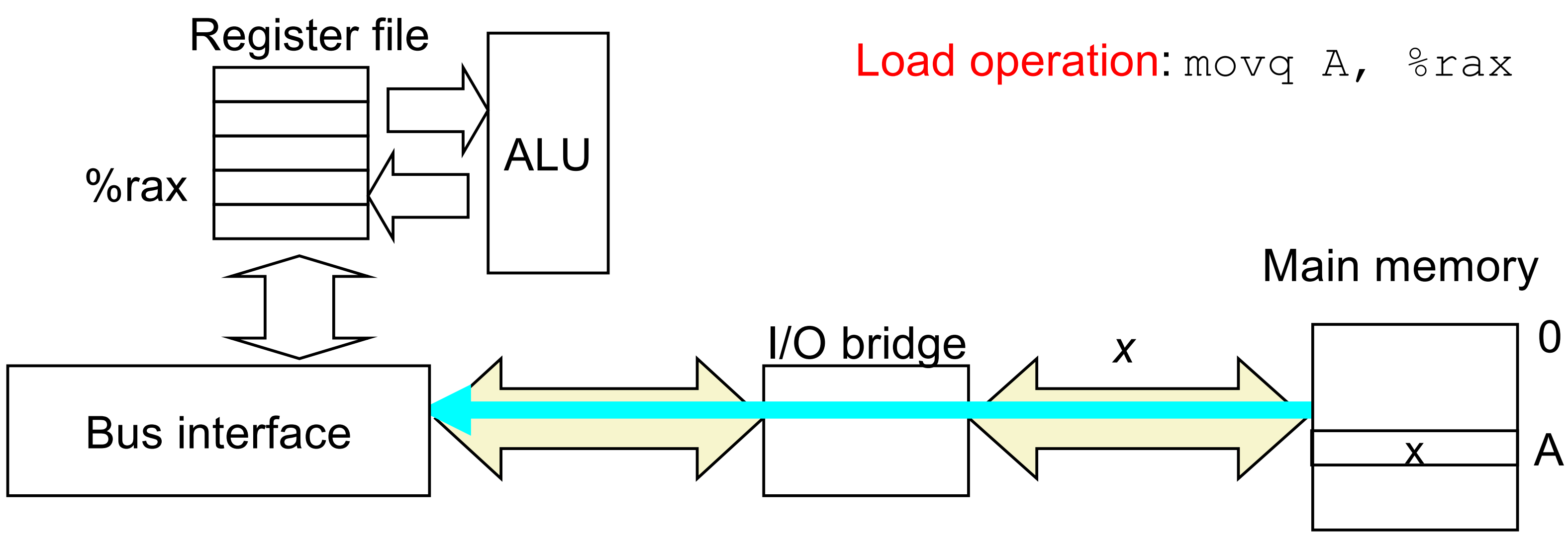
- CPU places address A on the memory bus.



A is an address represented in one of the forms of expressing it i.e. Imm(r_i, r, s)

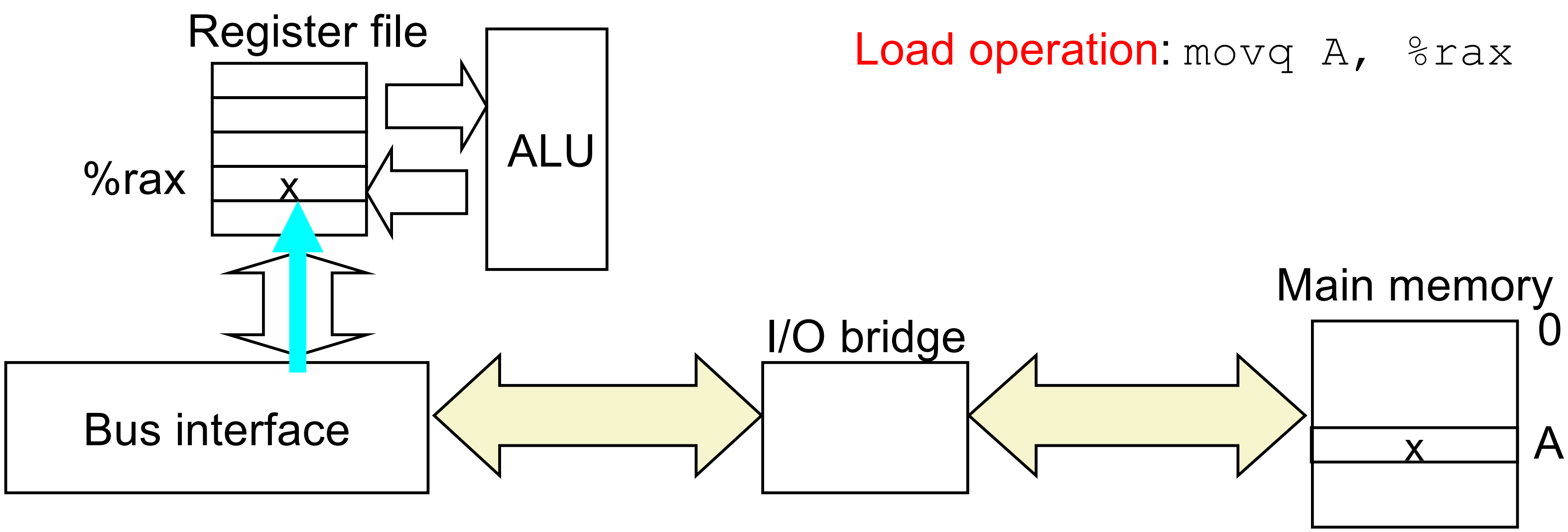
Memory Read Transaction (2)

- Main memory reads A from the memory bus, retrieves word x, and places it on the bus.



Memory Read Transaction (3)

- CPU read word x from the bus and copies it into register %rax.

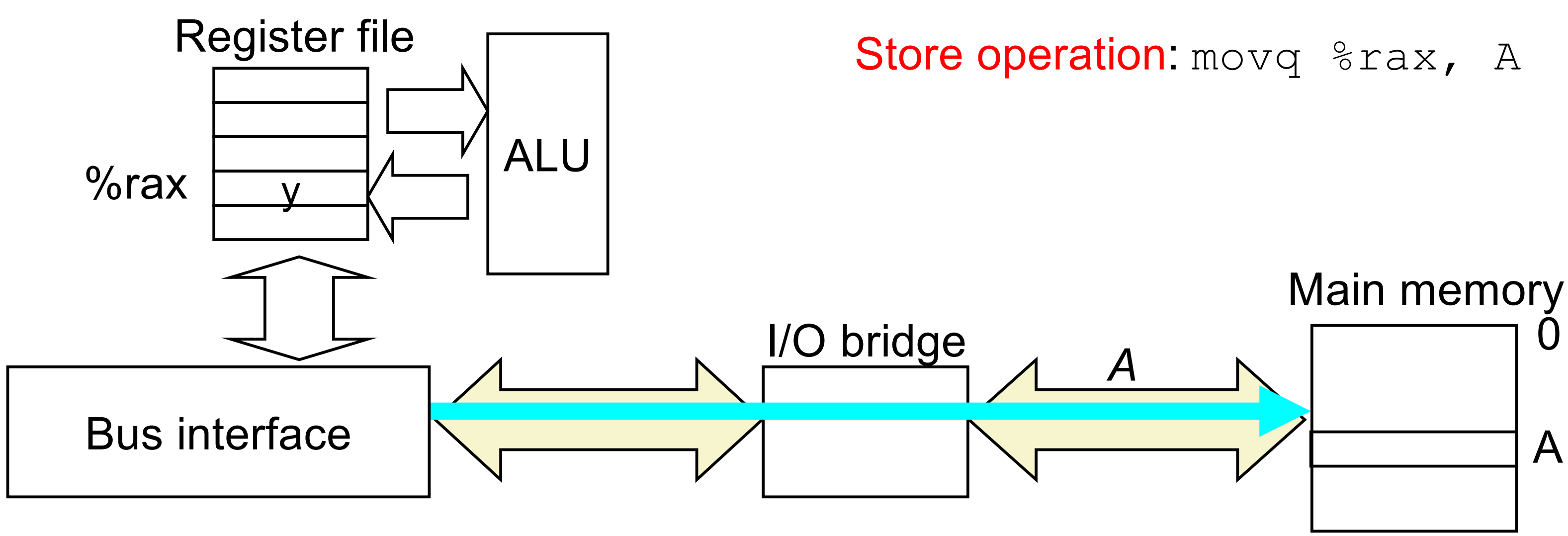


Load operation: `movq A, %rax`

A is an address represented in one of the forms of expressing it i.e. Imm(r_i, r, s)

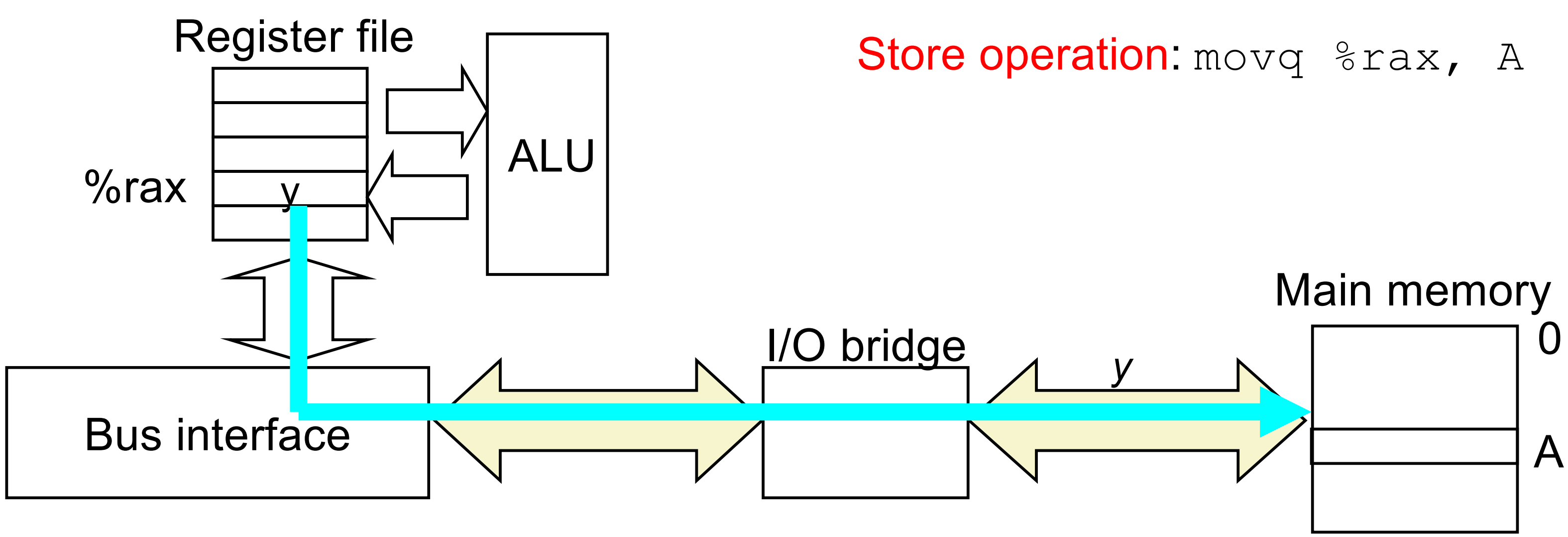
Memory Write Transaction (1)

- CPU places address A on bus. Main memory reads it and waits for the corresponding data word to arrive.



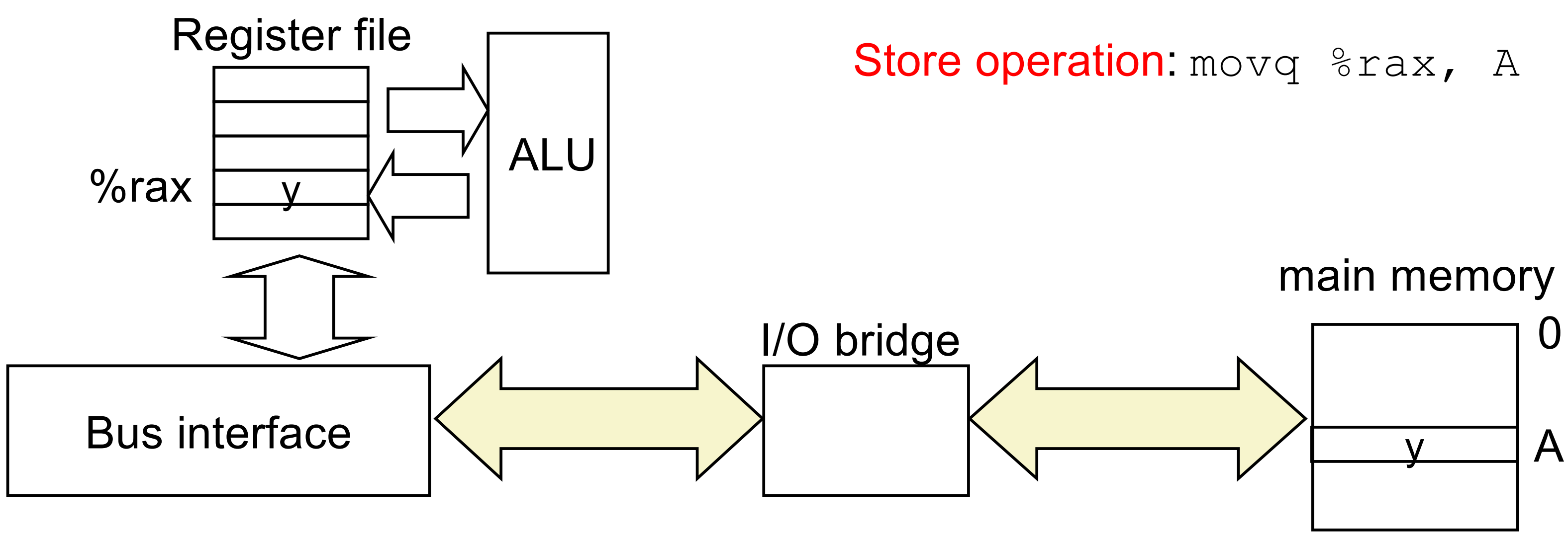
Memory Write Transaction (2)

- CPU places data word y on the bus.



Memory Write Transaction (3)

- Main memory reads data word y from the bus and stores it at address A.



Topics

Chapter 6: The Memory Hierarchy

- Storage technologies and trends (cont.)
  - secondary storage in disk drives
- Locality of reference
- Caching in the memory hierarchy

What’s Inside A Disk Drive?

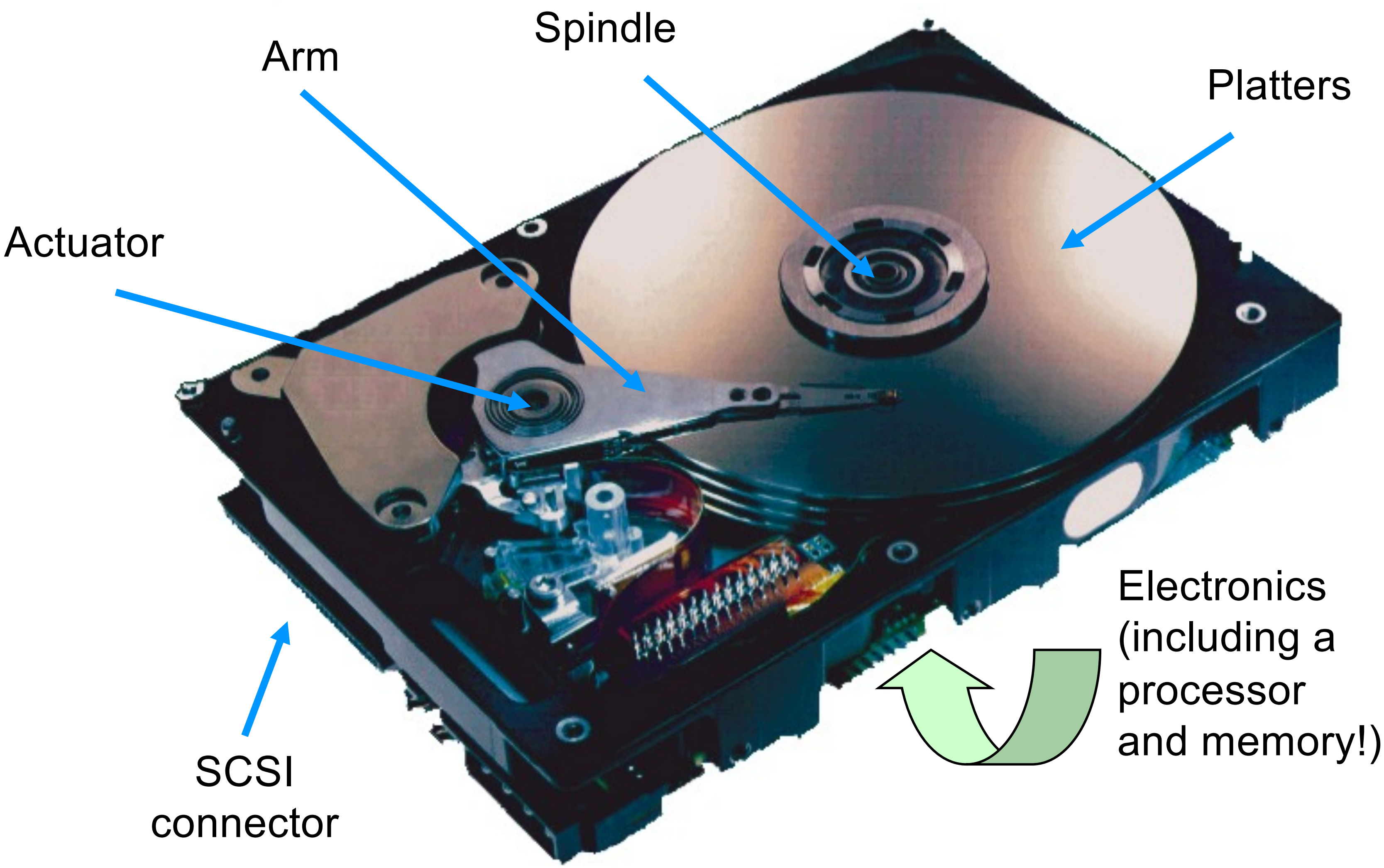
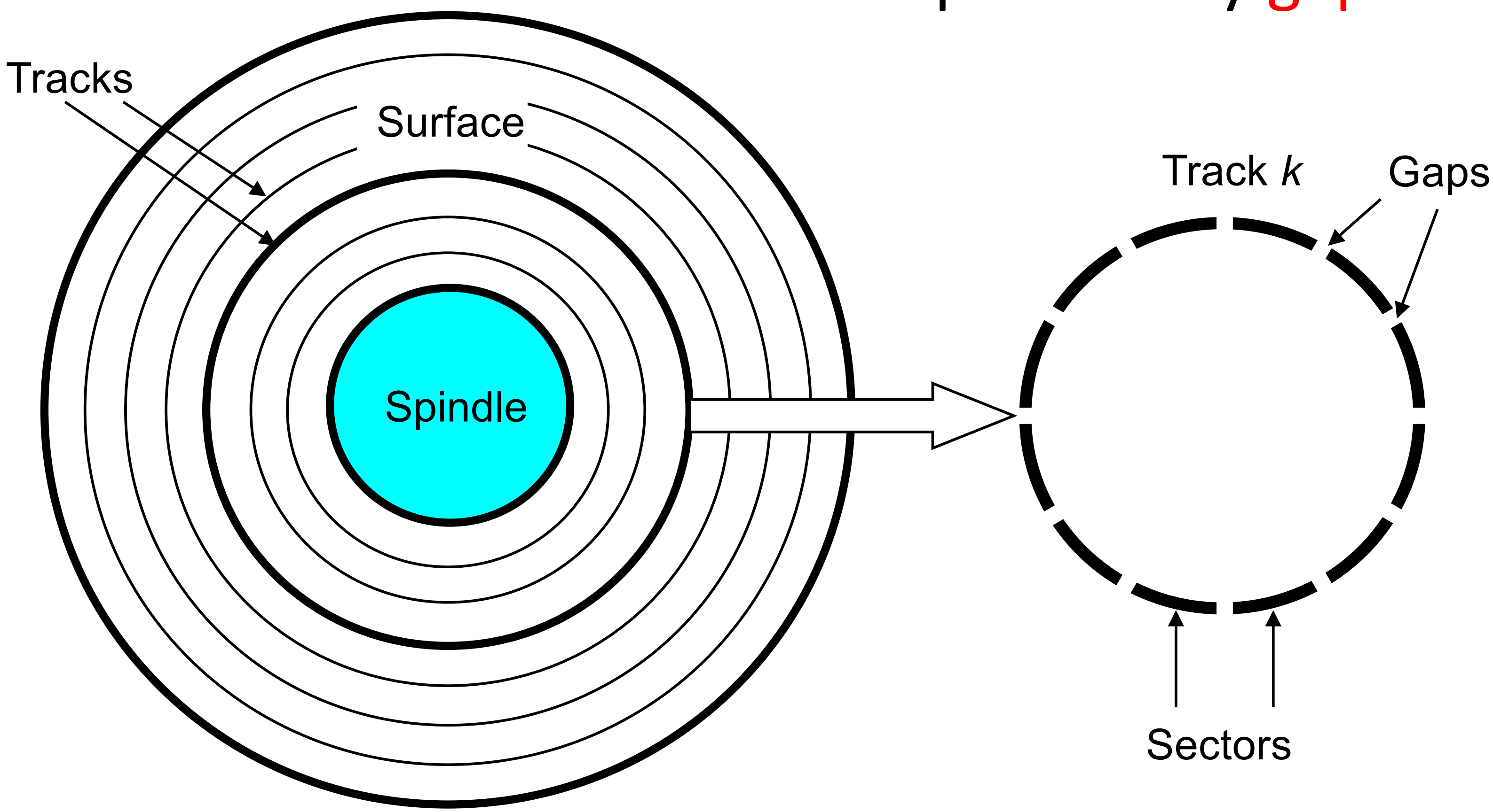


Image courtesy of Seagate Technology

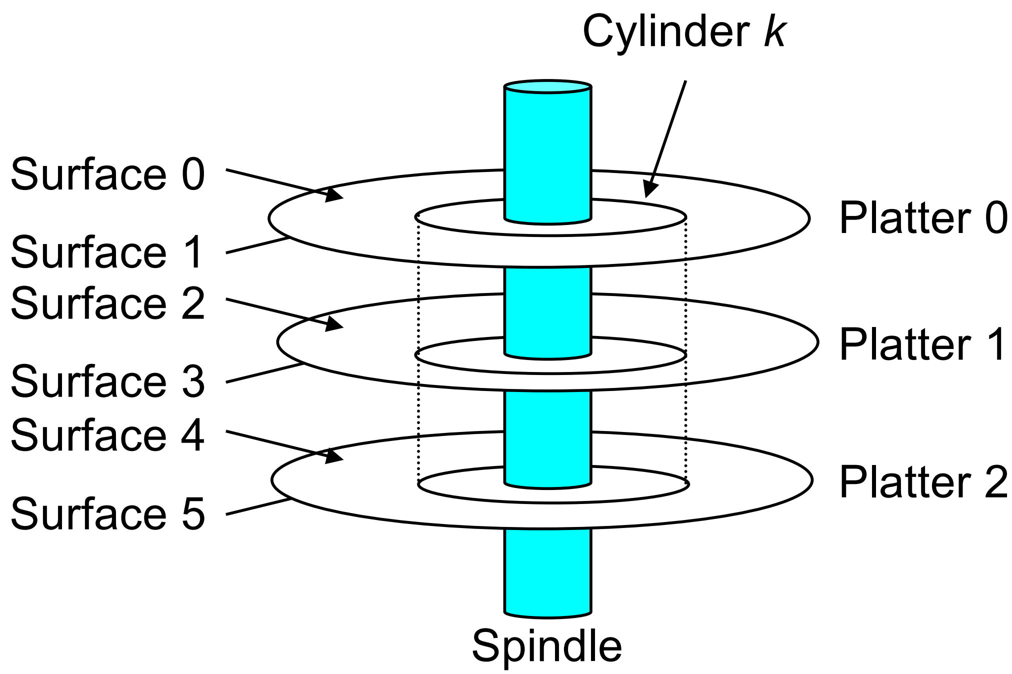
Disk Geometry

- Disks consist of platters, each with two surfaces.
- Each surface consists of concentric rings called tracks.
- Each track consists of sectors separated by gaps.



Disk Geometry (Muliple-Platter View)

- Aligned tracks form a cylinder.

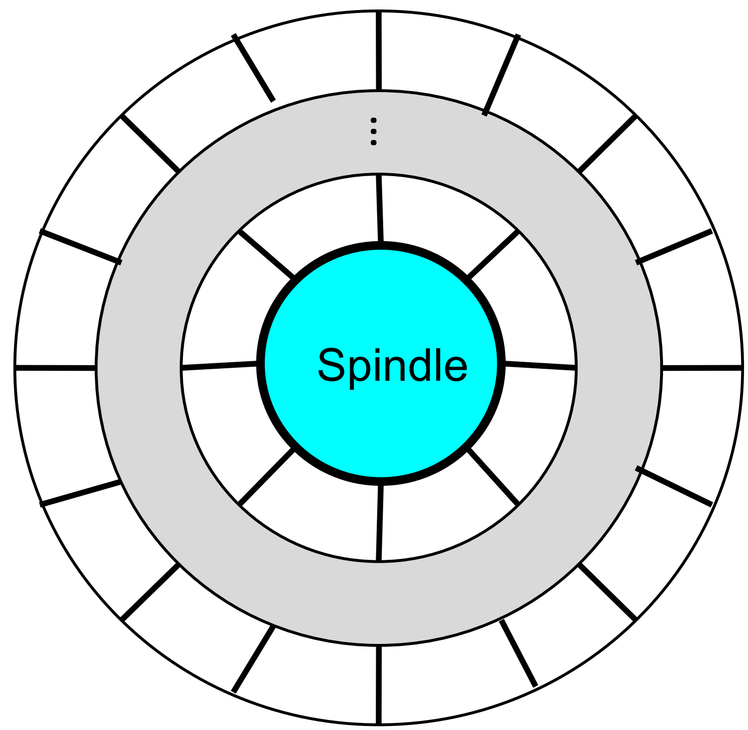


Disk Capacity

- Capacity: maximum number of bits that can be stored.
  - Vendors express capacity in units of gigabytes (GB), where 1 GB = 10^9 Bytes.
- Capacity is determined by these technology factors:
  - Recording density (bits/in): number of bits that can be squeezed into a 1 inch segment of a track.
  - Track density (tracks/in): number of tracks that can be squeezed into a 1 inch radial segment.
  - Areal density (bits/in2): product of recording and track density.

Recording zones

- Modern disks partition tracks into disjoint subsets called recording zones
  - Each track in a zone has the same number of sectors, determined by the circumference of innermost track.
  - Each zone has a different number of sectors/track, outer zones have more sectors/track than inner zones.
  - So we use **average** number of sectors/track when computing capacity.



Computing Disk Capacity

Capacity = (# bytes/sector) x (avg. # sectors/track) x
(# tracks/surface) x (# surfaces/platter) x
(# platters/disk)

Example:

- 512 bytes/sector
- 300 sectors/track (on average)
- 20,000 tracks/surface
- 2 surfaces/platter
- 5 platters/disk

Capacity = 512 x 300 x 20000 x 2 x 5
= 30,720,000,000
= 30.72 GB

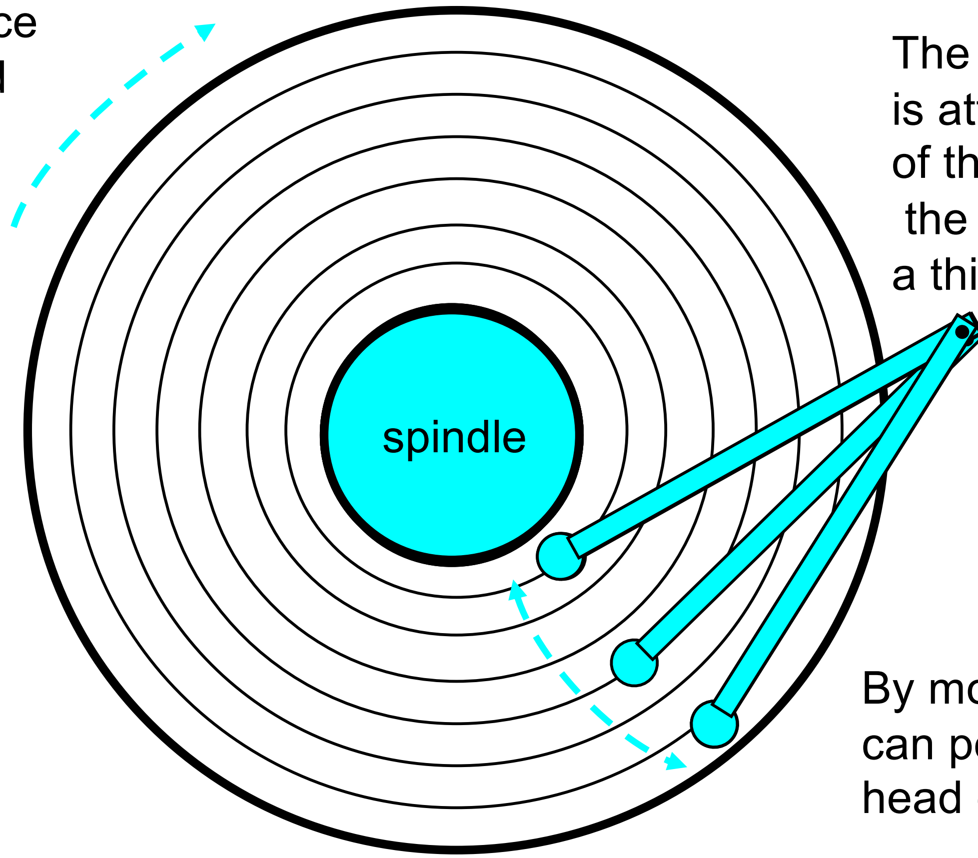
Units: 1GB = 10^9 bytes
1TB = 10^{12} bytes

Problem 6.2 (p. 592)

- What is the capacity of a disk with 2 platters, 10,000 cylinders, an average of 400 sectors per track, and 512 bytes per sector?

Disk Operation (Single-Platter View)

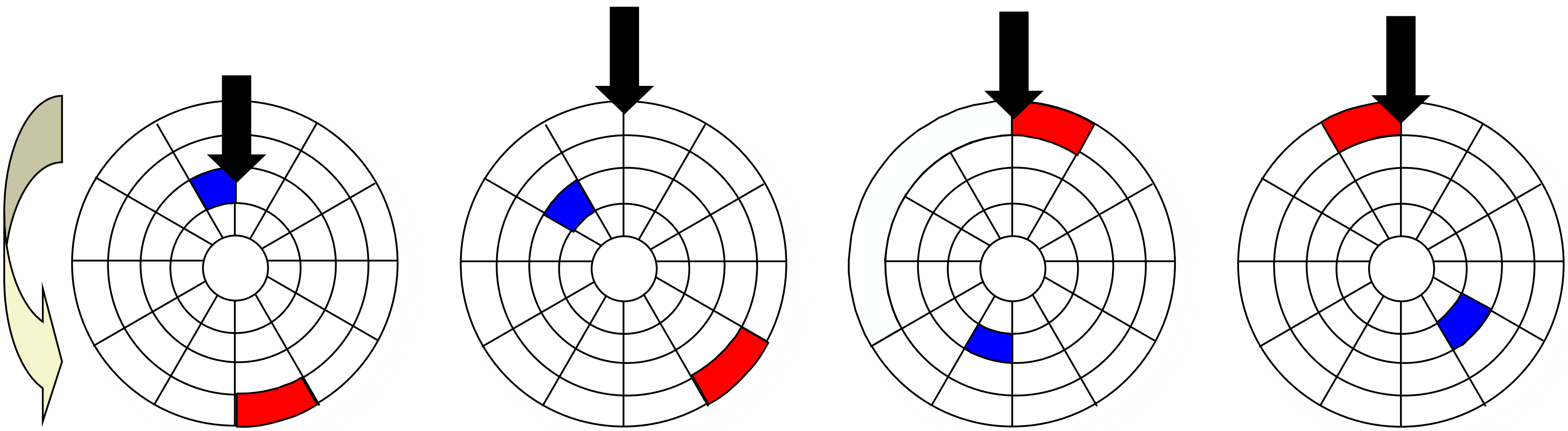
The disk surface spins at a fixed rotational rate



The read/write *head* is attached to the end of the *arm* and flies over the disk surface on a thin cushion of air.

By moving radially, the arm can position the read/write head over any track.

Disk Access – Service Time Components



After BLUE read | Seek for RED | Rotational latency | After RED read

Data transfer | Seek | Rotational latency | Data transfer

Topics

Chapter 6: The Memory Hierarchy

- Storage technologies and trends (cont.)
  - secondary storage in disk drives
    - accessing times
    - logical disk blocks, I/O bus, reading and writing on disk
- Locality of reference
- Caching in the memory hierarchy

Disk Access Time

- Average time to access some target sector approximated:
  - $T_{access} = T_{avg\ seek} + T_{avg\ rotation} + T_{avg\ transfer}$
- Seek time ($T_{avg\ seek}$)
  - Time to position heads over cylinder containing target sector.
  - Typical $T_{avg\ seek}$ is 3—9 ms
- Rotational latency ($T_{avg\ rotation}$)
  - Time waiting for first bit of target sector to pass under r/w head.
  - $T_{avg\ rotation}$ = 1/2 x 1/RPMs x 60 sec/1 min
  - Typical $T_{avg\ rotation}$ = 7200 RPMs
- Transfer time ($T_{avg\ transfer}$)
  - Time to read the bits in the target sector.
  - $T_{avg\ transfer}$ = 1/RPM x 1/(avg # sectors/track) x 60 secs/1 min.

Disk Access Time Example

- Given:
  - Rotational rate = 7,200 RPM
  - Average seek time = 9 ms
  - Average # sectors/track = 400
- Derived:
  - $T_{avg\ rotation}$ = 1/2 x (60 secs/7200 RPM) x 1000 ms/sec = 4 ms.
  - $T_{avg\ transfer}$ = 60/7200 RPM x 1/400 secs/track x 1000 ms/sec = 0.02 ms
  - T_{access} = 9 ms + 4 ms + 0.02 ms
- Important points:
  - Access time dominated by seek time and rotational latency.
  - First bit in a sector is the most expensive, the rest are free.
  - SRAM access time is about 4 ns/doubleword, DRAM about 60 ns
    - Disk is about 40,000 times slower than SRAM,
    - 2,500 times slower than DRAM.

Logical Disk Blocks

- Modern disks present a simpler abstract view of the complex sector geometry:
  - The set of available sectors is modeled as a sequence of B-sized (sector sized) logical blocks (0, 1, 2, ...)
- Mapping between logical blocks and actual (physical) sectors
  - Maintained by hardware/firmware device called disk controller.
  - Converts requests for logical blocks into (surface,track,sector) triples.
- Allows controller to set aside *spare cylinders* for each zone. (for bad cylinders)
  - Accounts for the difference in "formatted capacity" and "maximum capacity".

Practice Problem 6.4 (pp. 595- 596)

- This exercise is fundamental to understand why good locality is a determining factor on disk access performance.
- Given the characteristics of a disk, calculate the time to access and transfer a file in two different situations:
  - Best case
  - Random case

Parameters for problem 6.4

1 MB file consisting of 512-byte logical blocks is stored on a disk drive with the following characteristics:

Rotational rate: 10,000RPM ➔ 60sec/10,000rpm = 0.006 sec (per rotation) ➔ 6ms per rotation

$T_{avgRotation} = ½\, T_{maxRotation} = 6/2$ ms = 3 ms

$T_{avgseek}$: 5 ms (average seek time)

Average number of sectors/track: 1,000

Surfaces: 4

Sector size: 512 bytes

Statement of Problem 6.4

For each case below, suppose that a Program reads the logical blocks of the file sequentially and that the time to position the head over the first block is Tavgseek + Tavgrotation

- A. Best case: Estimate the optimal time (in ms) required to read the file given the best possible mapping of logical blocks to disk sectors (i.e., sequential).
- B. Random case: Estimate the time (in ms) required to read the file if blocks are mapped randomly to disk sectors.