



Memory Hierarchy

CS:APP 6.1

These slides adapted from materials provided by the textbook authors.

Memory Hierarchy

- Storage technologies and trends
- Locality of reference
- Caching in the memory hierarchy

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)
- DRAM (Dynamic RAM)

SRAM vs DRAM Summary

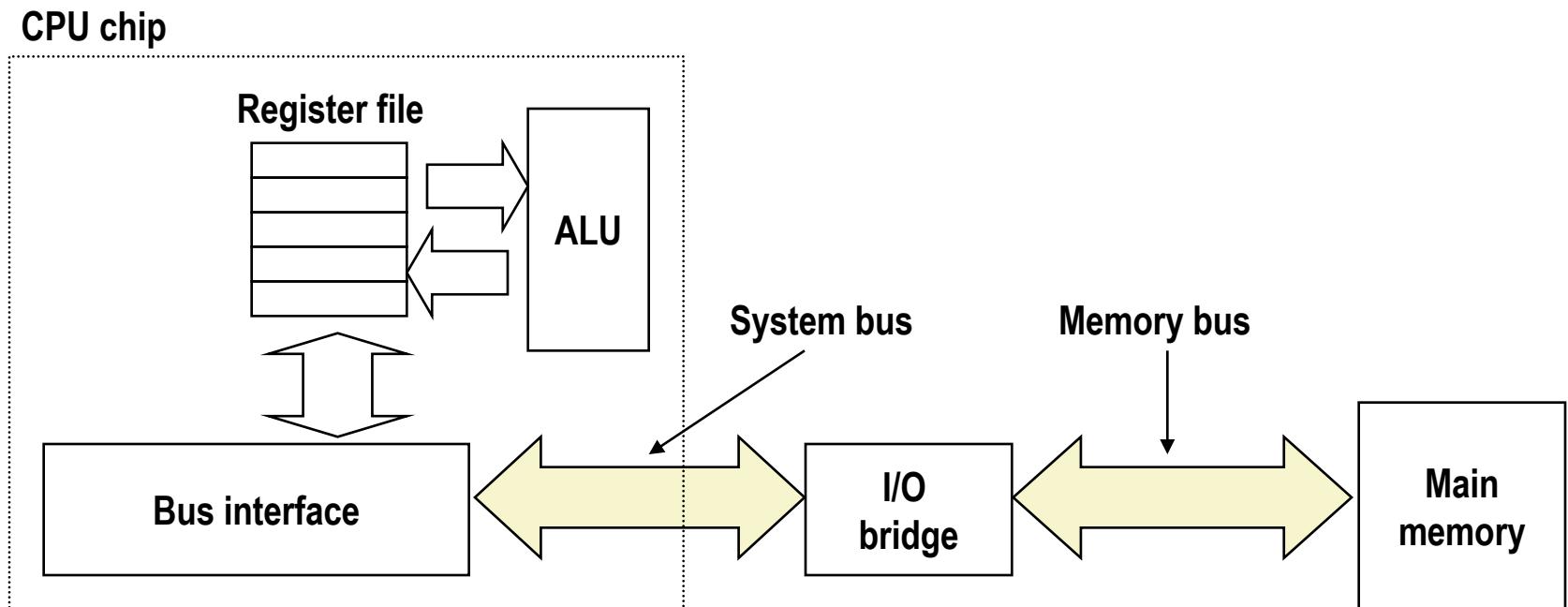
	Trans. per bit	Access time	Needs refresh?	Needs EDC?	Cost	Applications
SRAM	4 or 6	1X	No	Maybe	100x	Cache memories
DRAM	1	10X	Yes	Yes	1X	Main memories, frame buffers

Nonvolatile Memories

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

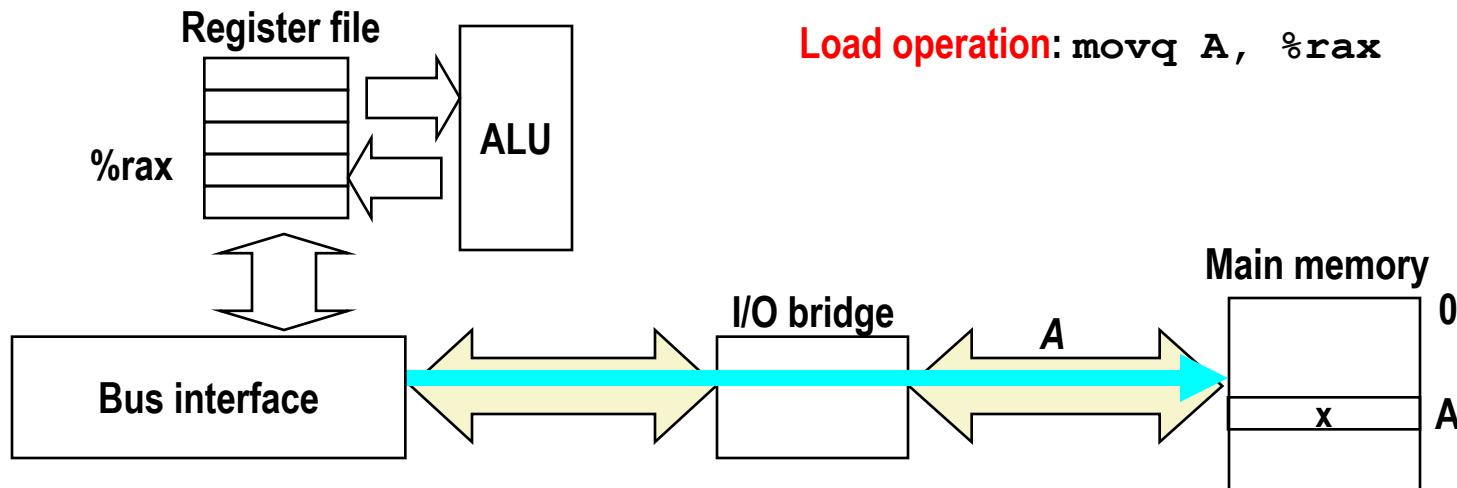
Traditional Bus Structure Connecting CPU and Memory

- A **bus** is a collection of parallel wires that carry address, data, and control signals.
- Buses are typically shared by multiple devices.



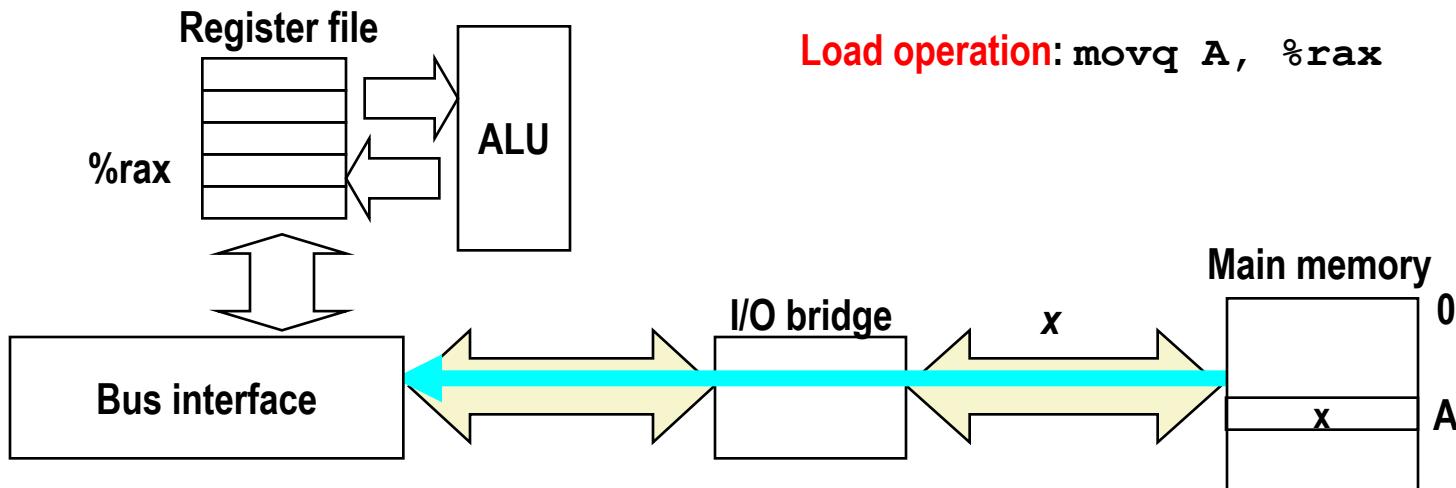
Memory Read Transaction (1)

- CPU places address A on the memory bus.



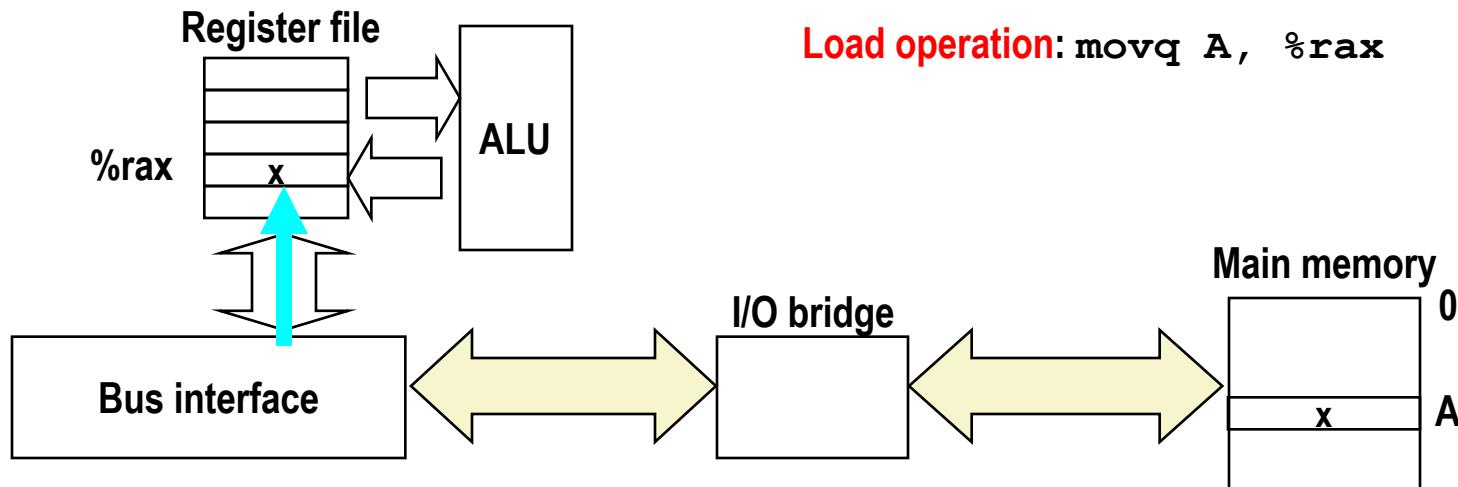
Memory Read Transaction (2)

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



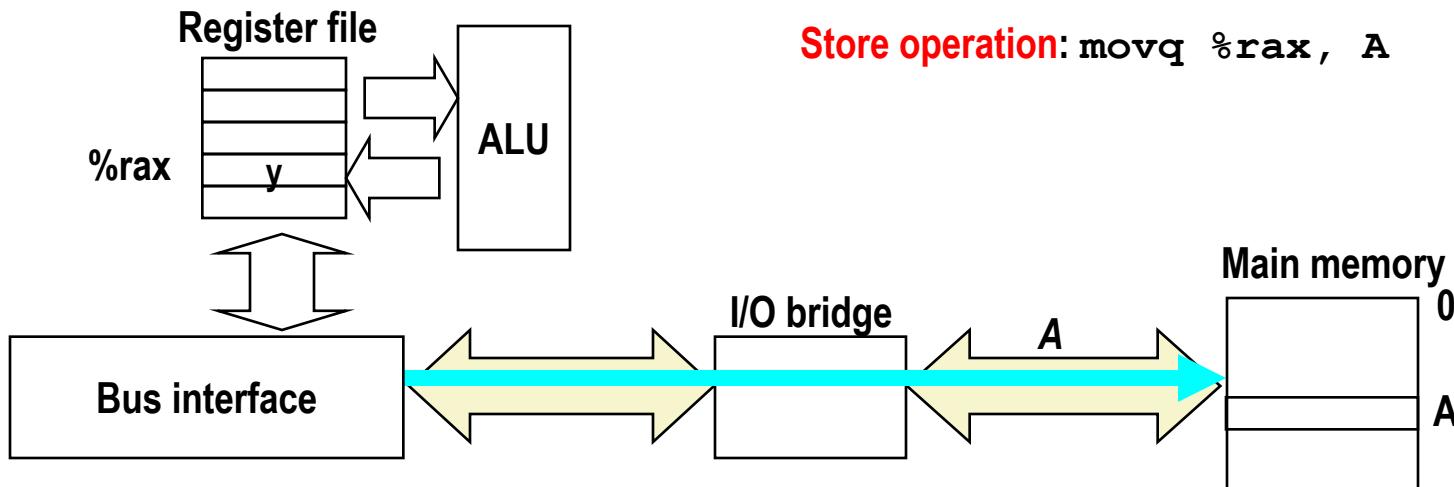
Memory Read Transaction (3)

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



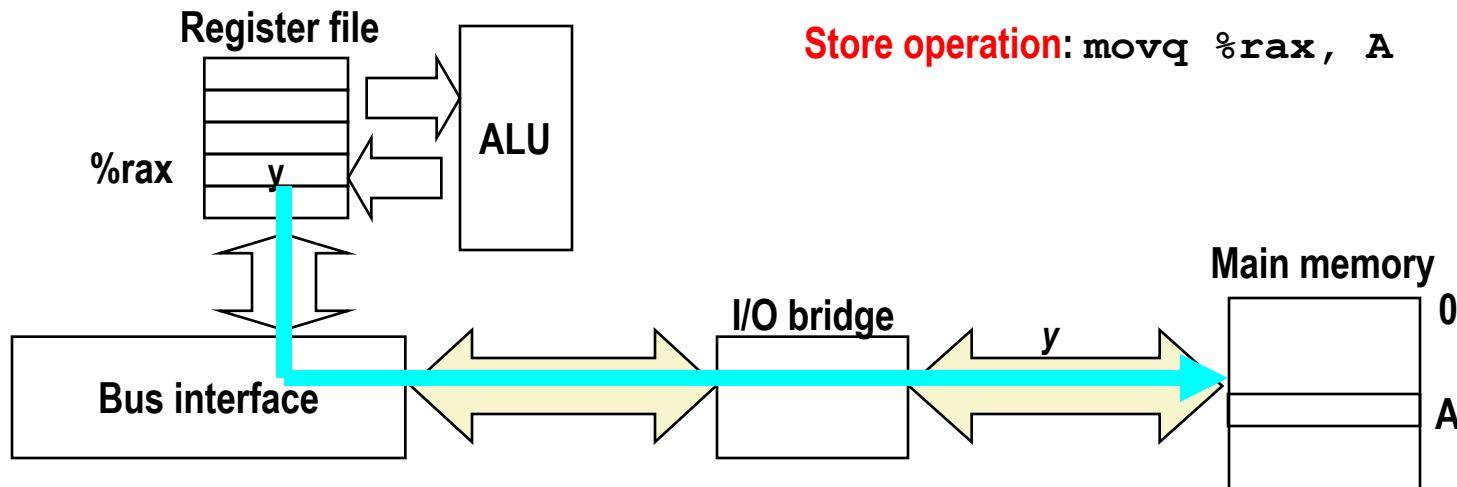
Memory Write Transaction (1)

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



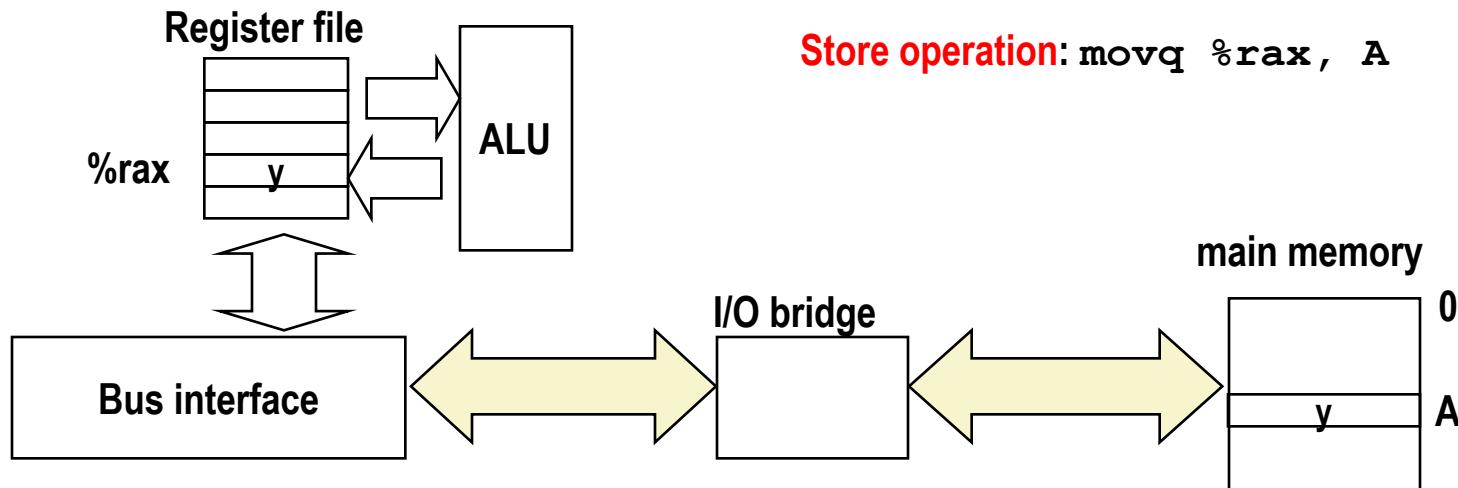
Memory Write Transaction (2)

- CPU places data word y on the bus.



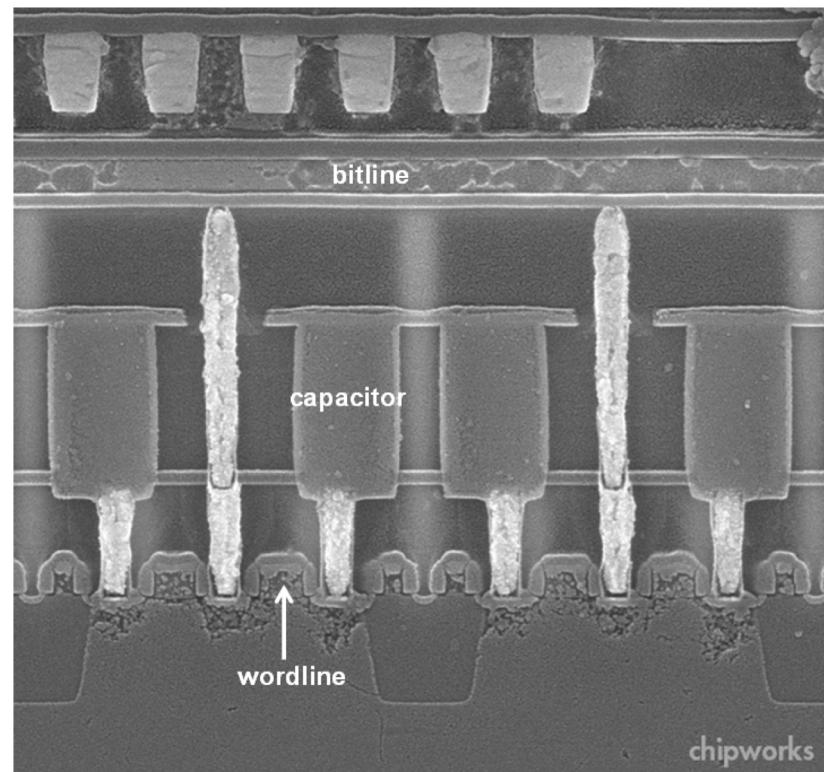
Memory Write Transaction (3)

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



What's in a DRAM cell?

- Dynamic RAM stores a charge in a capacitor
- That charge depletes over time and when read
- Reading and writing DRAM requires both reading and writing to replenish charge
- “refresh” needed as charge depeletes



65nm Embedded DRAM in Xbox GPU

What's Inside A Disk Drive?

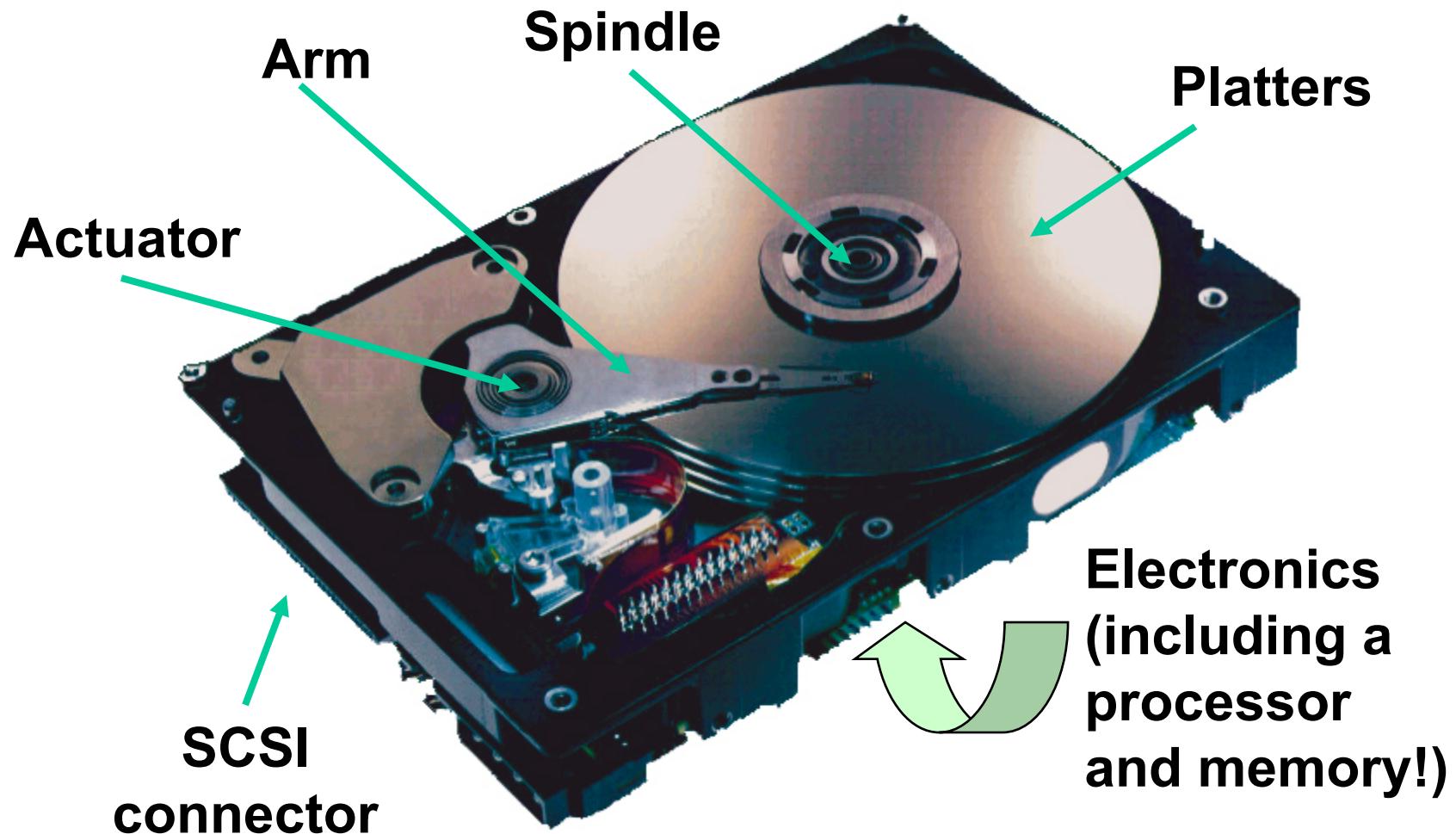
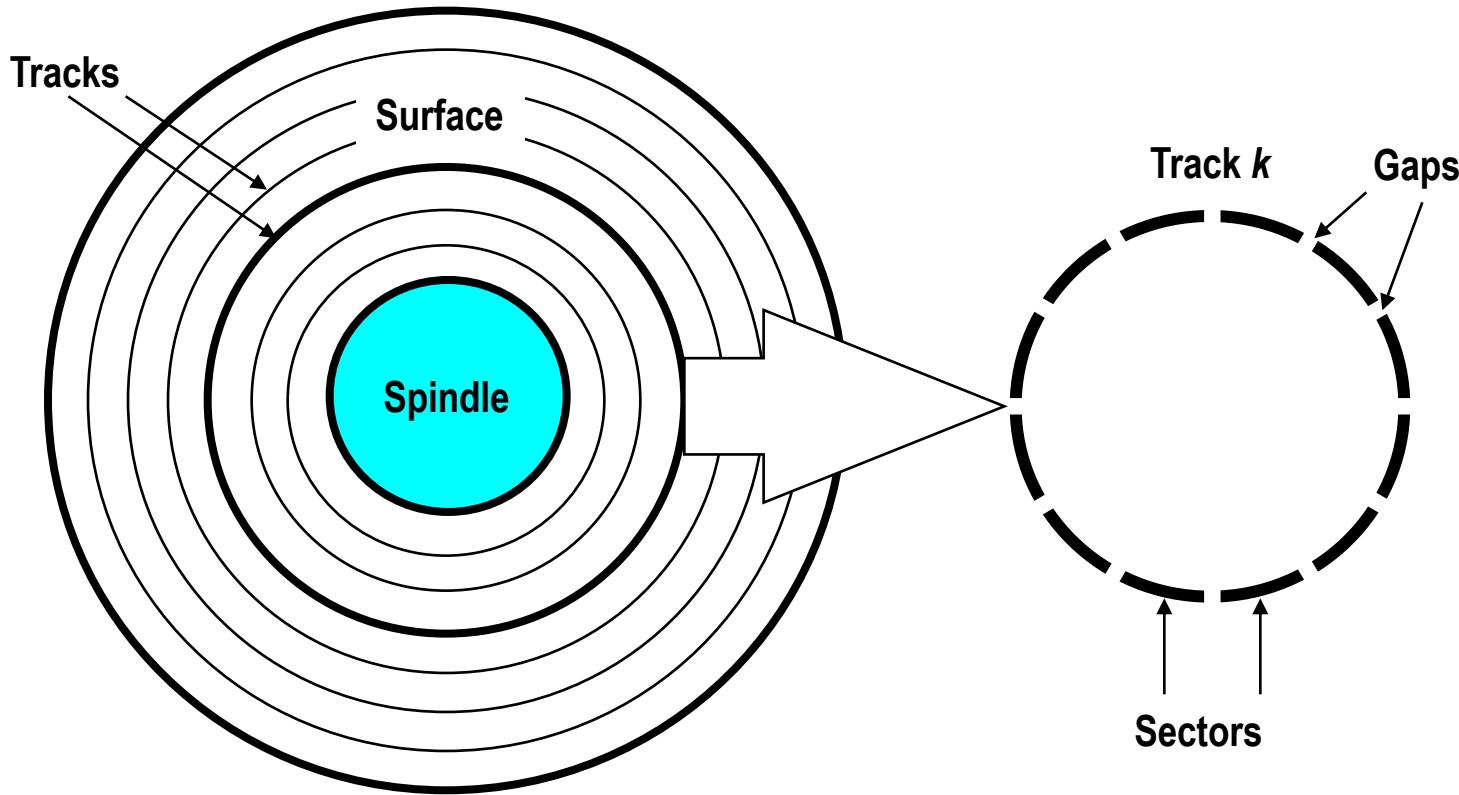


Image courtesy of Seagate Technology

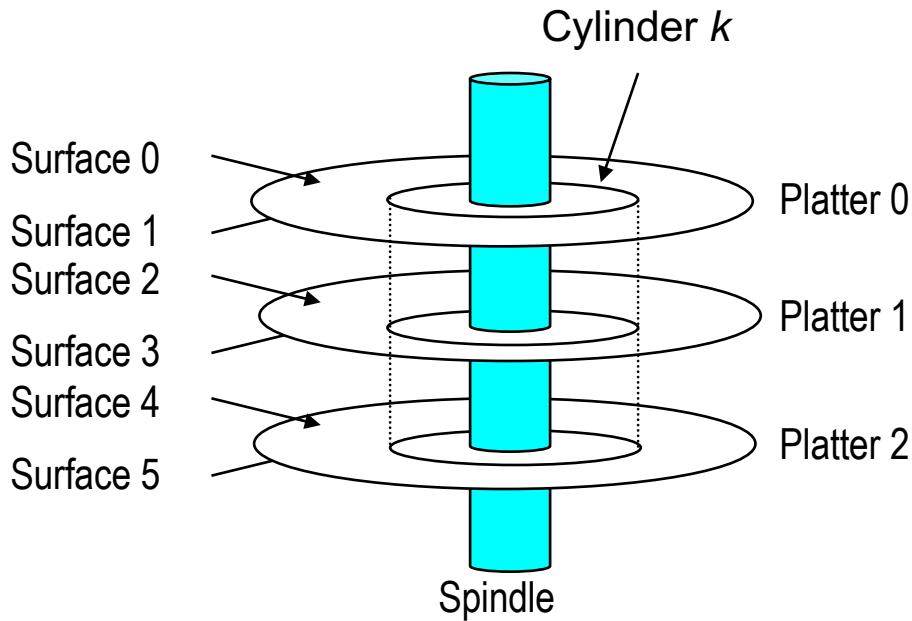
Disk Geometry

- 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 (Multiple-Platter View)

- Aligned tracks form a cylinder.

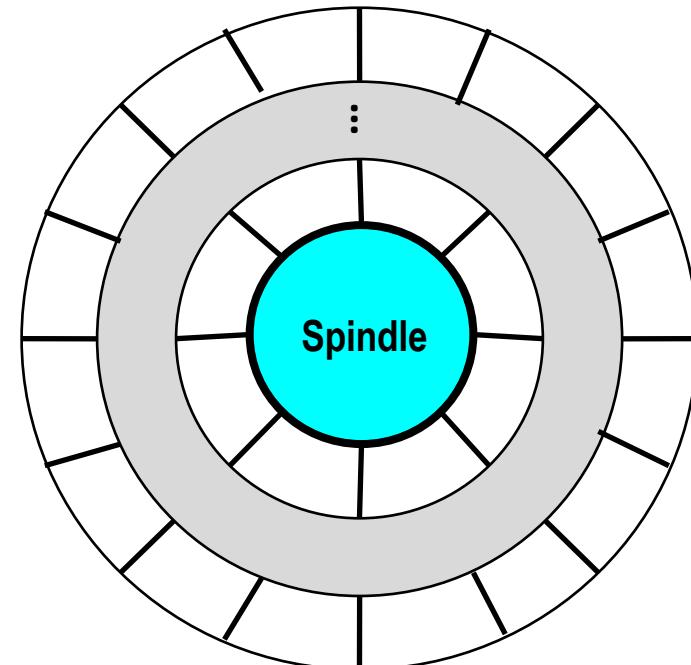


Disk Capacity

- **Capacity:** maximum number of bits that can be stored.
 - Vendors express capacity in units of gigabytes (GB), where
 $1 \text{ GB} = 10^9 \text{ Bytes}$.
- **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)**

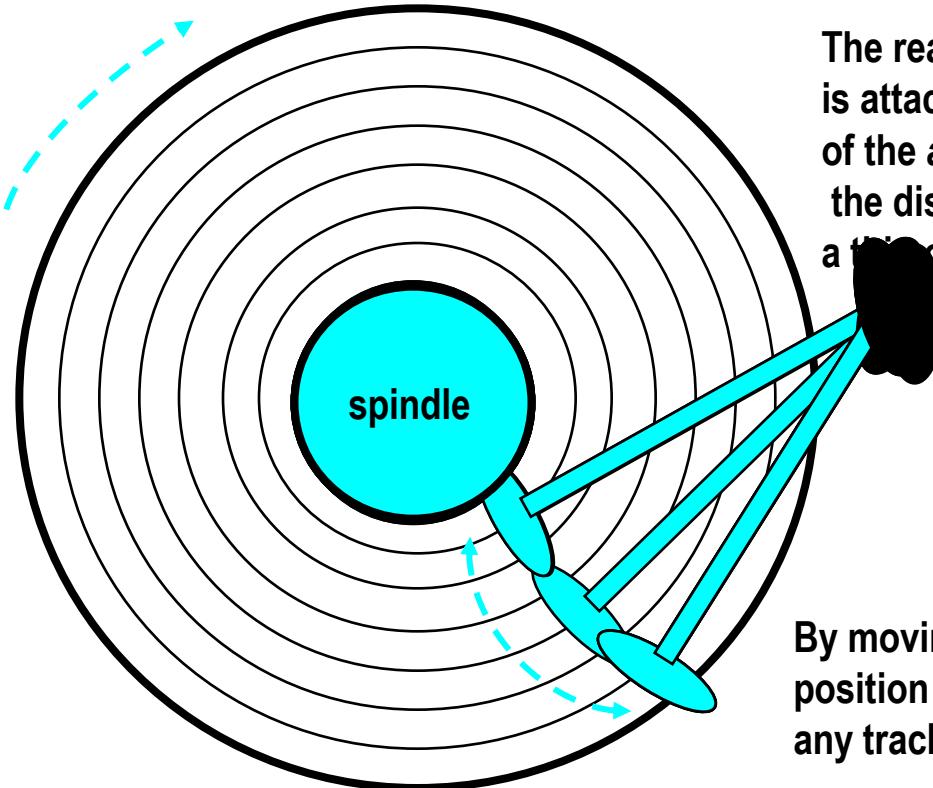
Example:

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

$$\begin{aligned}\text{Capacity} &= 512 \times 300 \times 20000 \times 2 \times 5 \\ &= 30,720,000,000 \\ &= 30.72 \text{ GB}\end{aligned}$$

Disk Operation (Single-Platter View)

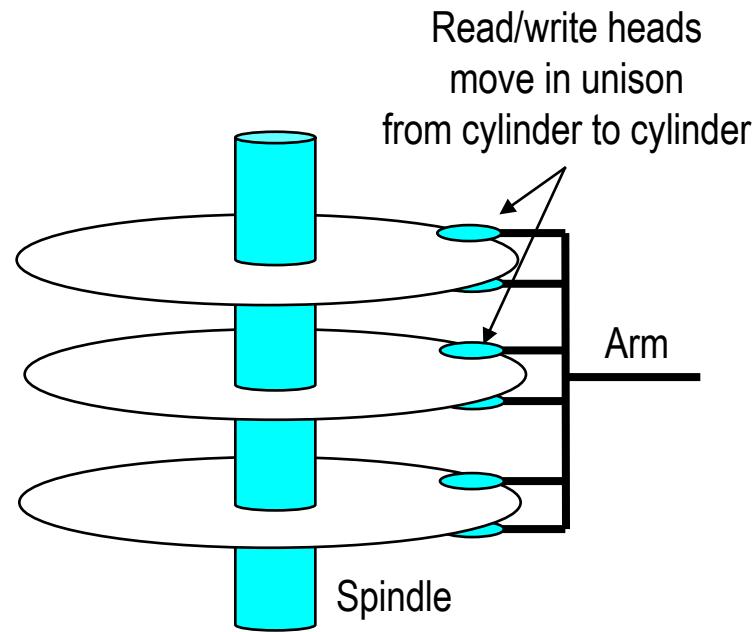
The disk surface spins at a fixed rotational rate



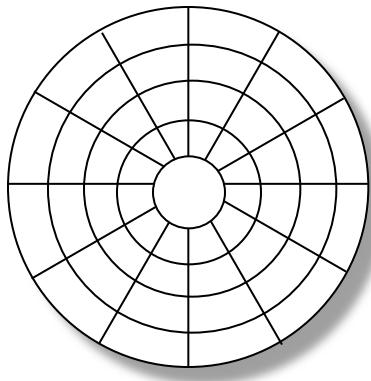
The read/write *head* is attached to the end of the *arm* and flies over the disk surface on a cushion of air.

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

Disk Operation (Multi-Platter View)



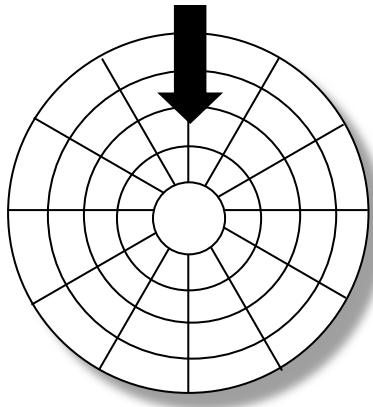
Disk Structure - top view of single platter



Surface organized into tracks

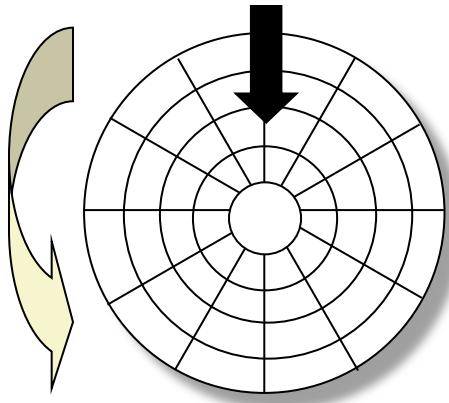
Tracks divided into sectors

Disk Access



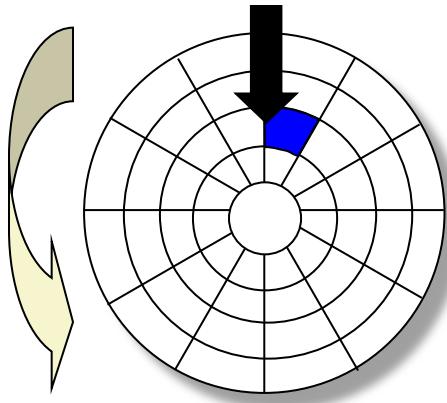
Head in position above a track

Disk Access



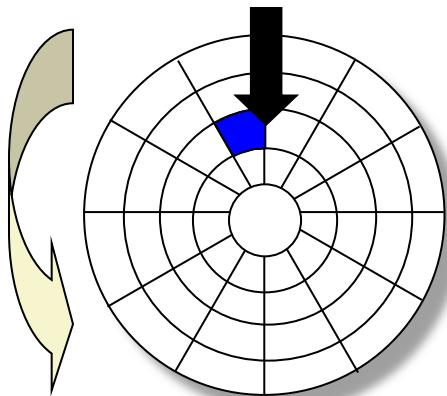
Rotation is counter-clockwise

Disk Access – Read



About to read blue sector

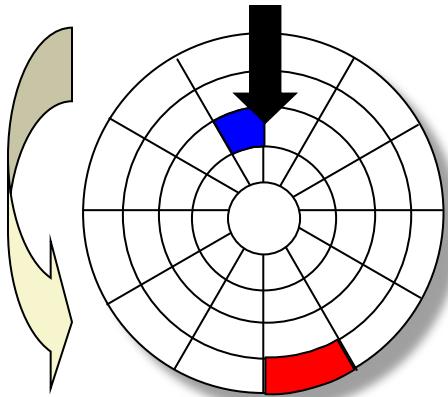
Disk Access – Read



After **BLUE** read

After reading blue sector

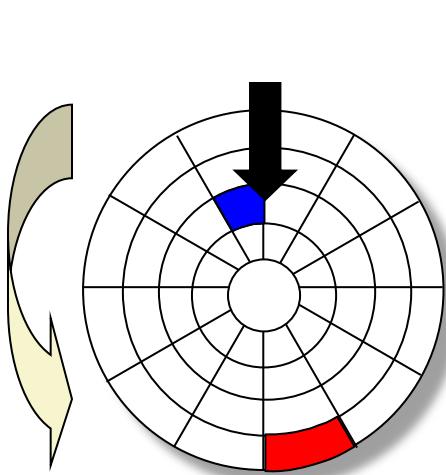
Disk Access – Read



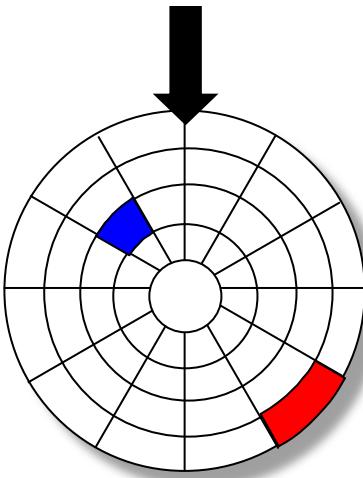
After **BLUE** read

Red request scheduled next

Disk Access – Seek



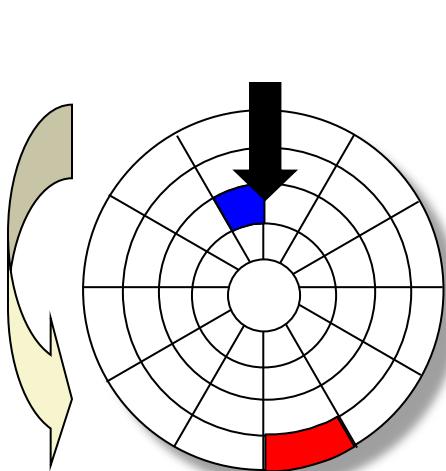
After **BLUE** read



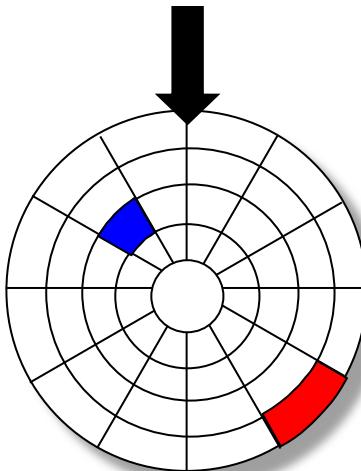
Seek for **RED**

Seek to red's track

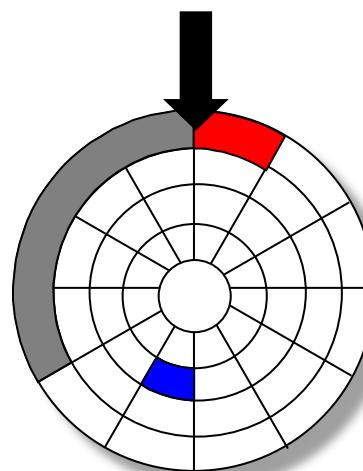
Disk Access – Rotational Latency



After **BLUE** read



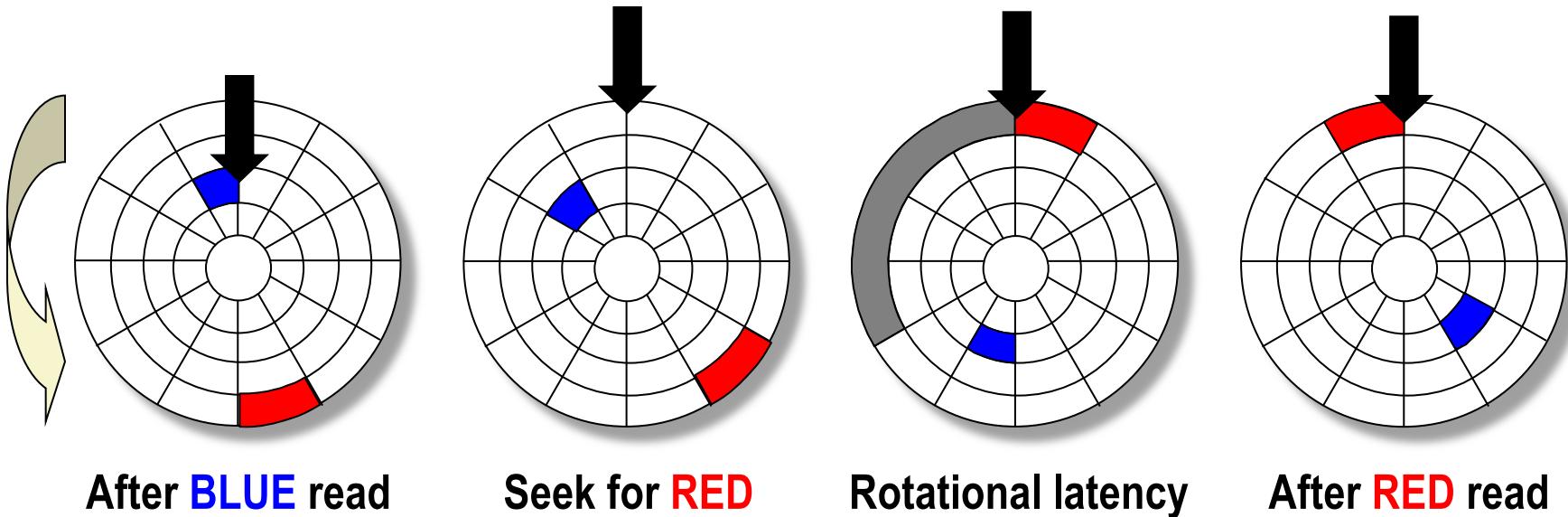
Seek for **RED**



Rotational latency

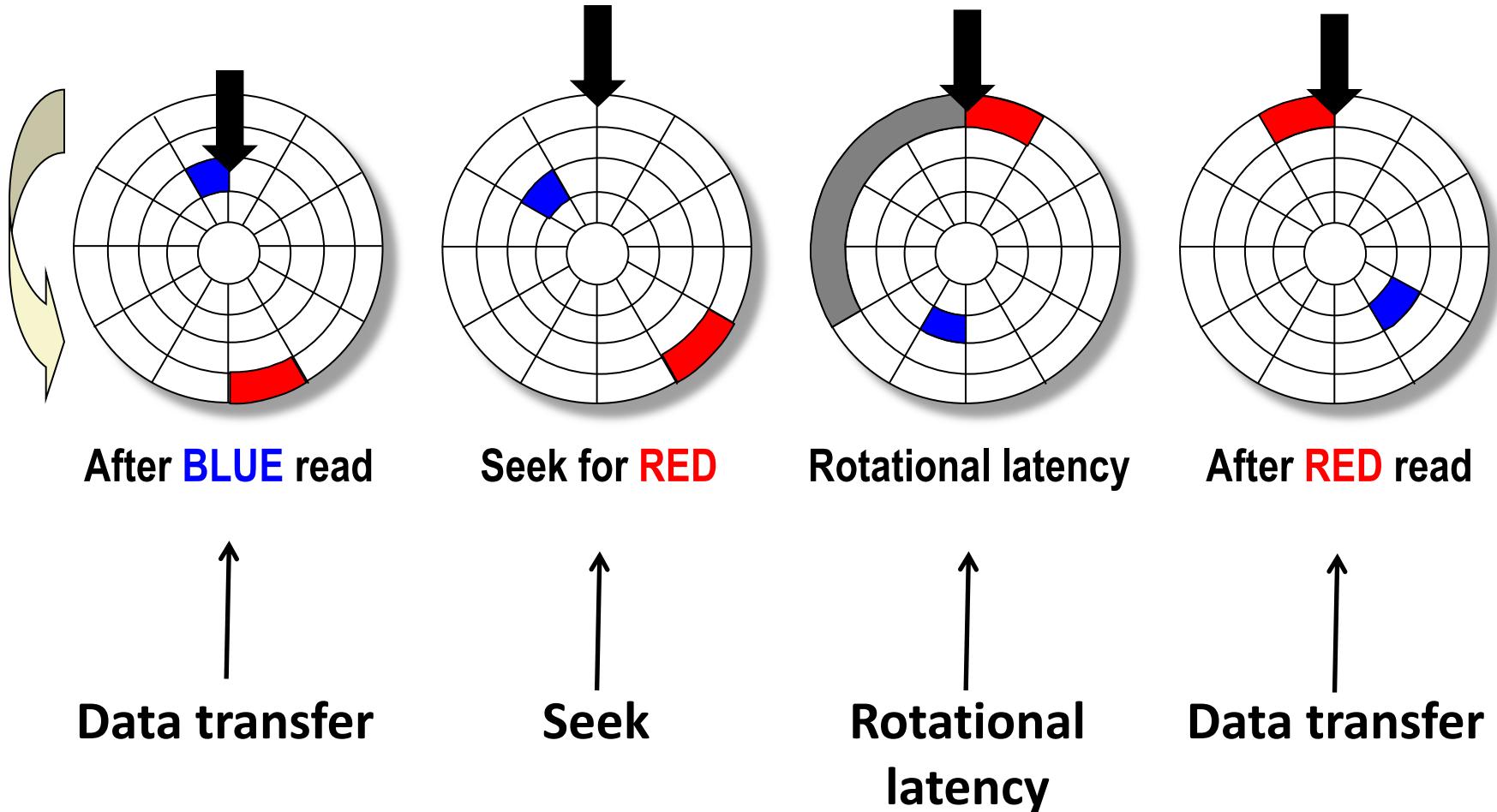
Wait for red sector to rotate around

Disk Access – Read



Complete read of red

Disk Access – Service Time Components



Disk Access Time

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

Disk Access Time Example

■ Given:

- Rotational rate = 7,200 RPM
- Average seek time = 9 ms.
- Avg # sectors/track = 400.

■ Derived:

- $T_{avg\ rotation} = 1/2 \times (60\ secs/7200\ RPM) \times 1000\ ms/sec = 4\ ms.$
- $T_{avg\ transfer} = 60/7200\ RPM \times 1/400\ secs/track \times 1000\ ms/sec = 0.02\ ms$
- $T_{access} = 9\ ms + 4\ ms + 0.02\ ms$

■ Important points:

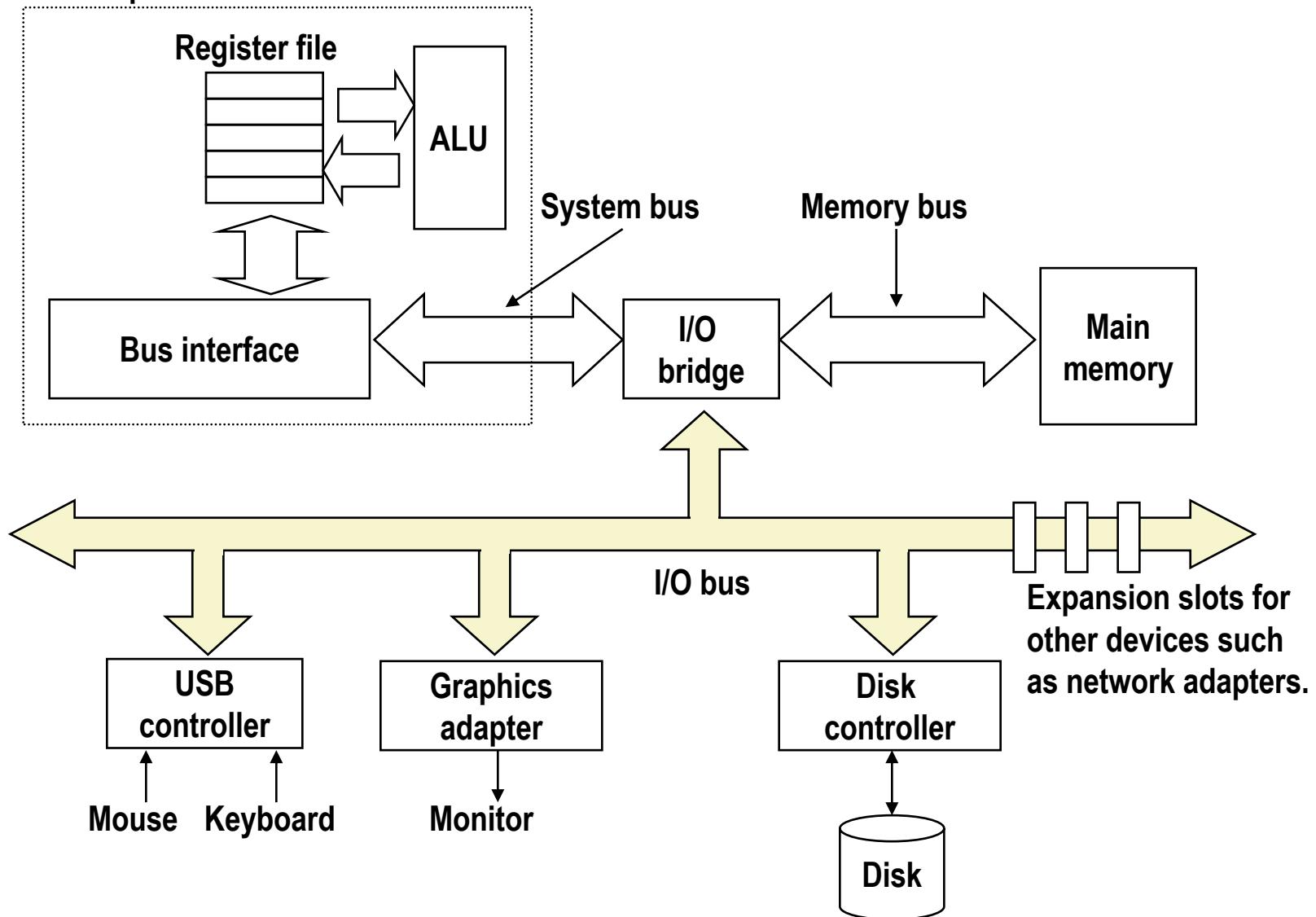
- Access time dominated by seek time and rotational latency.
- First bit in a sector is the most expensive, the rest are free.
- SRAM access time is about 4 ns/doubleword, DRAM about 60 ns
 - Disk is about 40,000 times slower than SRAM,
 - 2,500 times slower than DRAM.

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 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.
 - Accounts for the difference in “formatted capacity” and “maximum capacity”.

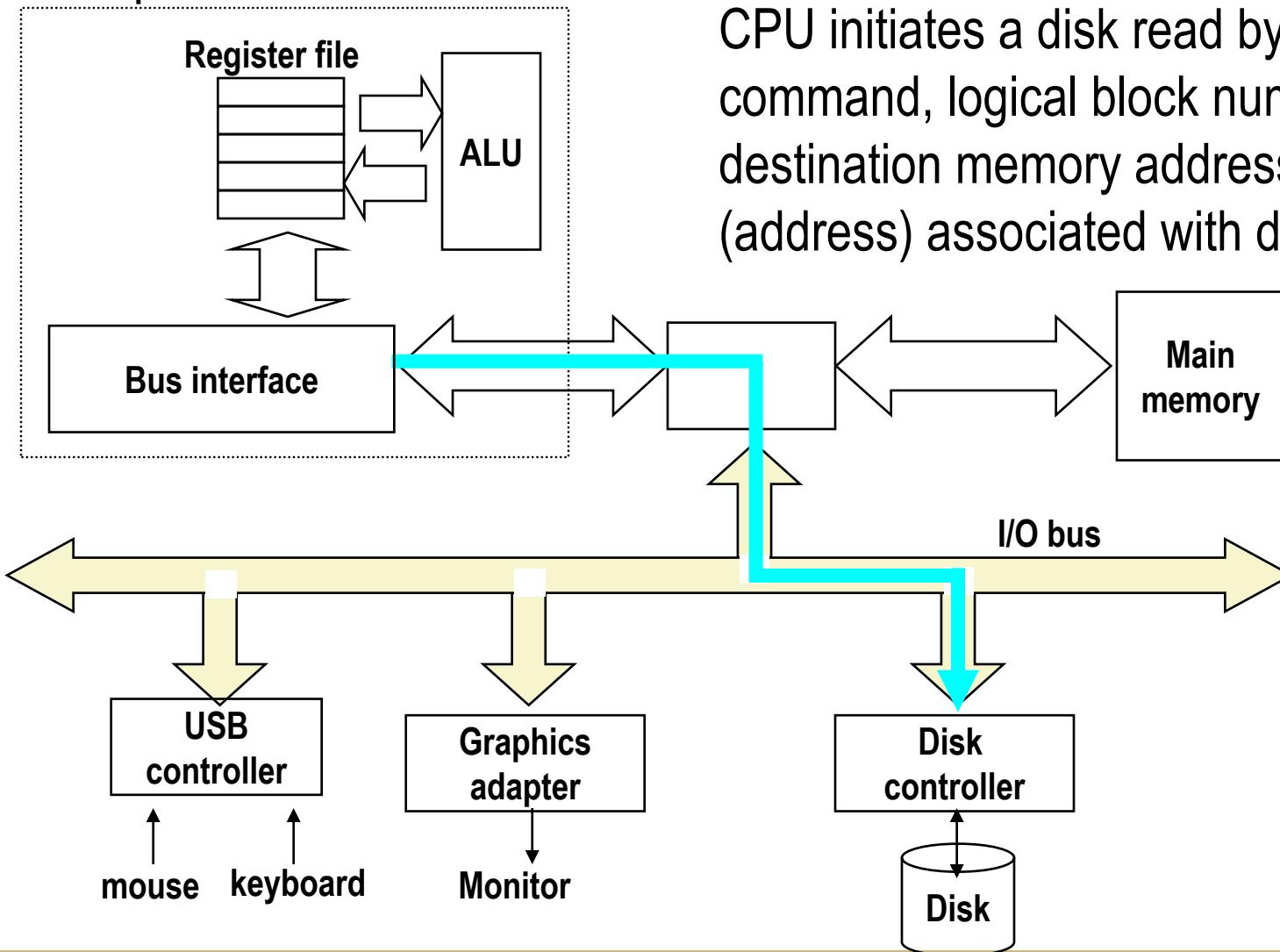
I/O Bus

CPU chip



Reading a Disk Sector (1)

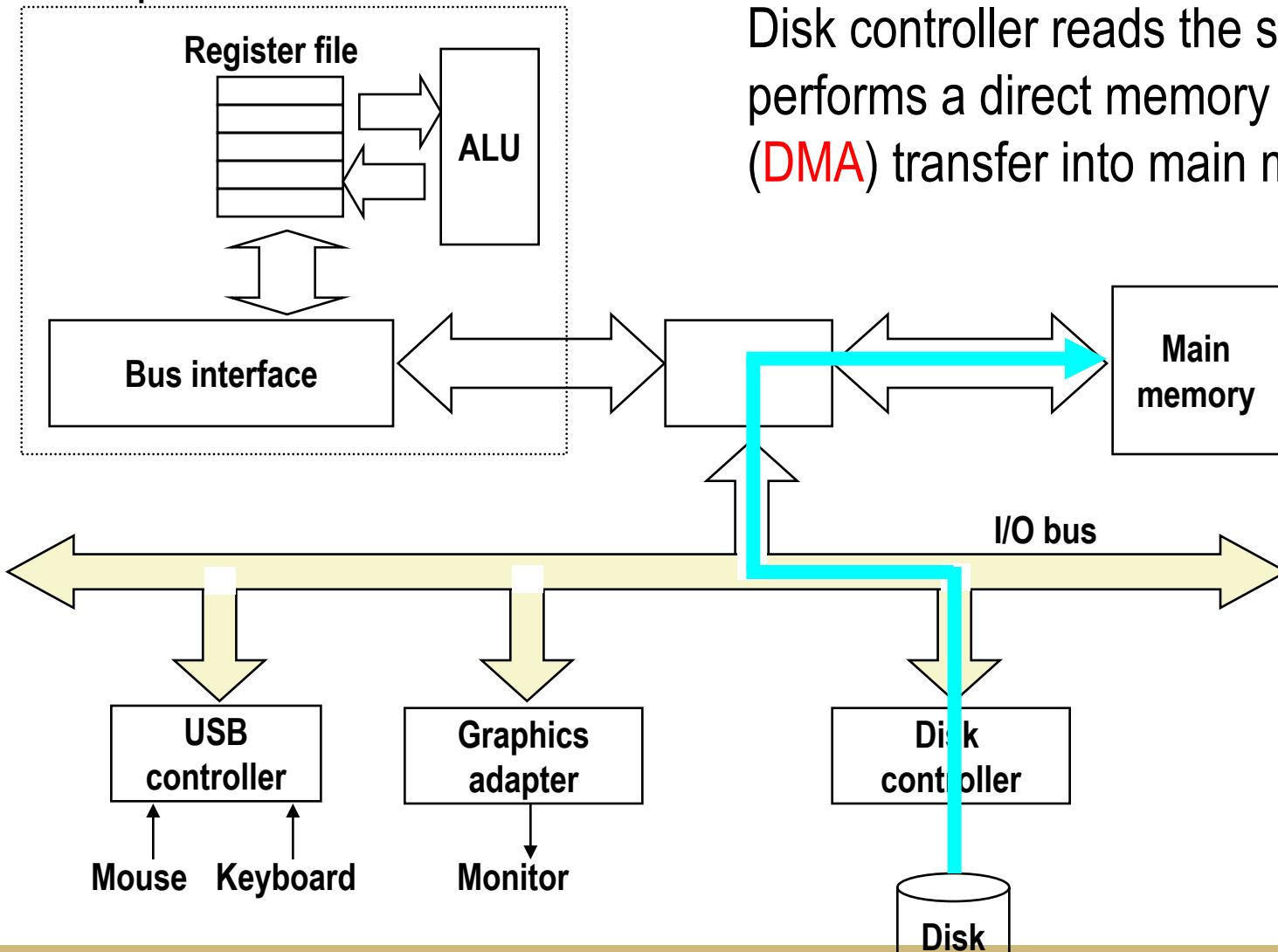
CPU chip



CPU initiates a disk read by writing a command, logical block number, and destination memory address to a **port** (address) associated with disk controller.

Reading a Disk Sector (2)

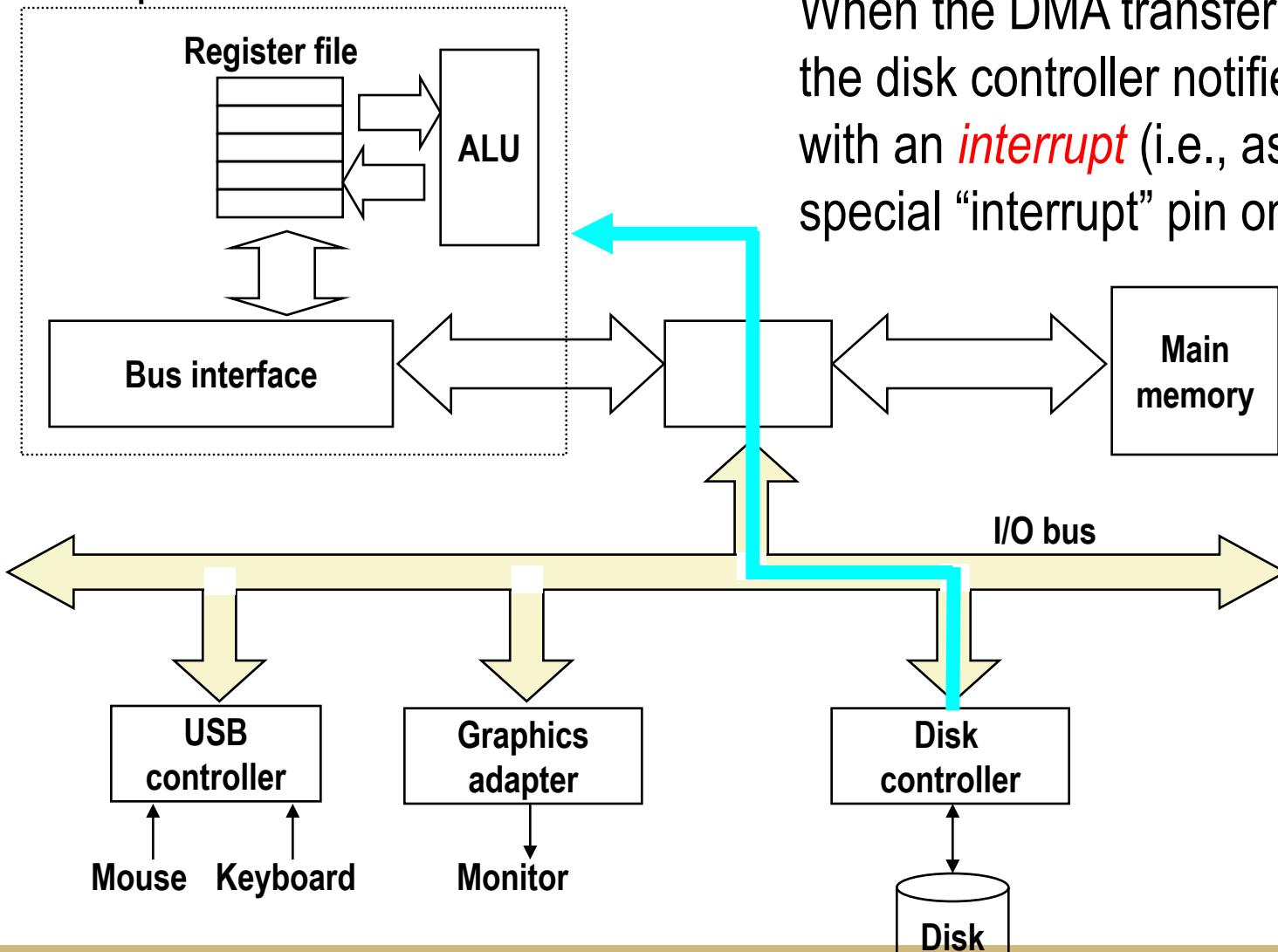
CPU chip



Disk controller reads the sector and performs a direct memory access (**DMA**) transfer into main memory.

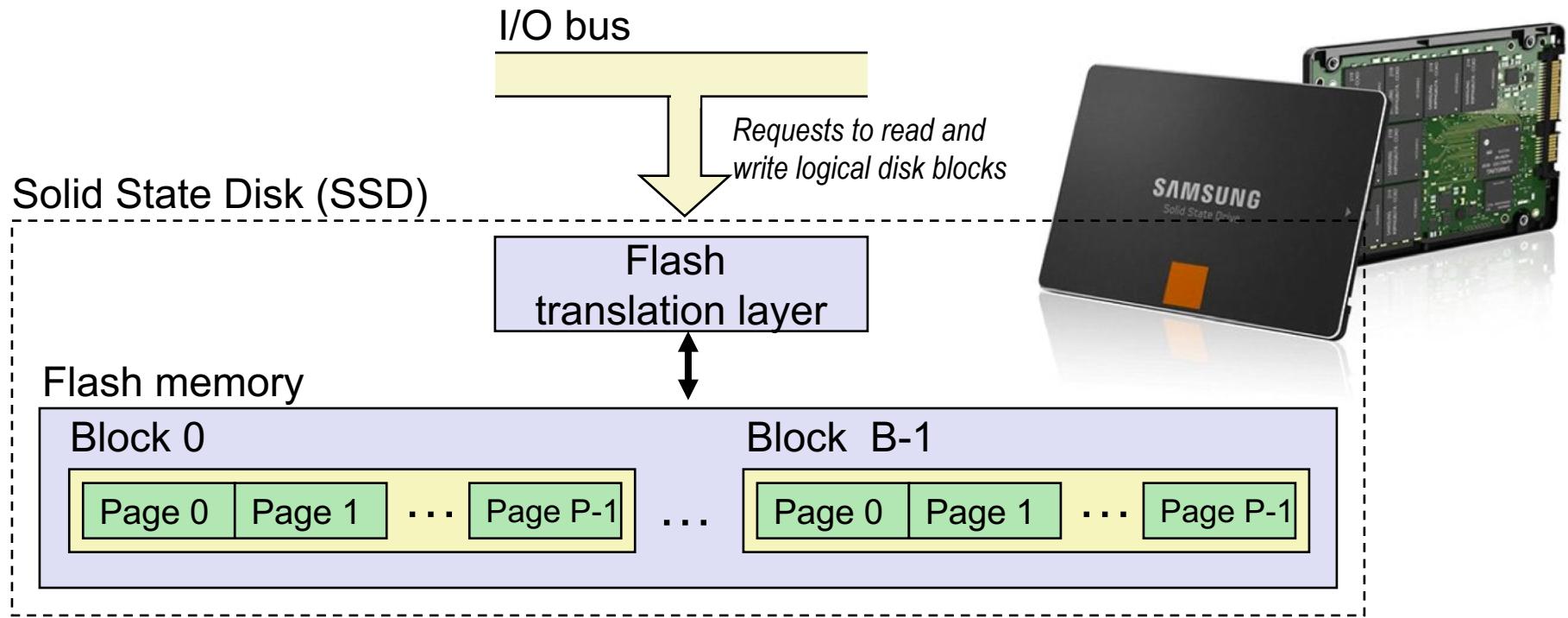
Reading a Disk Sector (3)

CPU chip



When the DMA transfer completes, the disk controller notifies the CPU with an *interrupt* (i.e., asserts a special “interrupt” pin on the CPU)

Solid State Disks (SSDs)



- **Pages: 512KB to 4KB, Blocks: 32 to 128 pages**
- **Data read/written in units of pages.**
- **Page can be written only after its block has been erased**
- **A block wears out after about 100,000 repeated writes.**

SSD Performance Characteristics

Sequential read tput	550 MB/s	Sequential write tput	470 MB/s
Random read tput	365 MB/s	Random write tput	303 MB/s
Avg seq read time	50 us	Avg seq write time	60 us

- **Sequential access faster than random access**
 - Common theme in the memory hierarchy
- **Random writes are somewhat slower**
 - Erasing a block takes a long time (~1 ms)
 - Modifying a block page requires all other pages to be copied to new block
 - In earlier SSDs, the read/write gap was much larger.

Source: Intel SSD 730 product specification.

SSD Tradeoffs vs Rotating Disks

■ Advantages

- No moving parts → faster, less power, more rugged

■ Disadvantages

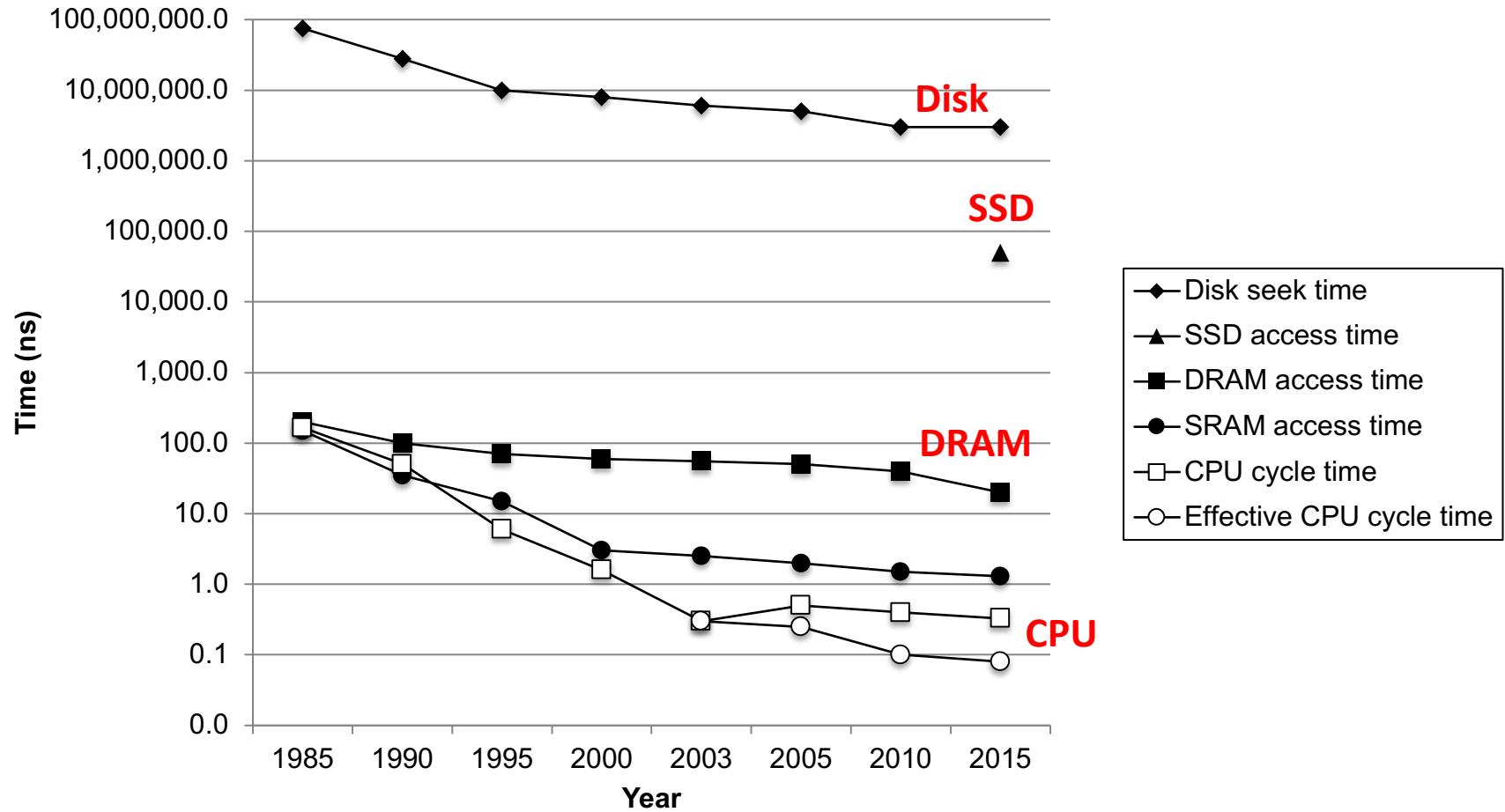
- Have the potential to wear out
 - Mitigated by “wear leveling logic” in flash translation layer
 - E.g. Intel SSD 730 guarantees 128 petabyte (128×10^{15} bytes) of writes before they wear out
- In 2015, about 30 times more expensive per byte

■ Applications

- MP3 players, smart phones, laptops
- Beginning to appear in desktops and servers

The CPU-Memory Gap

The gap widens between DRAM, disk, and CPU speeds.



Locality to the Rescue!

The key to bridging this CPU-Memory gap is a fundamental property of computer programs known as **locality**