15-213

"The course that gives CMU its Zip!"

Disk-based Storage
Oct. 23, 2008

Topics

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

Announcements

Exam next Thursday

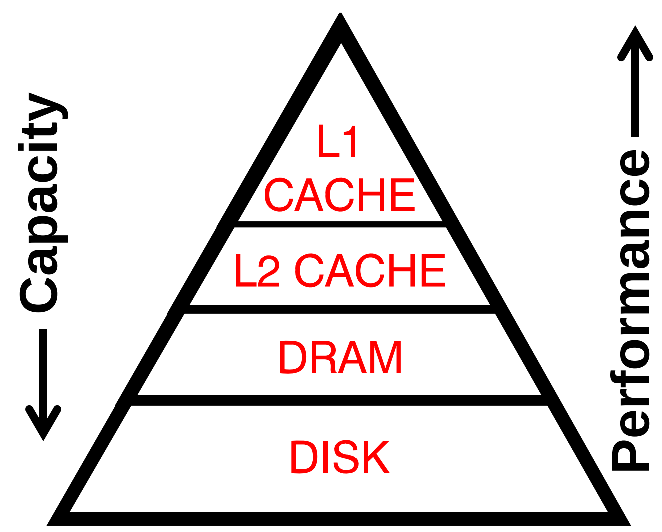
- style like exam #1: in class, open book/notes, no electronics

Disk-based storage in computers

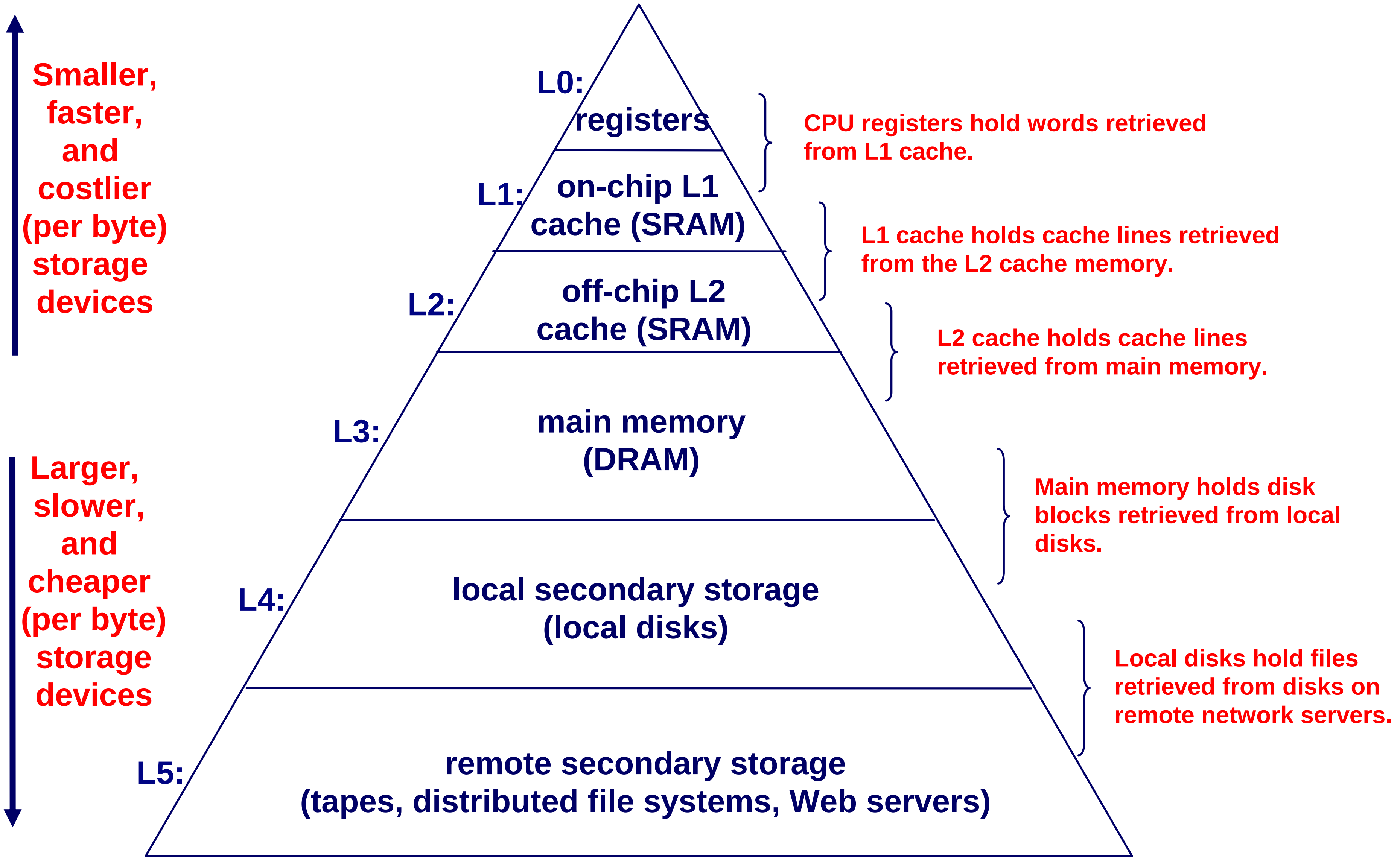
- **Memory/storage hierarchy**
  - **Combining many technologies to balance costs/benefits**
  - **Recall the memory hierarchy and virtual memory lectures**

Memory/storage hierarchies

- **Balancing performance with cost**
  - **Small memories are fast but expensive**
  - **Large memories are slow but cheap**
- **Exploit locality to get the best of both worlds**
  - **locality = re-use/nearness of accesses**
  - **allows most accesses to use small, fast memory**



An Example Memory Hierarchy

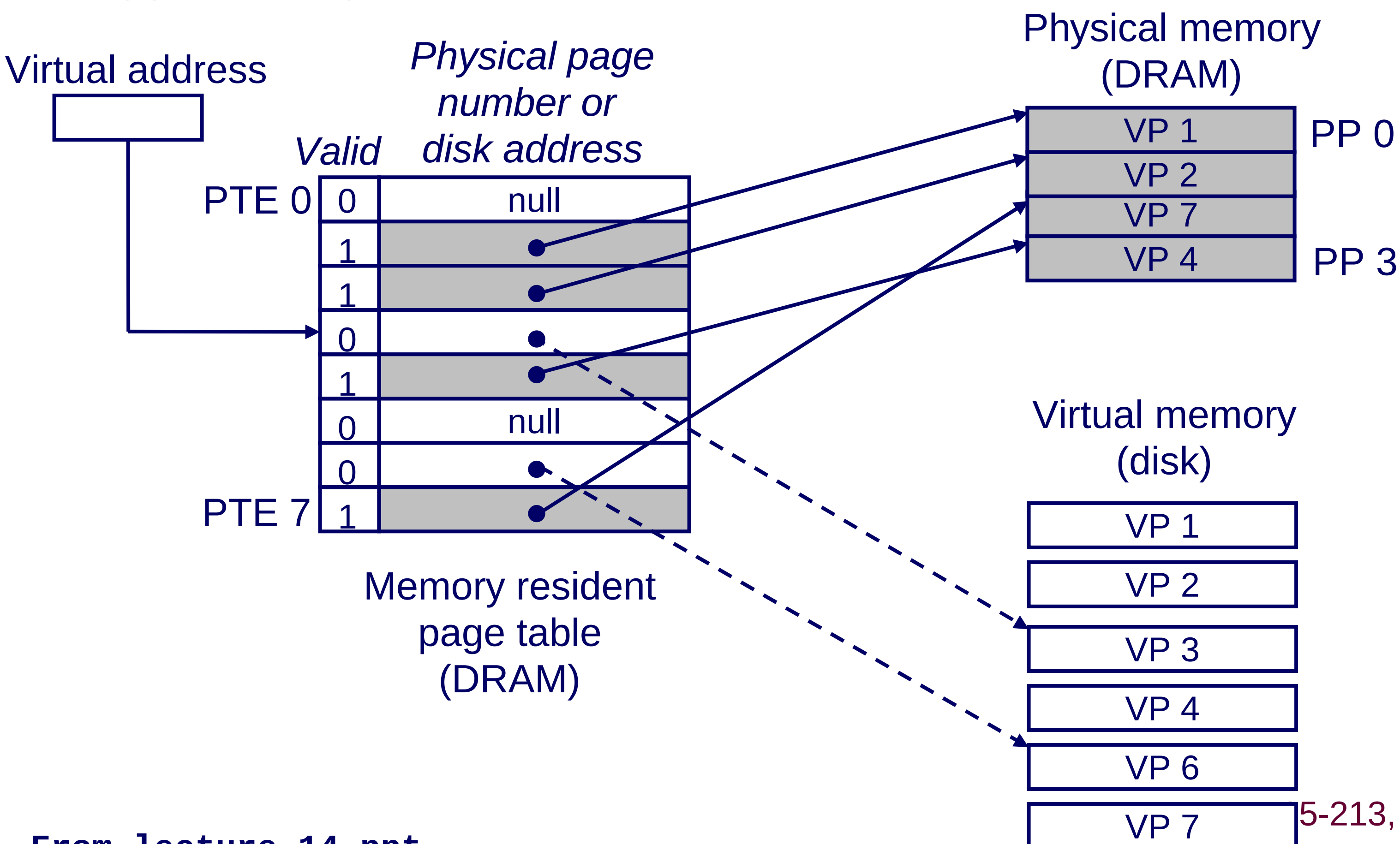


`From lecture-9.ppt`

Page Faults

A *page fault* is caused by a reference to a VM word that is not in physical (main) memory

- **Example: An instruction references a word contained in VP 3, a miss that triggers a page fault exception**



`From lecture-14.ppt`

Disk-based storage in computers

- **Memory/storage hierarchy**
  - Combining many technologies to balance costs/benefits
  - Recall the memory hierarchy and virtual memory lectures

- **Persistence**
  - Storing data for lengthy periods of time
    - DRAM/SRAM is "volatile": contents lost if power lost
    - Disks are "non-volatile": contents survive power outages
  - To be useful, it must also be possible to find it again later
    - this brings in many interesting data organization, consistency, and management issues
      - take 18-746/15-746 Storage Systems ☺
    - we'll talk a bit about file systems next

What's Inside A Disk Drive?

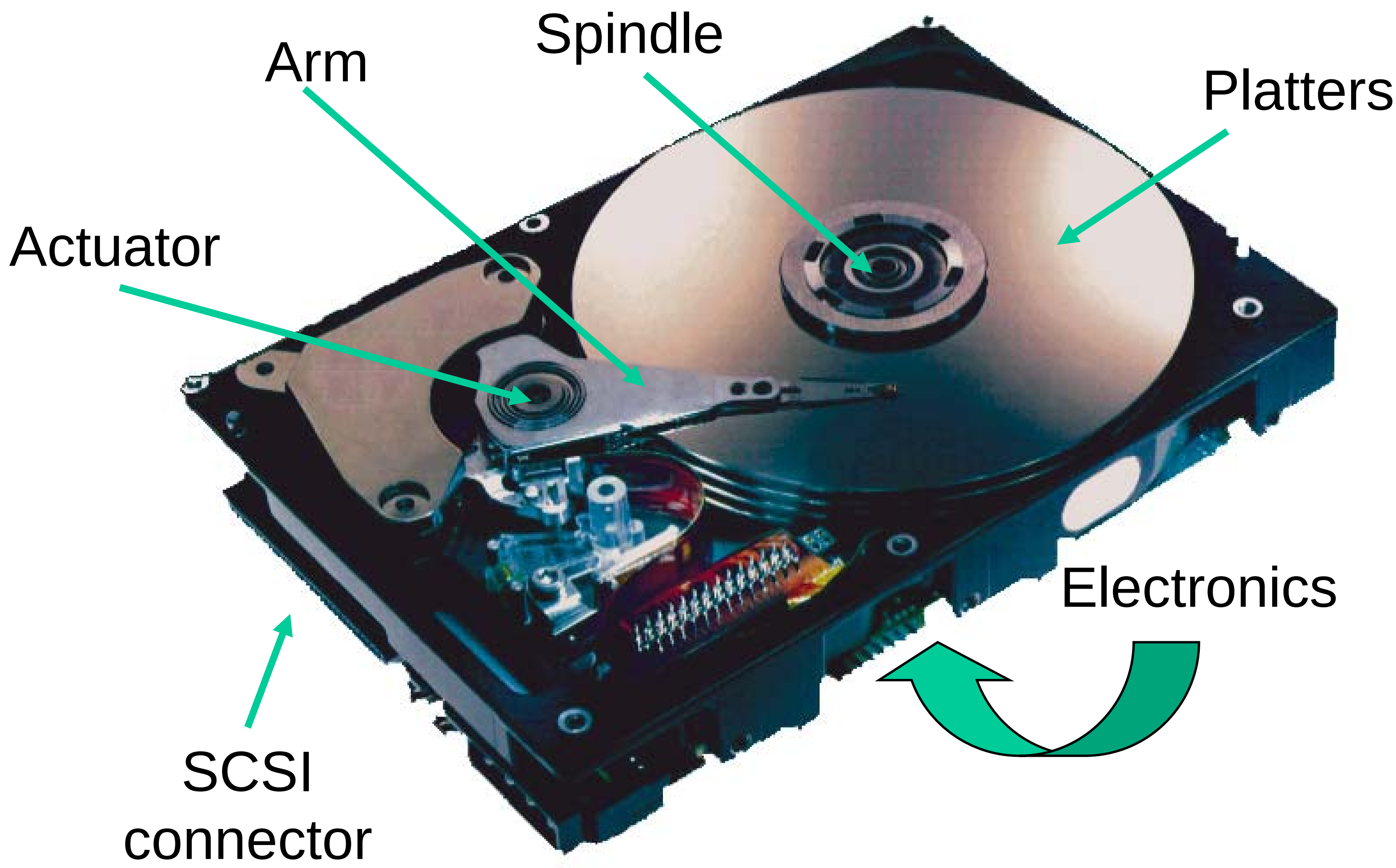
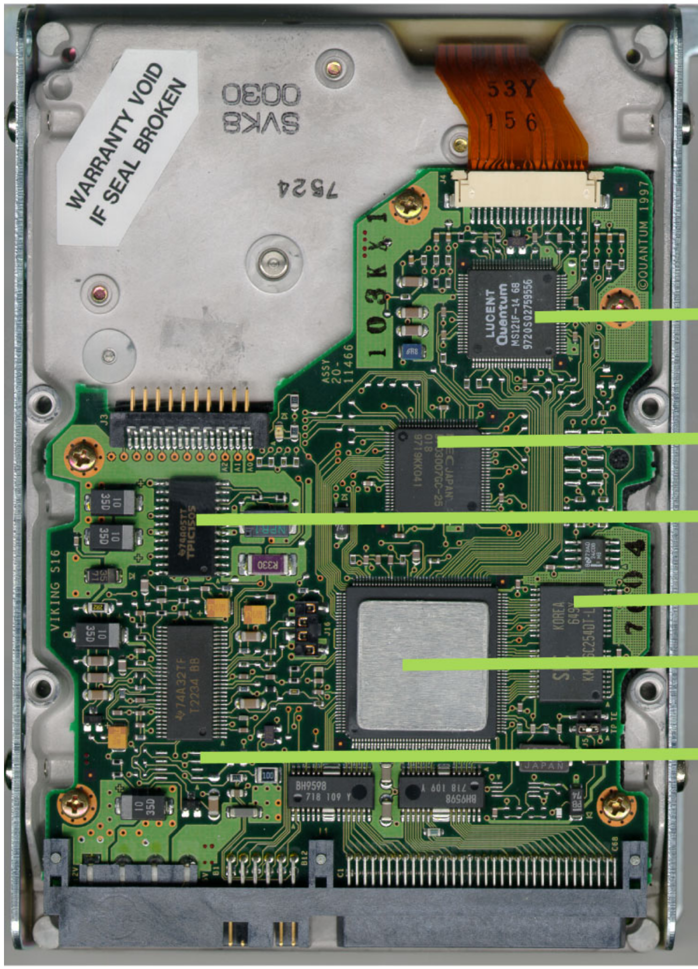


Image courtesy of Seagate Technology

Disk Electronics

Quantum Viking (circa 1997)



6 Chips

R/W Channel

uProcessor
32-bit, 25 MHz

Power Array

2 MB DRAM

Control ASIC
SCSI, servo, ECC

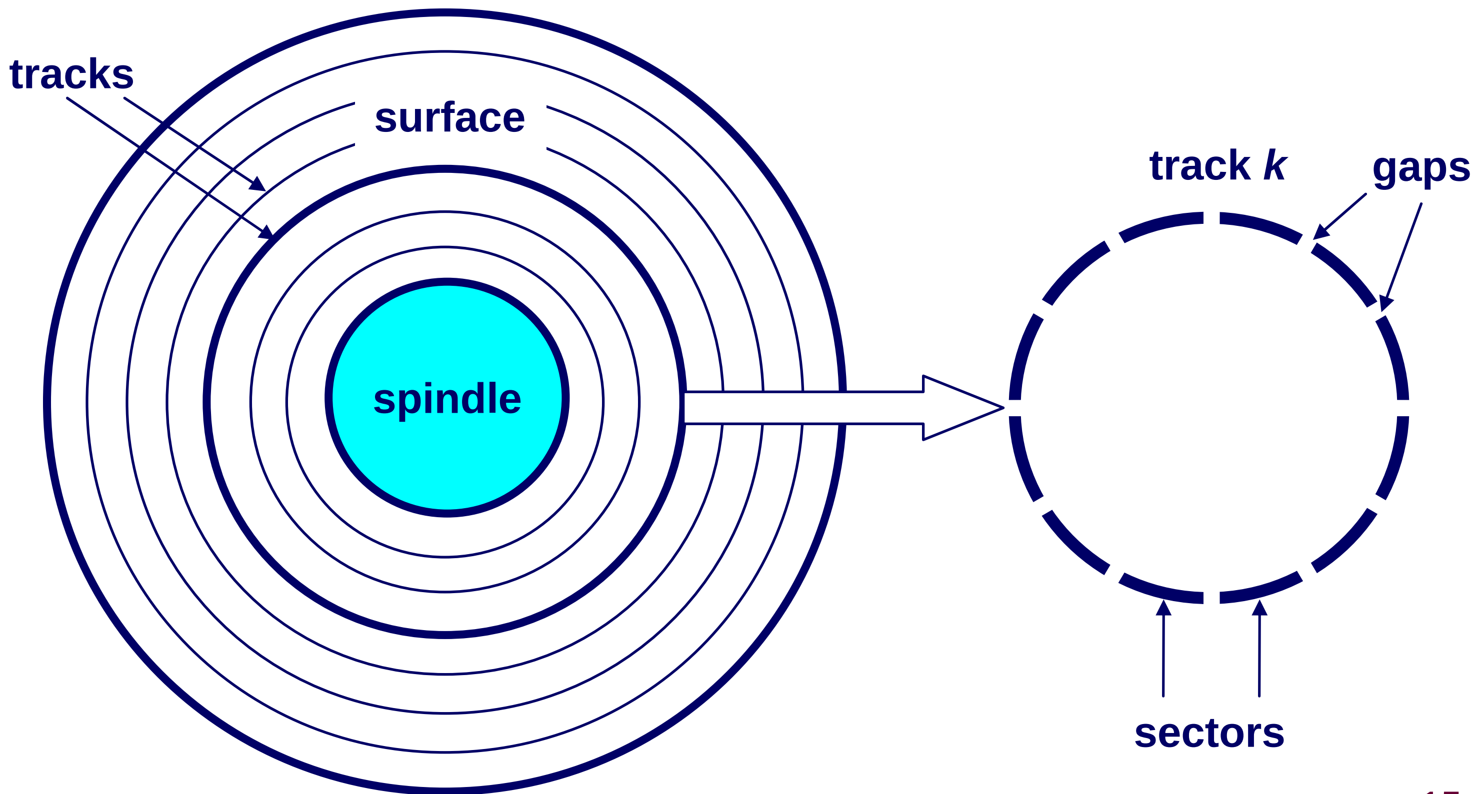
Motor/Spindle

Just like a small computer – processor, memory, network iface

- Connect to disk
- Control processor
- Cache memory
- Control ASIC
- Connect to motor

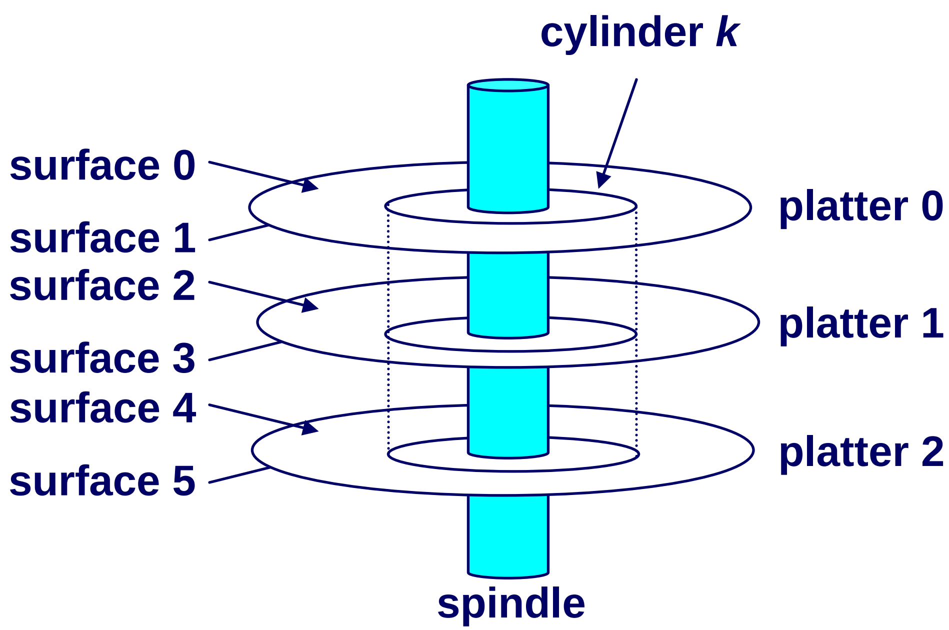
Disk "Geometry"

- Disks contain **platters**, each with two **surfaces**
- Each surface organized in concentric rings called **tracks**
- Each track consists of **sectors** separated by **gaps**

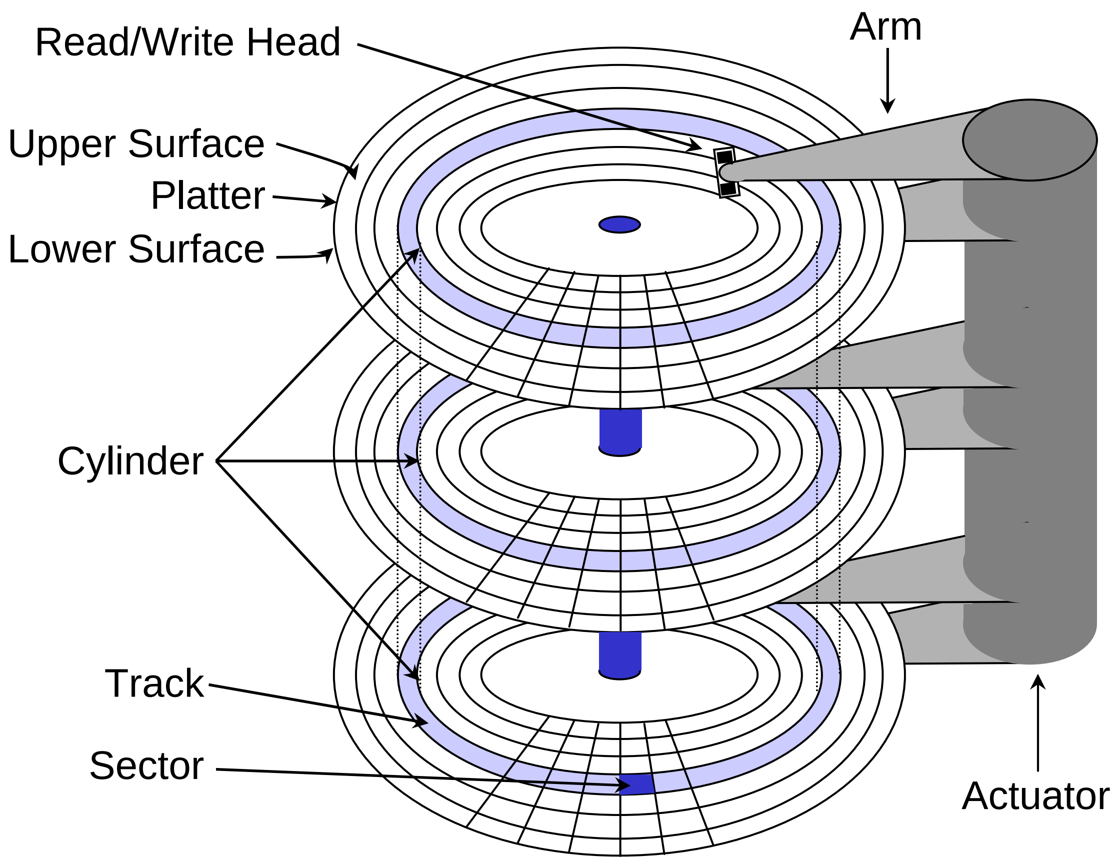


Disk Geometry (Muliple-Platter View)

Aligned tracks form a cylinder

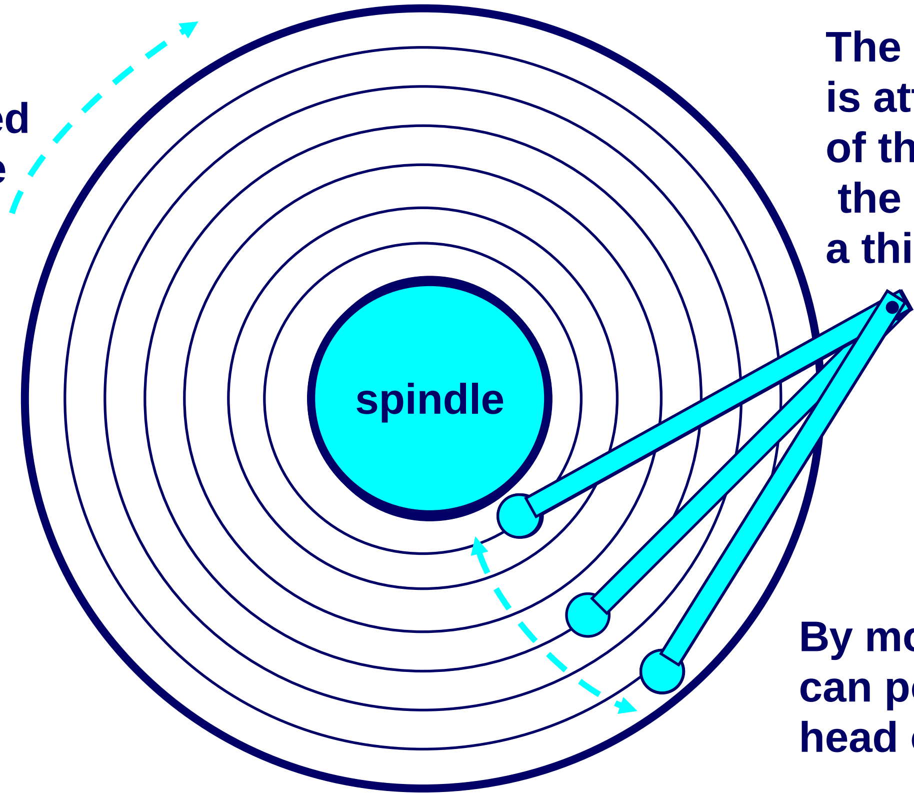


Disk Structure



Disk Operation (Single-Platter View)

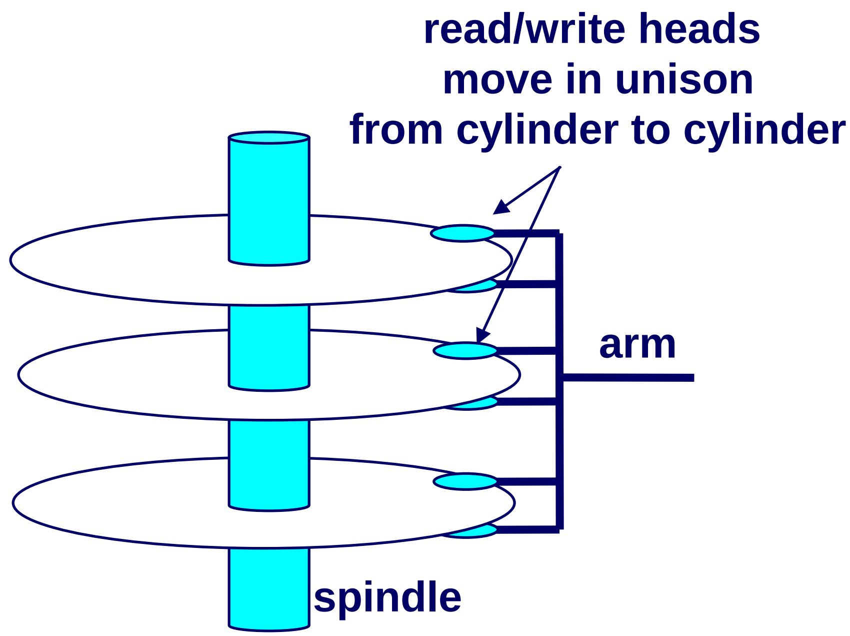
The disk surface spins at a fixed rotational rate



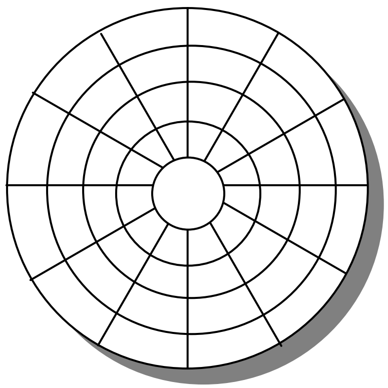
The read/write *head* is attached to the end of the *arm* and flies over the disk surface on a thin cushion of air

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

Disk Operation (Multi-Platter View)



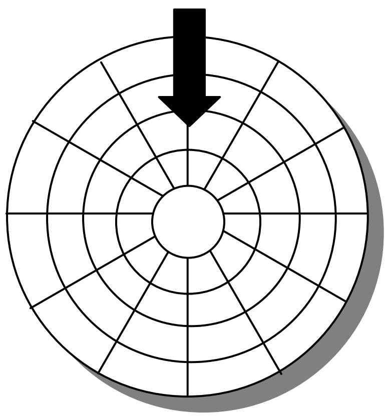
Disk Structure - top view of single platter



Surface organized into tracks

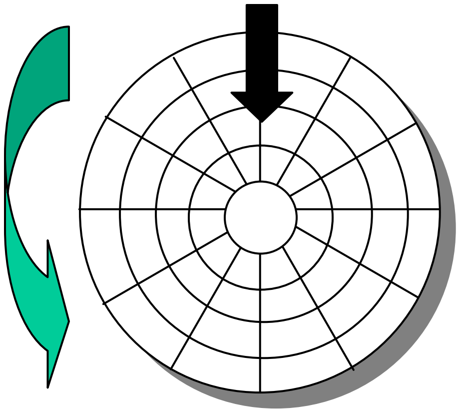
Tracks divided into sectors

Disk Access



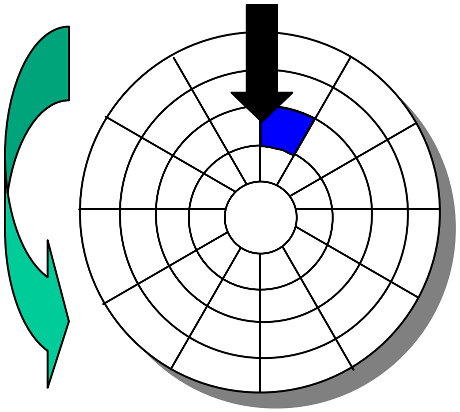
Head in position above a track

Disk Access



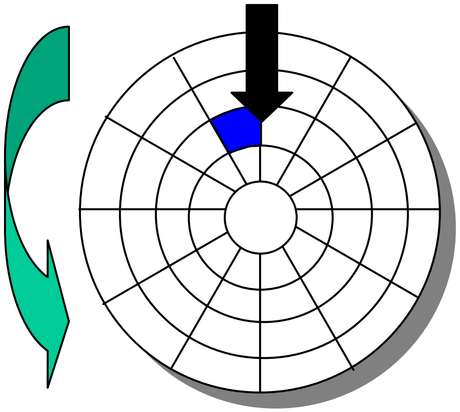
Rotation is counter-clockwise

Disk Access – Read



About to read blue sector

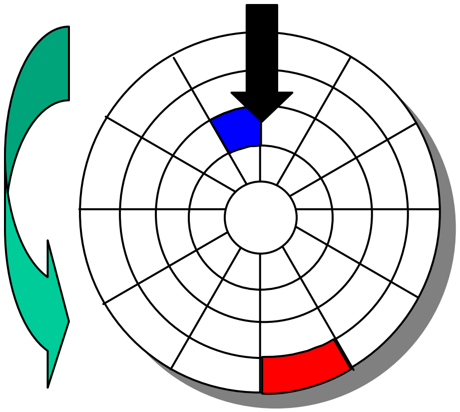
Disk Access – Read



After BLUE read

After reading blue sector

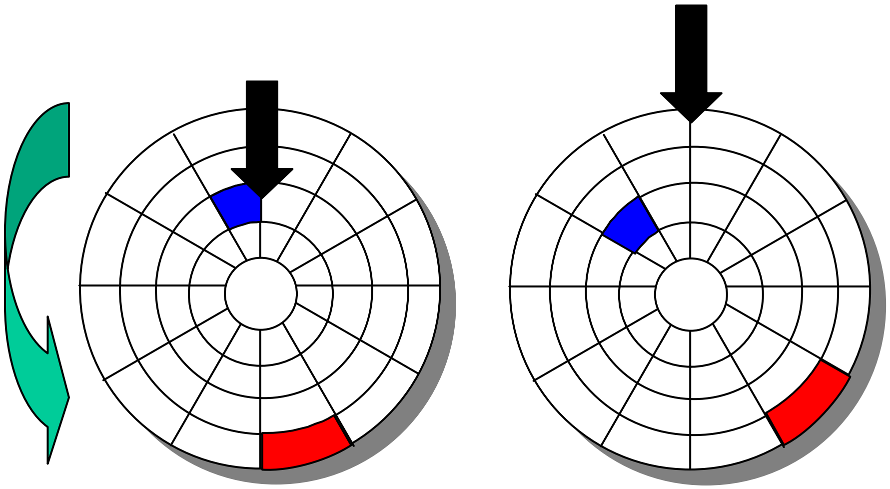
Disk Access – Read



After BLUE read

Red request scheduled next

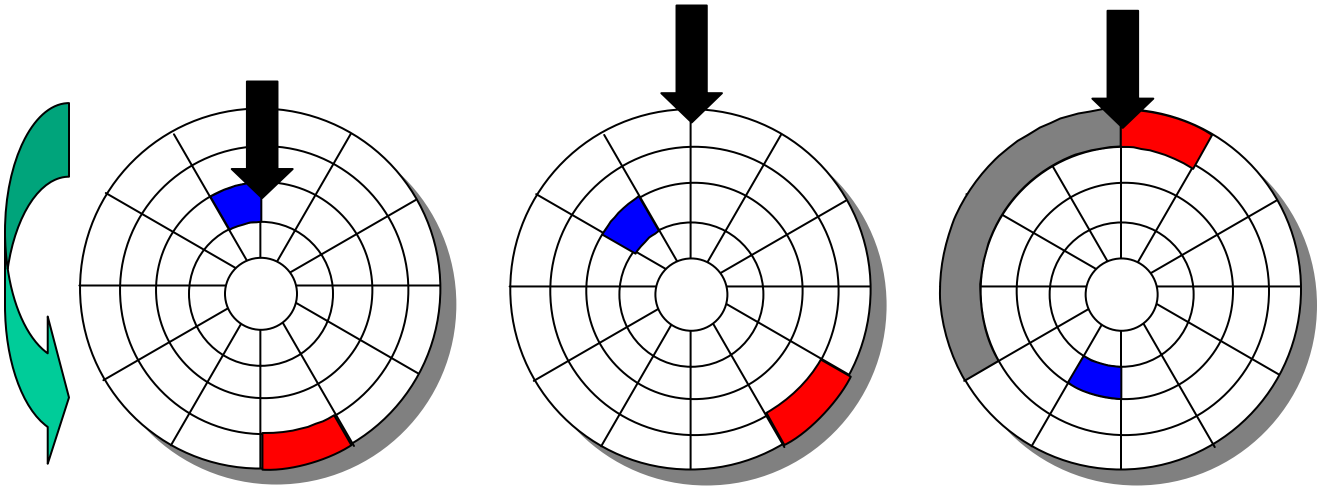
Disk Access – Seek



After BLUE read Seek for RED

Seek to red's track

Disk Access – Rotational Latency



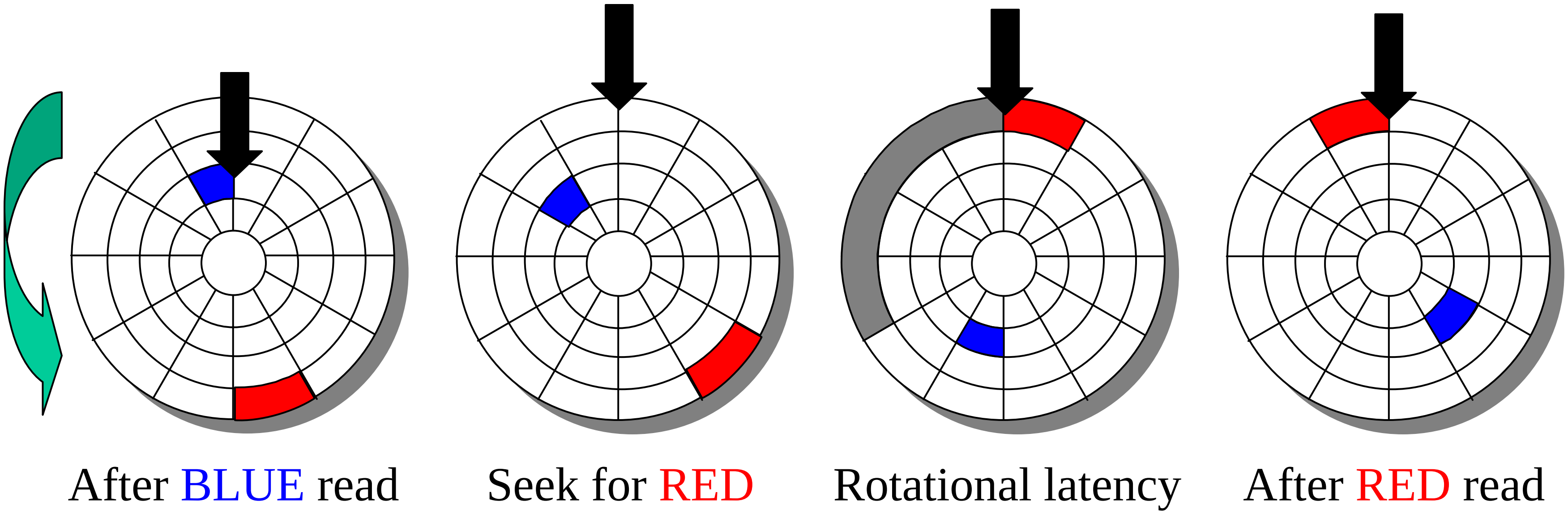
After BLUE read

Seek for RED

Rotational latency

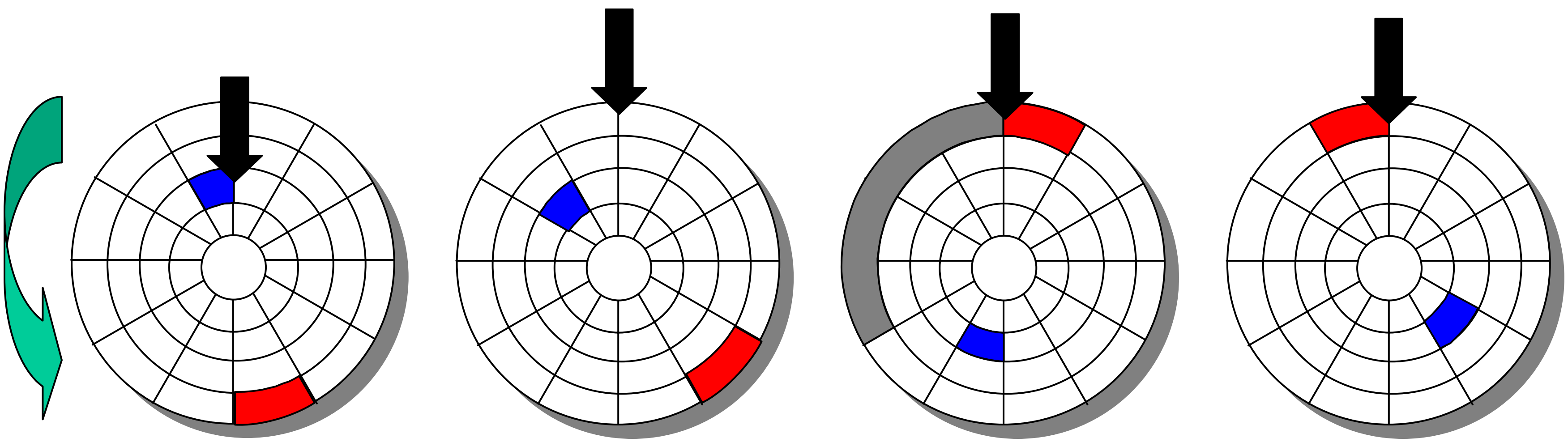
Wait for red sector to rotate around

Disk Access – Read



Complete read of red

Disk Access – Service Time Components



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

Seek
Rotational Latency
Data Transfer

Disk Access Time

Average time to access a specific sector approximated by:

- Taccess = Tavg seek + Tavg rotation + Tavg transfer

Seek time (Tavg seek)

- Time to position heads over cylinder containing target sector
- Typical Tavg seek = 3-5 ms

Rotational latency (Tavg rotation)

- Time waiting for first bit of target sector to pass under r/w head
- Tavg rotation = 1/2 x 1/RPMs x 60 sec/1 min
  - e.g., 3ms for 10,000 RPM disk

Transfer time (Tavg transfer)

- Time to read the bits in the target sector
- Tavg transfer = 1/RPM x 1/(avg # sectors/track) x 60 secs/1 min
  - e.g., 0.006ms for 10,000 RPM disk with 1,000 sectors/track
  - given 512-byte sectors, ~85 MB/s data transfer rate

Disk Access Time Example

Given:

- Rotational rate = 7,200 RPM
- Average seek time = 5 ms
- Avg # sectors/track = 1000

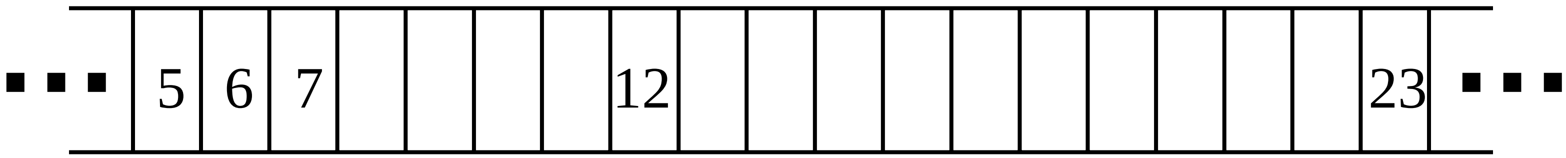
Derived average time to access random sector:

- Tavg rotation = 1/2 x (60 secs/7200 RPM) x 1000 ms/sec = 4 ms
- Tavg transfer = 60/7200 RPM x 1/400 secs/track x 1000 ms/sec = 0.008 ms
- Taccess = 5 ms + 4 ms + 0.008 ms = 9.008 ms
  - Time to second sector: 0.008 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
  - ~100,000 times longer to access a word on disk than in DRAM

Disk storage as array of blocks

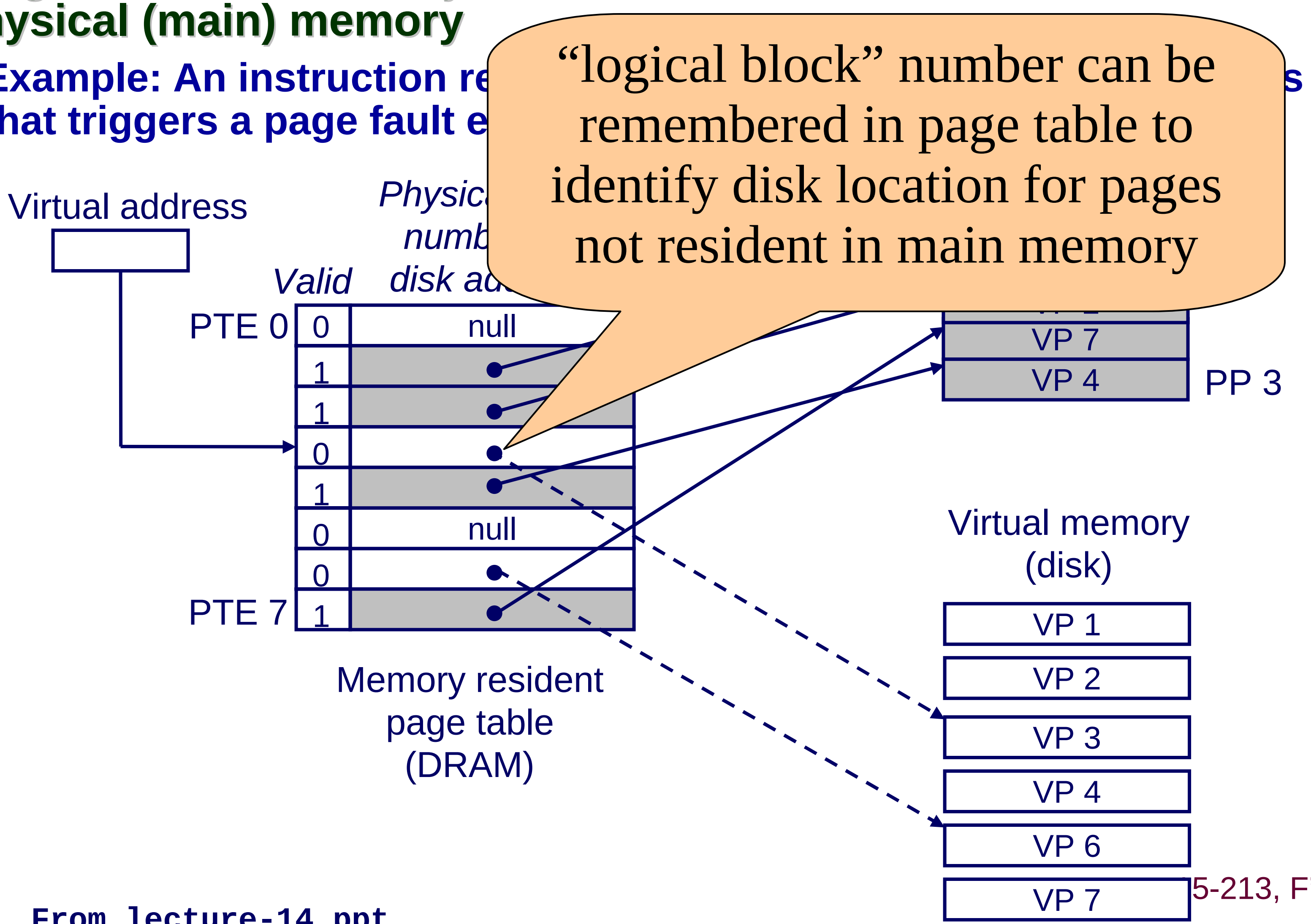


OS's view of storage device
(as exposed by SCSI or IDE/ATA protocols)

- **Common "logical block" size: 512 bytes**
- **Number of blocks: device capacity / block size**
- **Common OS-to-storage requests defined by few fields**
  - **R/W, block #, # of blocks, memory source/dest**

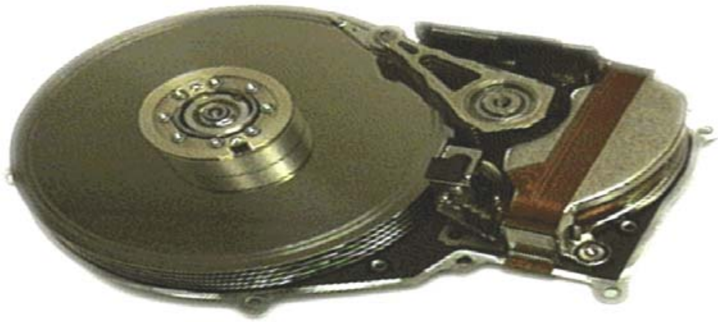
Page Faults

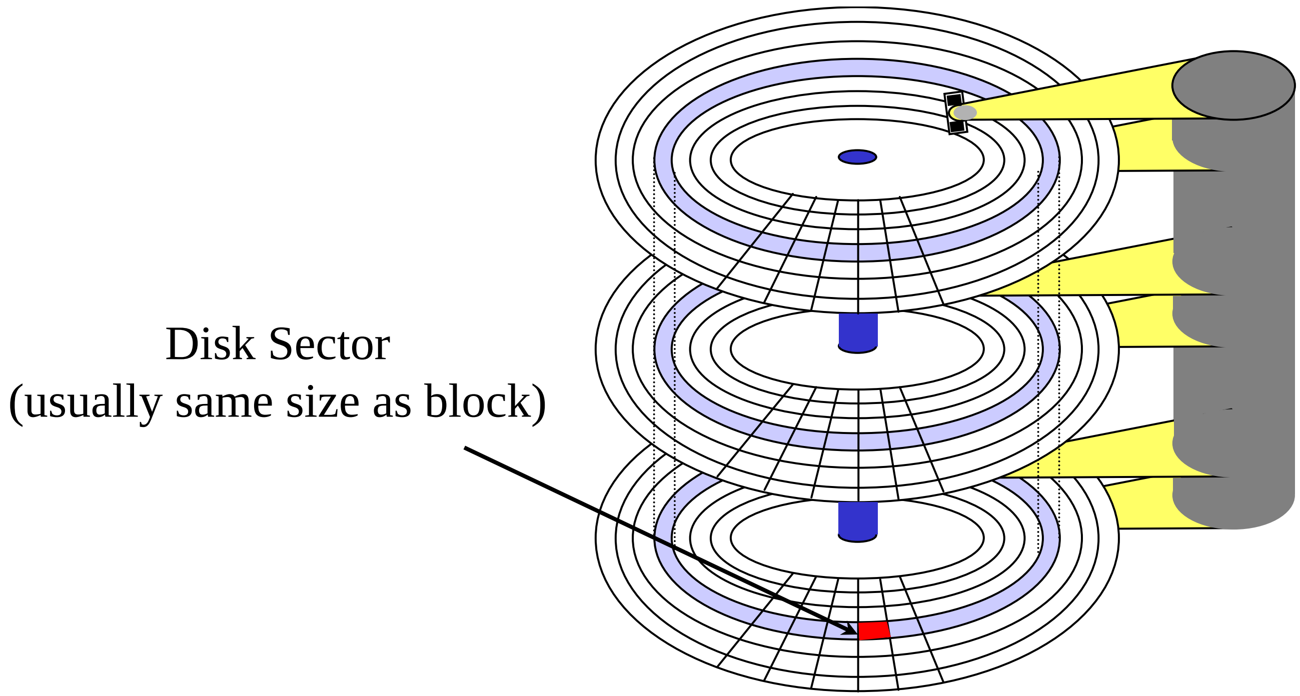
- **A *page fault* is caused by a reference to a VM word that is not in physical (main) memory**
  - **Example: An instruction re... that triggers a page fault e...**



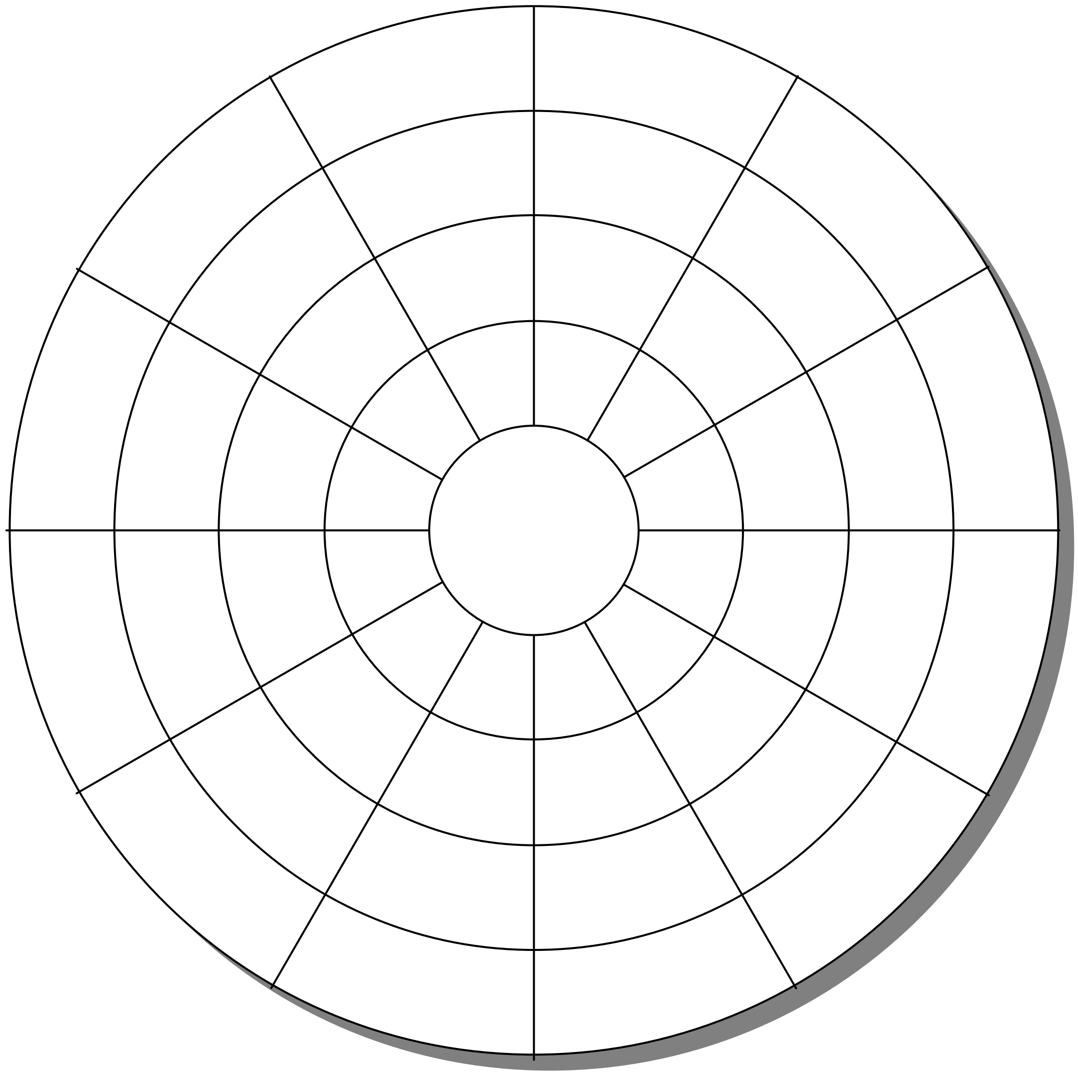
From lecture-14.ppt

In device, “blocks” mapped to physical store

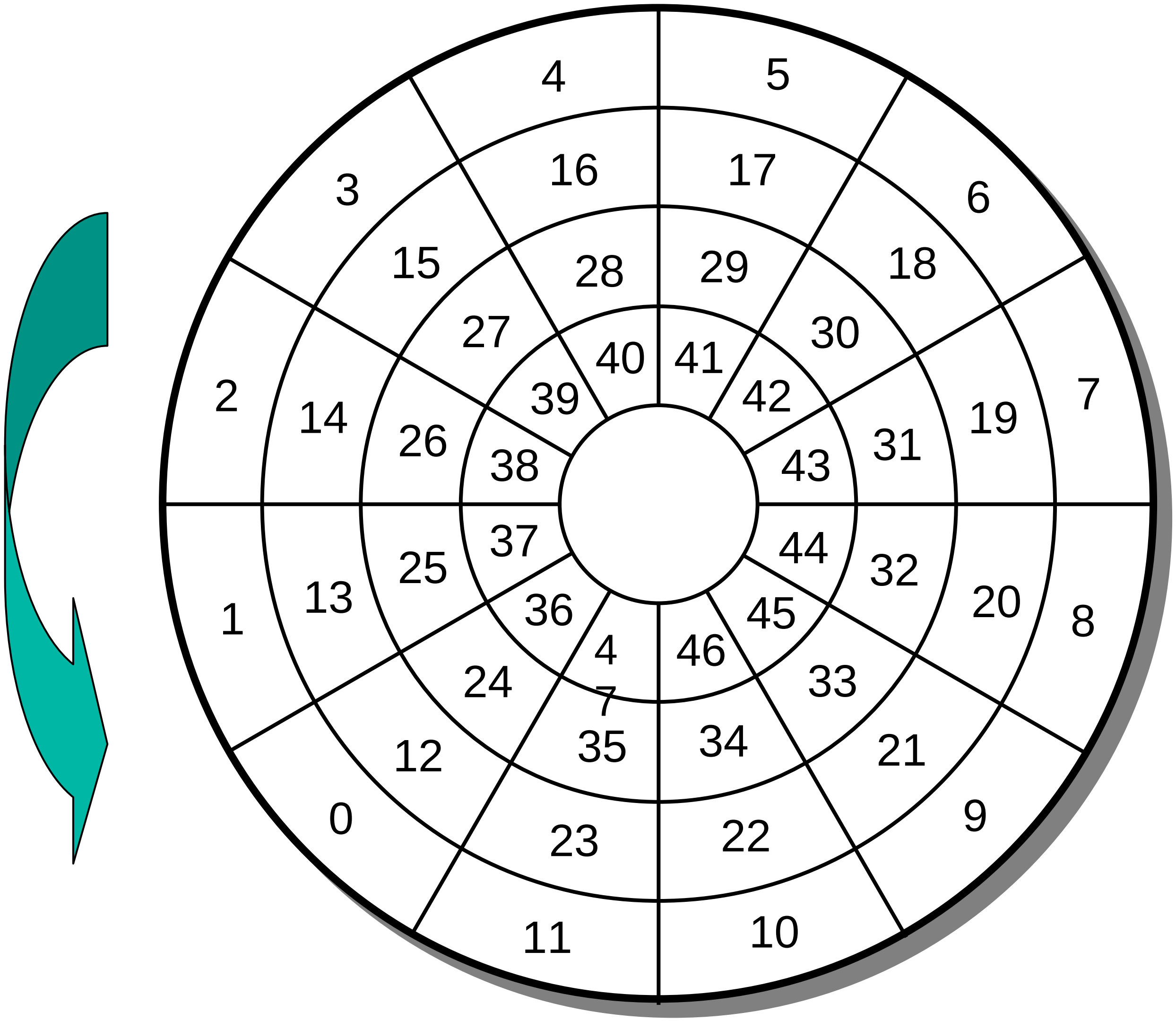




Physical sectors of a single-surface disk



LBN-to-physical for a single-surface disk



Disk Capacity

Capacity: **maximum number of bits that can be stored**

- **Vendors express capacity in units of gigabytes (GB), where $1\text{ GB} = 10^9$ Bytes (Lawsuit pending! Claims deceptive advertising)**

Capacity is determined by these technology factors:

- **Recording density (bits/in): number of bits that can be squeezed into a 1 inch linear segment of a track**
- **Track density (tracks/in): number of tracks that can be squeezed into a 1 inch radial segment**
- **Areal density (bits/in^2): product of recording and track density**

Computing Disk Capacity

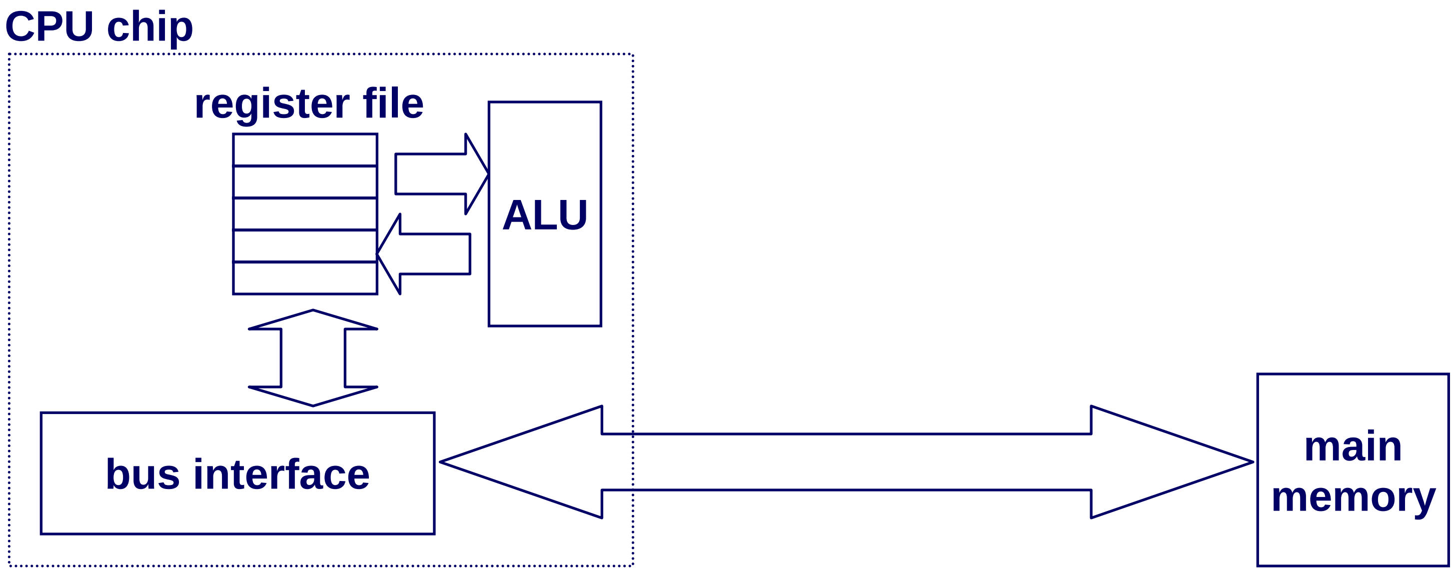
Capacity = (# bytes/sector) x (avg. # sectors/track) x
(# tracks/surface) x (# surfaces/platter) x
(# platters/disk)

Example:

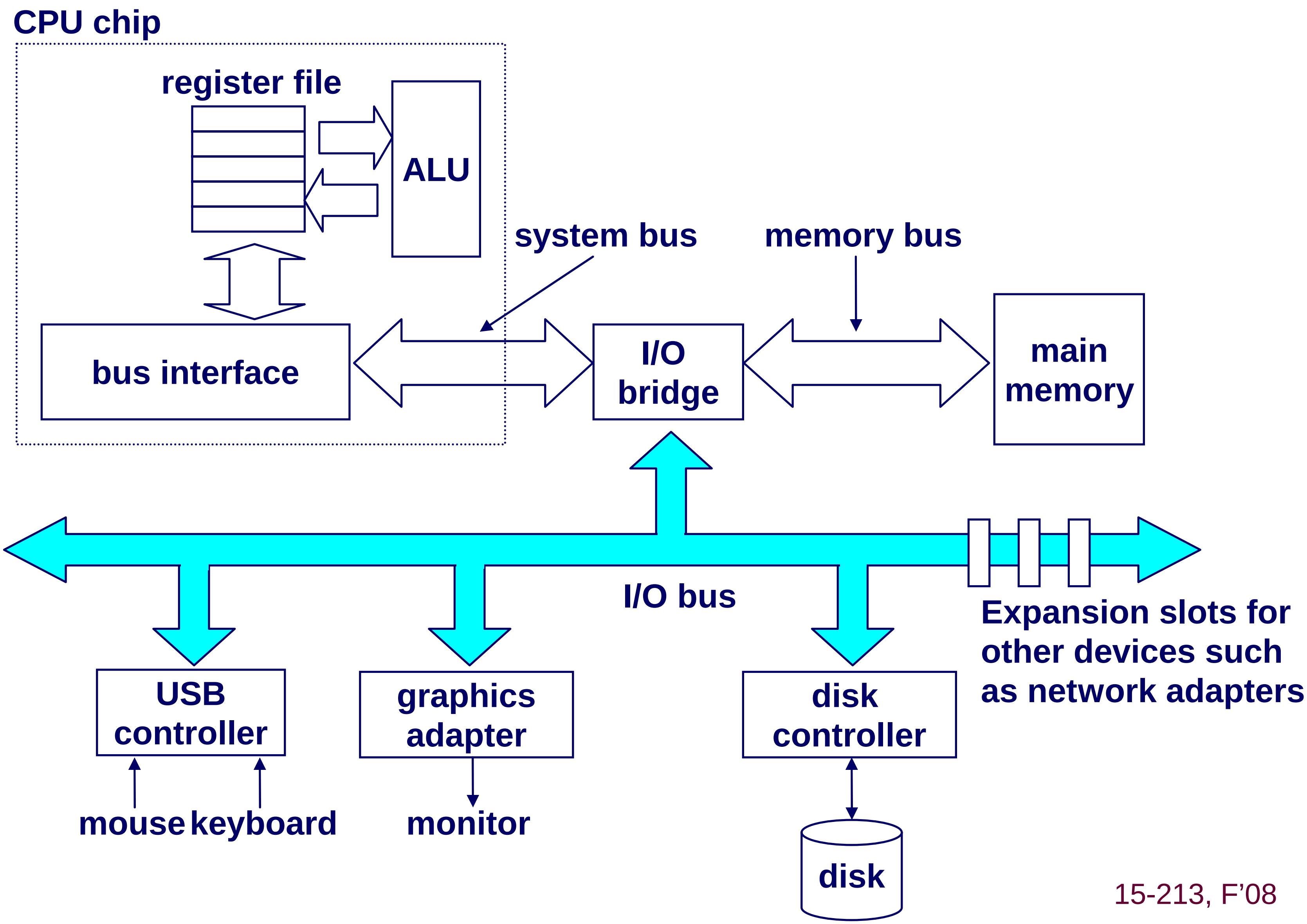
- **512 bytes/sector**
- **1000 sectors/track (on average)**
- **20,000 tracks/surface**
- **2 surfaces/platter**
- **5 platters/disk**

Capacity = 512 x 1000 x 80000 x 2 x 5
= 409,600,000,000
= 409.6 GB

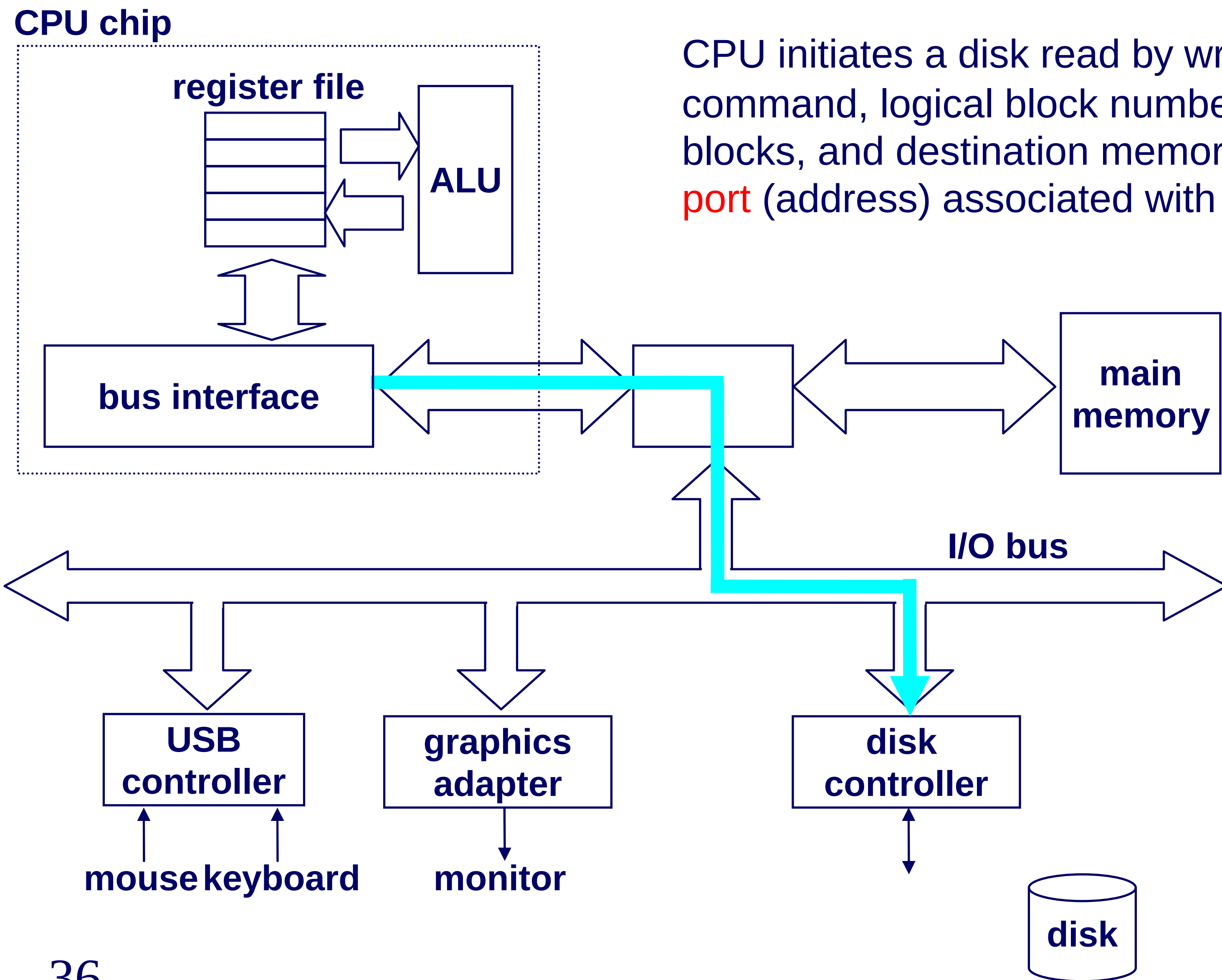
Looking back at the hardware



Connecting I/O devices: the I/O Bus

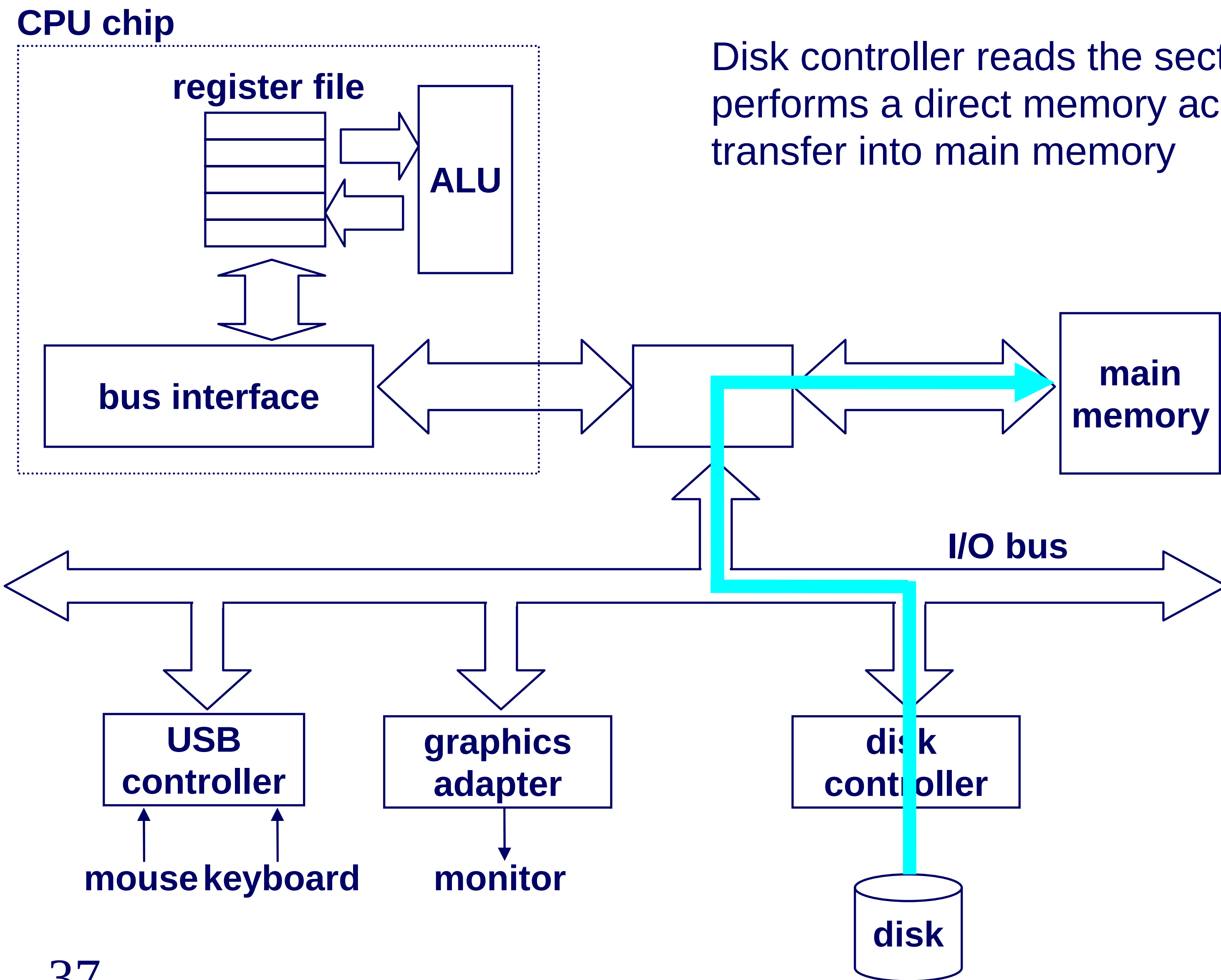


Reading from disk (1)



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

Reading from disk (2)



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

Reading from disk (3)

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