Computer Organization and Software Systems

CONTACT SESSION 2

PERFORMANCE ASSESSMENT & MEMORY ORGANISATION

Prof. C R Sarma
WILP.BITS-PILANI

BITS Pilani
Pilani Campus

Today's Class

Contact Hour	List of Topic Title	Text/Ref Book/external resource
3	**Performance Assessment** 1.5.1. MIPS Rate 1.5.2 Amdahl's Law	Class Slides
4	**Memory Organization** Storage Technologies Random Access Memory Disk Storage Solid State Disks Storage Technology Trends Locality Locality of Reference to Program Data Locality of instruction fetches Memory Hierarchy	T1, R2

Performance Assessment

BITS Pilani
Pilani Campus

Units

- Kilo- (K) = 1 thousand = 10^3 and 2^{10}
- Mega- (M) = 1 million = 10^6 and 2^{20}
- Giga- (G) = 1 billion = 10^9 and 2^{30}
- Tera- (T) = 1 trillion = 10^{12} and 2^{40}
- Peta- (P) = 1 quadrillion = 10^{15} and 2^{50}
- Exa – (E) = 1 quintillion = 10^{18} and 2^{60}

Examples

Hertz = clock cycles per second (frequency)

- 1MHz = 1,000,000Hz
- Processor speeds are measured in MHz or GHz.

Byte = a unit of storage

- 1KB = 2^{10} = 1024 Bytes
- 1MB = 2^{20} = 1,048,576 Bytes
- Main memory (RAM) is measured in MB / GB
- Disk storage is measured in GB for small systems, TB for large systems.

Units…

- Milli- (m) = 1 thousandth = 10^{-3}
- Micro- (μ) = 1 millionth = 10^{-6}
- Nano- (n) = 1 billionth = 10^{-9}
- Pico- (p) = 1 trillionth = 10^{-12}
- Femto- (f) = 1 quadrillionth = 10^{-15}

Examples

- Millisecond = 1 thousandth of a second
  - Hard disk drive access times are often 10 to 20 milliseconds.
- Nanosecond = 1 billionth of a second
  - Main memory access times are often 50 to 70 nanoseconds.
- Micron (micrometer) = 1 millionth of a meter
  - Circuits on computer chips are measured in microns.

Important Terms

- **Execution time** : The total time required for the computer to complete a task, including disk accesses, memory accesses, I/O activities, operating system overhead, CPU execution
- **Throughput or bandwidth** :number of tasks completed per unit time.

Example

Do the following changes to a computer system increase throughput, decrease execution time, or both?

1. Replacing the processor in a computer with a faster version
2. Adding additional processors to a system that uses multiple processors for separate tasks

Contd…

- Relationship between Performance and execution time of Computer X

$$\text{Performance}_X = \frac{1}{\text{Execution time}_X}$$

- if the performance of X is greater than the performance of Y, we have

$$\text{Performance}_X > \text{Performance}_Y$$

$$\frac{1}{\text{Execution time}_X} > \frac{1}{\text{Execution time}_Y}$$

$$\text{Execution time}_Y > \text{Execution time}_X$$

Contd…

- Quantitative performance analysis
  - Computer X is “n” times faster than Computer Y

$$\frac{\text{Performance}_X}{\text{Performance}_Y} = n$$

- If X is *n* times faster than Y, then the execution time on Y is *n* times longer than it is on X:

$$\frac{\text{Performance}_X}{\text{Performance}_Y} = \frac{\text{Execution time}_Y}{\text{Execution time}_X} = n$$

Example

- If computer A runs a program in 10 seconds and computer B runs the same program in 15 seconds, how much faster is A than B?

$$\frac{\text{Performance}_A}{\text{Performance}_B} = \frac{\text{Execution time}_B}{\text{Execution time}_A} = n$$

- Computer A is therefore 1.5 times faster than B.

CPU performance and its factors

- CPU execution time for a program:

$$\frac{\text{CPU execution time}}{\text{for a program}} = \frac{\text{CPU clock cycles}}{\text{for a program}} \times \text{Clock cycle time}$$

$$\frac{\text{CPU execution time}}{\text{for a program}} = \frac{\text{CPU clock cycles for a program}}{\text{Clock rate}}$$

Example

- Our favorite program runs in 10 seconds on computer A, which has a 2 GHz clock. We are trying to help a computer designer build a computer, B, which will run this program in 6 seconds. The designer has determined that a substantial increase in the clock rate is possible, but this increase will affect the rest of the CPU design, causing computer B to require 1.2 times as many clock cycles as computer A for this program. What clock rate should we tell the designer to target?

$$\begin{array}{c}\text{CPU execution time}\\ \text{for a program}\end{array} = \frac{\text{CPU clock cycles for a program}}{\text{Clock rate}}$$

Let's first find the number of clock cycles required for the program on A:

$$\text{CPU time}_A = \frac{\text{CPU clock cycles}_A}{\text{Clock rate}_A}$$

$$10 \text{ seconds} = \frac{\text{CPU clock cycles}_A}{2 \times 10^9 \ \dfrac{\text{cycles}}{\text{second}}}$$

$$\text{CPU clock cycles}_A = 10 \text{ seconds} \times 2 \times 10^9 \ \frac{\text{cycles}}{\text{second}} = 20 \times 10^9 \text{ cycles}$$

CPU time for B can be found using this equation:

$$\text{CPU time}_B = \frac{1.2 \times \text{CPU clock cycles}_A}{\text{Clock rate}_B}$$

$$6 \text{ seconds} = \frac{1.2 \times 20 \times 10^9 \text{ cycles}}{\text{Clock rate}_B}$$

$$\text{Clock rate}_B = \frac{1.2 \times 20 \times 10^9 \text{ cycles}}{6 \text{ seconds}} = \frac{0.2 \times 20 \times 10^9 \text{ cycles}}{\text{second}} = \frac{4 \times 10^9 \text{ cycles}}{\text{second}} = 4 \text{ GHz}$$

Instruction Performance

- CPI: Clock cycles Per Instruction
  - Average number of clock cycles per instruction for a program or program fragment.

$$\text{CPU clock cycles} = \text{Instructions for a program} \times \begin{array}{c}\text{Average clock cycles}\\ \text{per instruction}\end{array}$$

Example

Computer A has a clock cycle time of 250 ps and a CPI of 2.0 for some program, and computer B has a clock cycle time of 500 ps and a CPI of 1.2 for the same program. Which computer is faster for this program and by how much?

Solution

- the number of processor clock cycles for each computer

  $\text{CPU clock cycles}_A = I \times 2.0$

  $\text{CPU clock cycles}_B = I \times 1.2$

- Execution time for each computer

  Execution time = CPU clock cycles × Clock cycle time

  $\text{Execution time}_A = I \times 2.0 \times 250 \text{ ps} = 500 \times I \text{ ps}$

  $\text{Execution time}_B = I \times 1.2 \times 500 \text{ ps} = 600 \times I \text{ ps}$

- Comparison:

$$\frac{\text{CPU performance}_A}{\text{CPU performance}_B} = \frac{\text{Execution time}_B}{\text{Execution time}_A} = \frac{600\text{ I ps}}{500\text{ I ps}} = 1.2$$

Amdahl's Law

- proposed by Gene Amdahl in 1967
- deals with the potential speedup of a program using multiple processors compared to a single processor

Example-

A program needs 20 hours using a single processor core, and a particular part of the program which takes one hour to execute cannot be parallelized.

The remaining 19 hours ($p = 0.95$) of execution time can be parallelized, then regardless of how many processors are devoted to a parallelized execution of this program, the minimum execution time cannot be less than that critical one hour.

$$1/(1-p)$$

Hence, the theoretical speedup is limited to at most 20 times

. For this reason, parallel computing with many processors is useful only for highly parallelizable programs.

Amdahl's Law

$$\text{Speedup} = \frac{\text{Performance after enhancement}}{\text{Performance before enhancement}} = \frac{\text{Execution time before enhancement}}{\text{Execution time after enhancement}}$$

S=Speedup,
f=fraction of time enhancement,
k=speedup of the faster component

$$S = \frac{1}{(1-f) + \frac{f}{k}}$$

If 90% of a program is speeded up to run 10 times faster f=0.9 and k=10
Overall speedup is 1/(1-0.9)+(0.9/10)= 1/(0.1+0.09)=1/(0.19)=5.26

Making 80% of a program run 20% faster
f=0.80 and k=1.2
1/(1-0.8)+(0.8/1.2)= 1/(0.2+0.8/1.2)=1/(0.2+0.66)=1/0.866=1.53

Example

On a large system CPU upgrade makes it faster by 50% for INR 10,000. A disk drive upgrade of INR 7000 speeds it up by 150%. Evaluate the speedups?
Generally Processes spend 70% in CPU and 30% waiting Diskdrives

Processor upgrade

$f = 0.70$, $k = 1.5$ $\quad S = \dfrac{1}{(1 - 0.7) + 0.7/1.5}$ **=1.304**

30% improvement

Disk Drive upgrade

$f = 0.30$, $k = 2.5$ $\quad S = \dfrac{1}{(1 - 0.3) + 0.3/2.5}$ **=1.219**

22% Improvement

CPU-30 % improvement -faster by 50%
---so 1% increment is INR 10000/30=INR 333

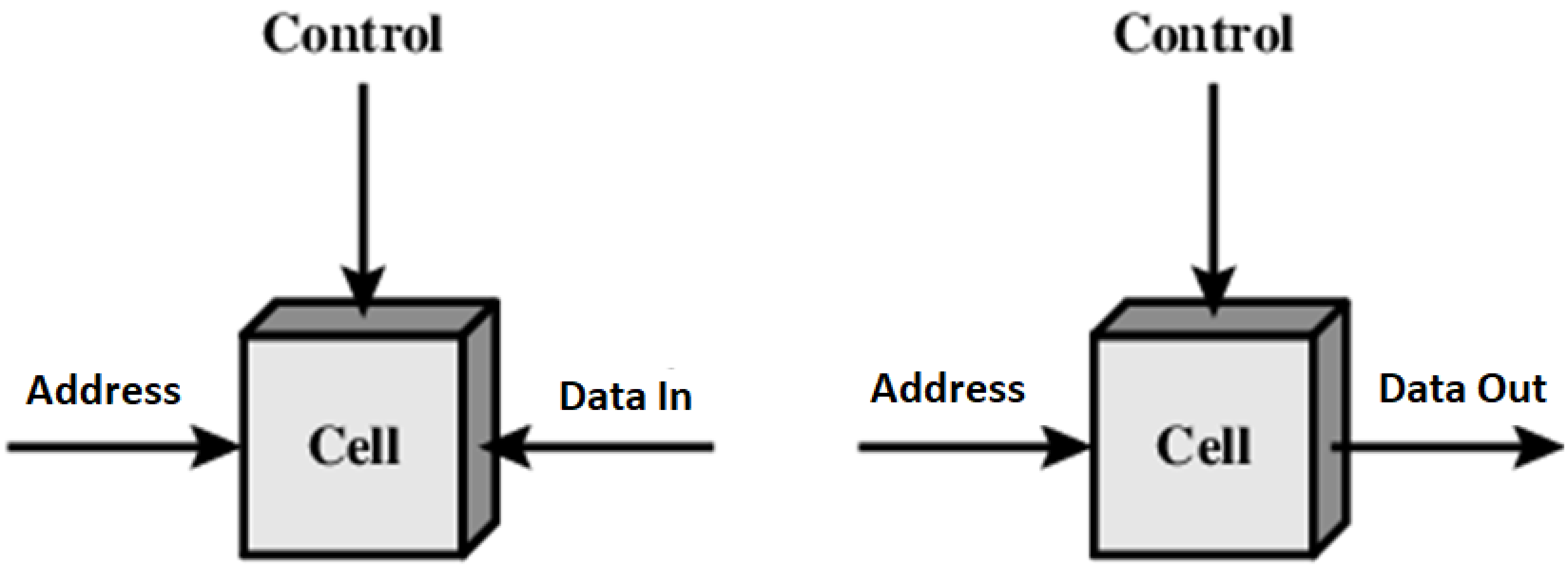
DISK DRIVE- 22% improvement – speeds up 150%---so a 1% increment is INR 7000/22=INR=318

Each 1% of improvement for the processor costs INR333, and for the disk a 1% improvement costs INR318. "Is cost/performance the most important metric?"

BITS Pilani
Pilani Campus

Memory Organization

Semiconductor Memory



(a) Write

(b) Read

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 and DRAM are volatile memories
  - Lose information if powered off.

SRAM vs DRAM Summary

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

Read Only Memory

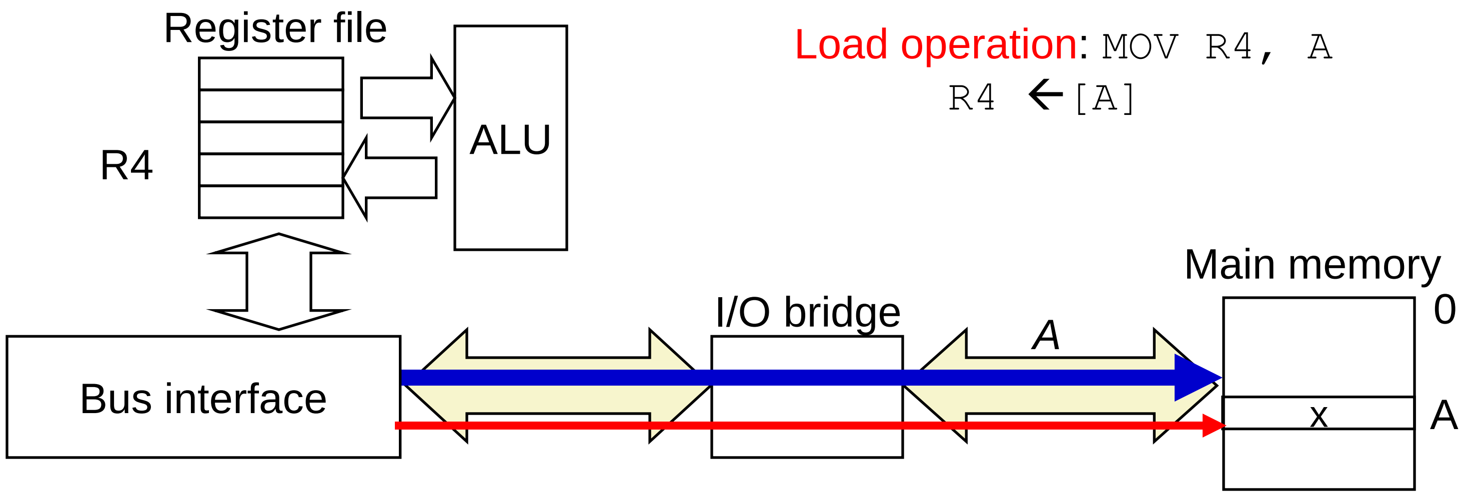
- Permanent Storage and Nonvolatile Memories
- Read Only Memory Variants:
  - Read-only memory (ROM): programmed during production
  - Programmable ROM (PROM): can be programmed once
  - Erasable PROM (EPROM): can be bulk erased (UV, X-Ray)
  - Electrically erasable PROM (EEPROM): electronic erase capability
  - Flash memory: EEPROMs. with partial (block-level) erase capability
    - Wears out after about 100,000 erasing
- Firmware

Applications

- Storing fonts for printers
- Storing sound data in musical instruments
- Video game consoles
- Implantable Medical devices.
- High definition Multimedia Interfaces(HDMI)
- BIOS chip in computer
- Program storage chip in modem, video card and many electronic gadgets, controllers for disks, network cards, ….

Memory Read Operation (1)

CPU places address *A* and then read control signal on the memory bus

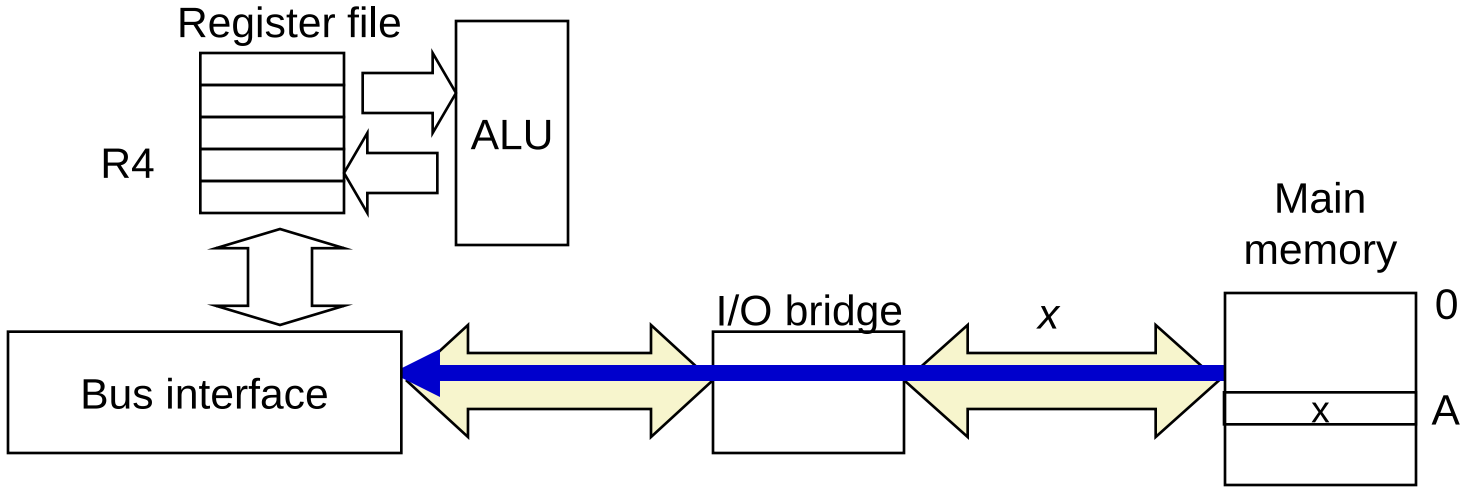


Memory Read Operation (2)

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

Load operation: `MOV R4, A`

`R4 ←[A]`

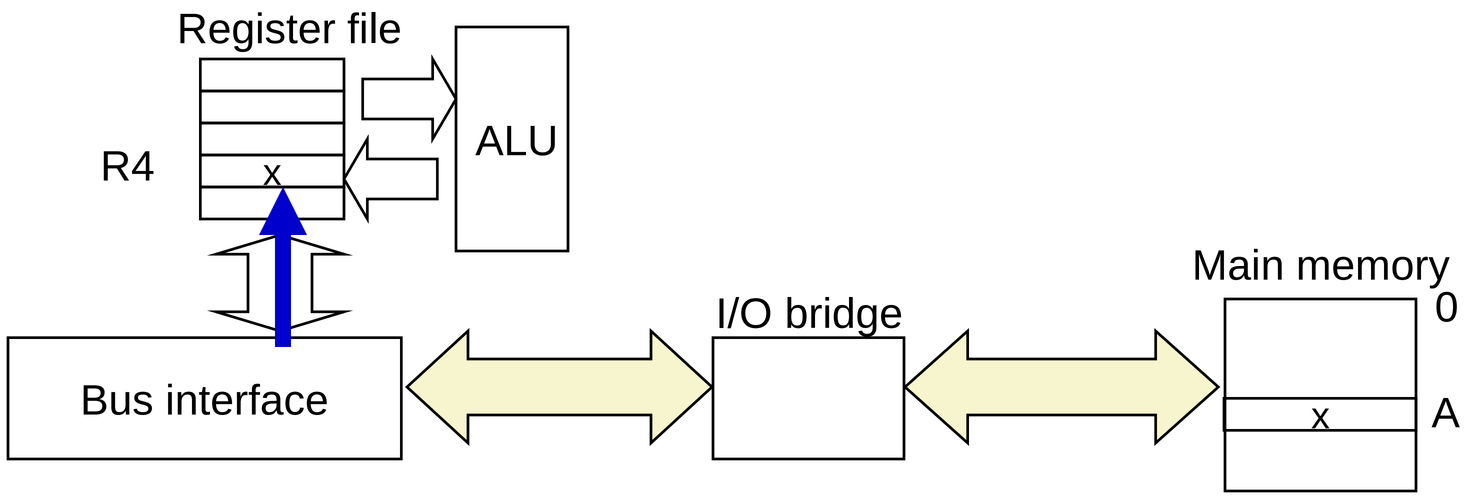


Memory Read Operation (3)

CPU read word x from the bus and copies it into register R4.

Load operation: `MOV R4, A`

`R4 ←[A]`

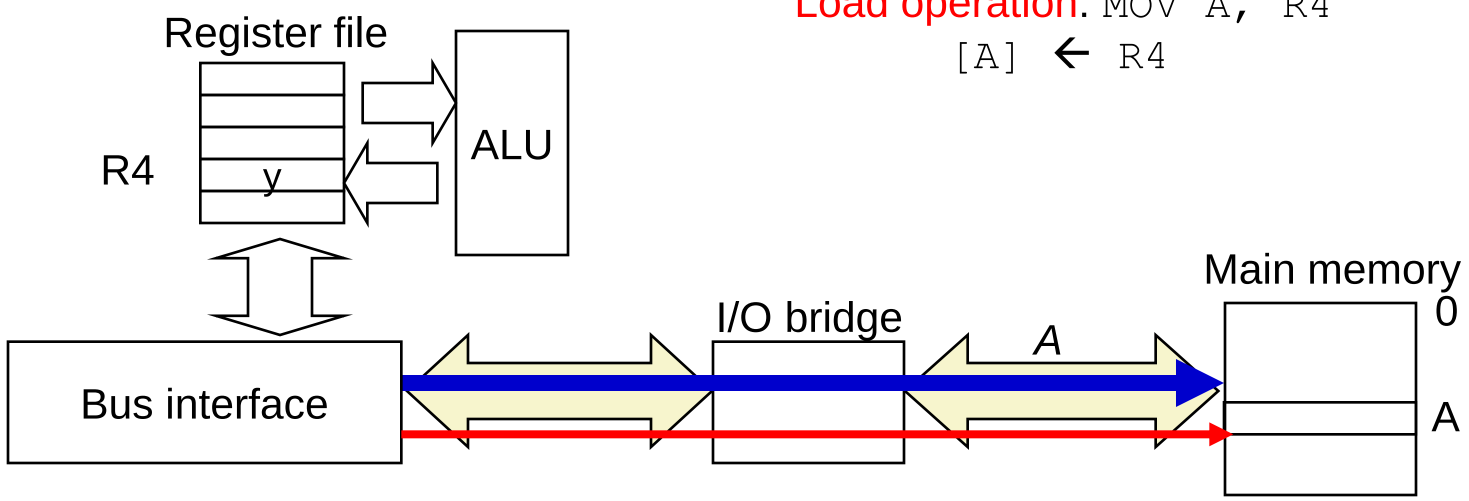


Memory Write Operation (1)

CPU places address A and WRITE control signal on bus. Main memory reads them and waits for the corresponding data word to arrive.

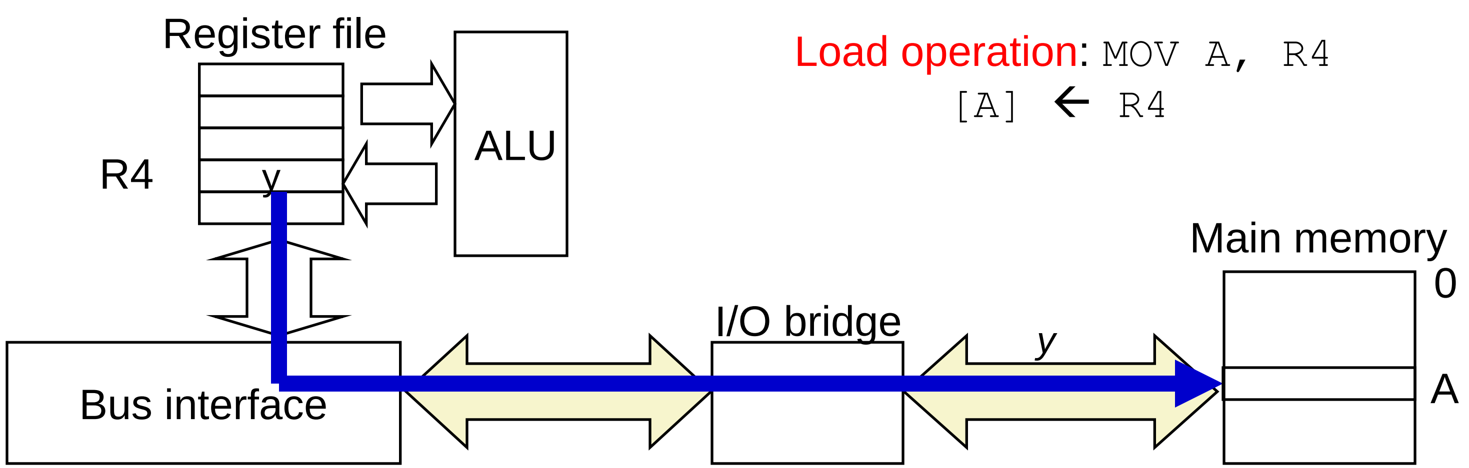
Load operation: `MOV A, R4`

`[A] ← R4`



Memory Write Operation (2)

CPU places data word y on the bus

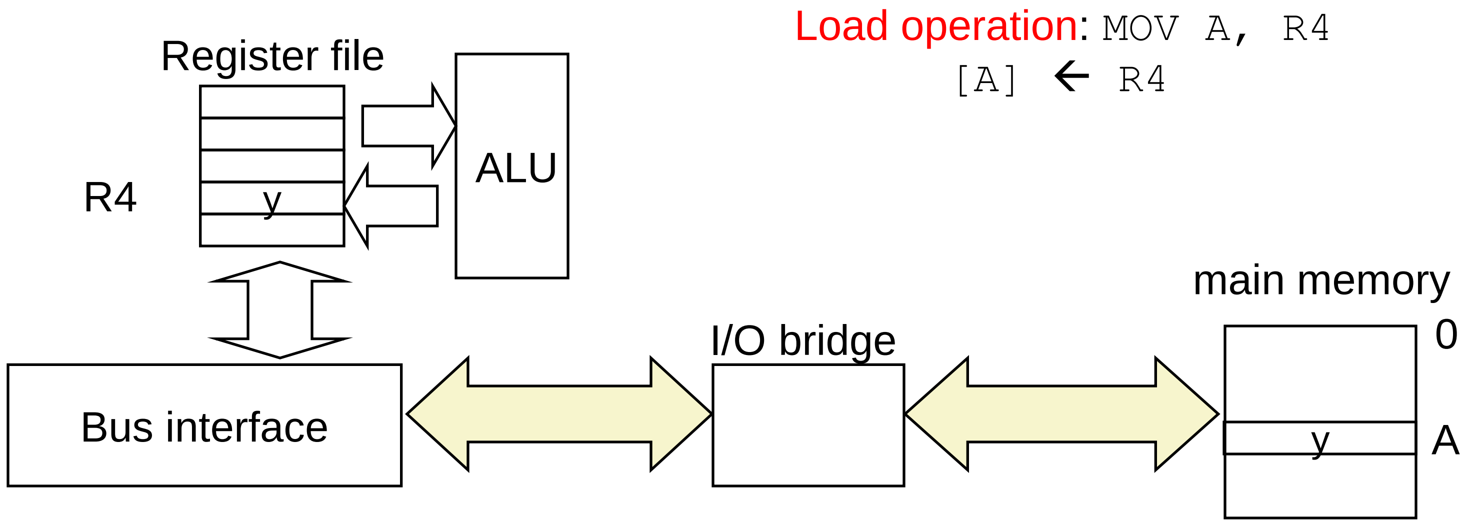


Load operation: `MOV A, R4`

`[A] ← R4`

Memory Write Operation (3)

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



Magnetic Disk Drive

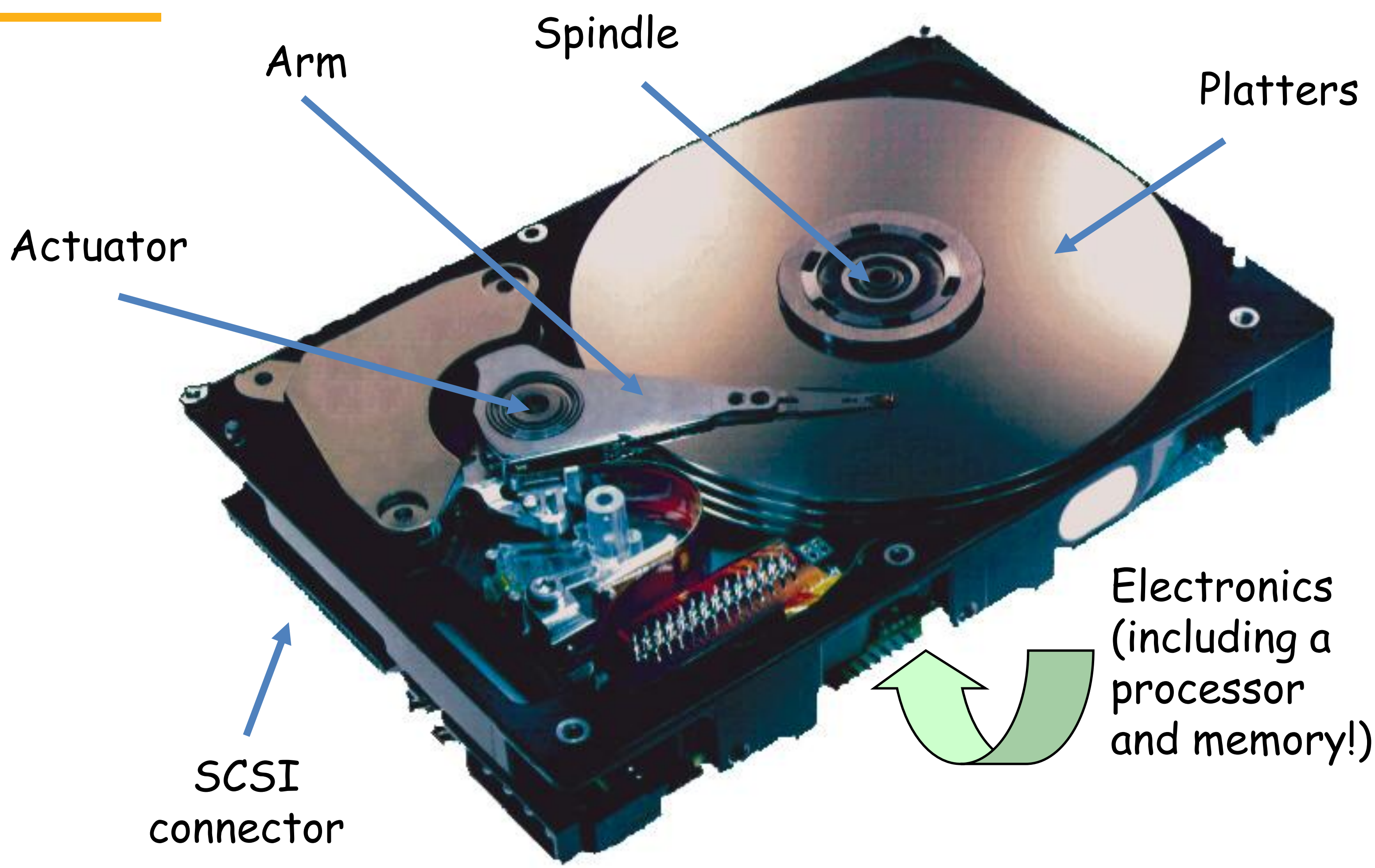
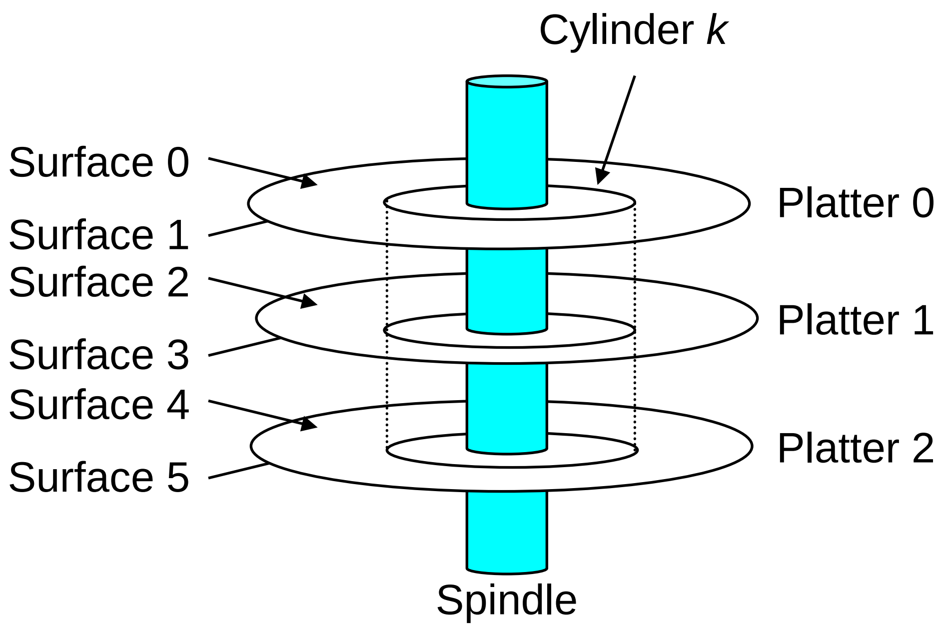


Image courtesy of Seagate Technology

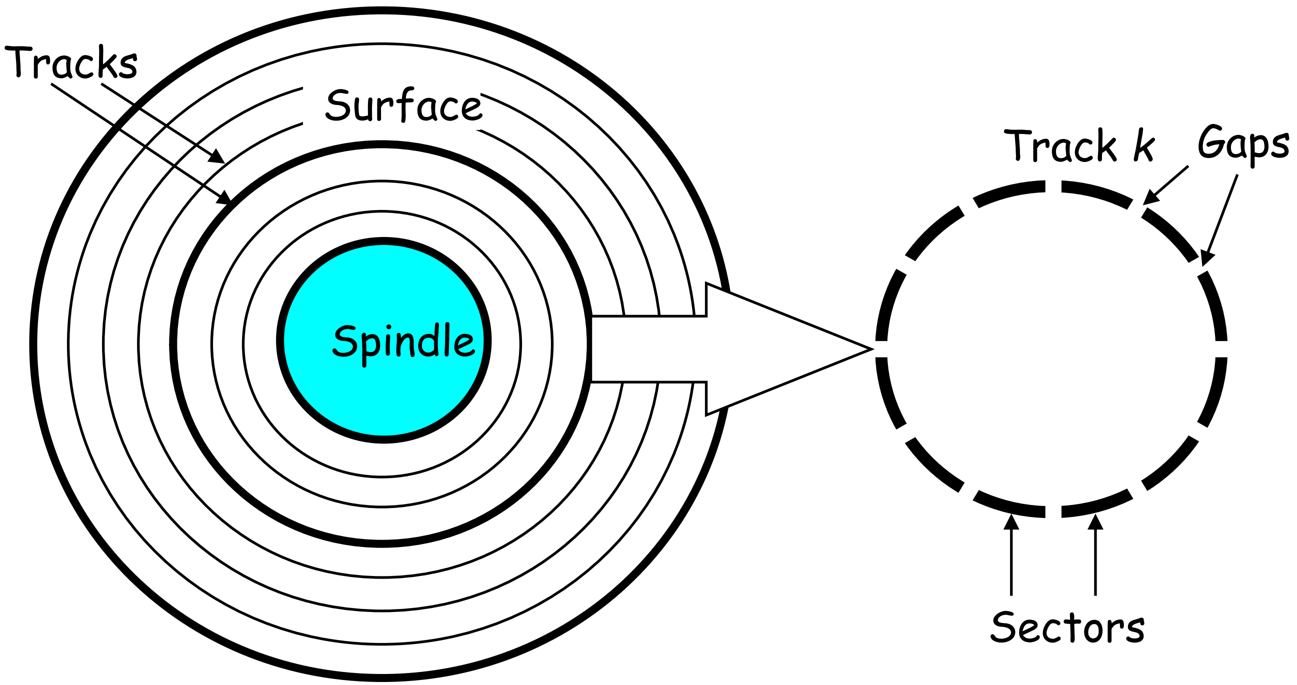
Disk Geometry

- Disks consist of platters, each with two surfaces.
- Each surface consists of concentric rings called tracks
- Aligned tracks form a cylinder



Disk Geometry

- Each track consists of sectors separated by gaps.

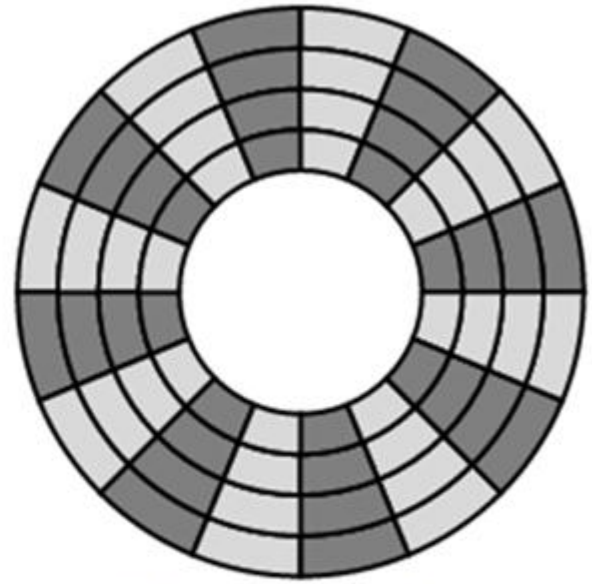


Disk Capacity

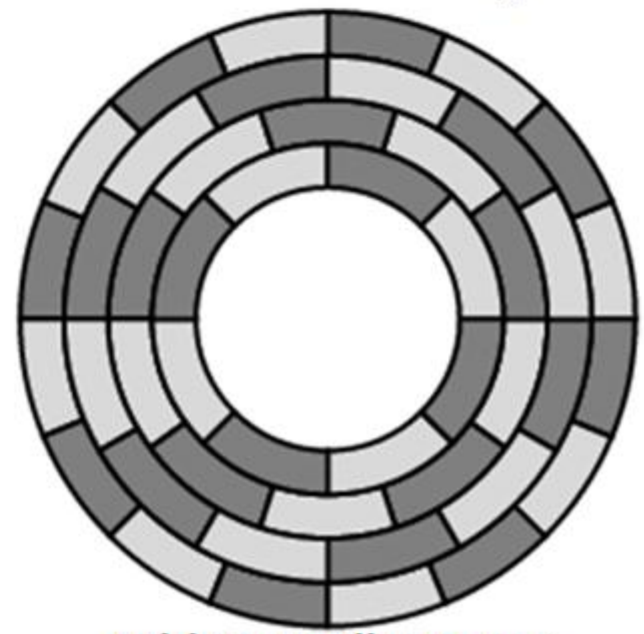
- Capacity: maximum number of bits that can be stored.
  - Vendors express capacity in units of gigabytes (GB /TB), where 1 GB = 2^{30} Bytes, 1 TB = 2^{40} Bytes,
- Capacity is determined by these technology factors:
  - Recording density (bits/in): number of bits that can be squeezed into a 1 inch 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.



Without Recording Zones



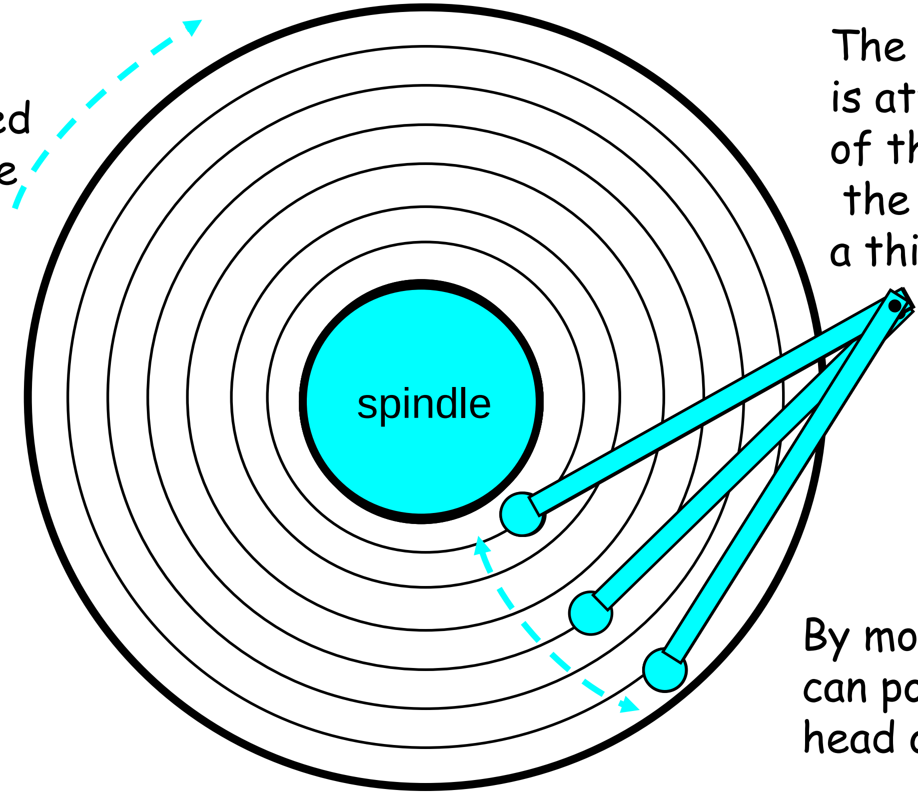
With Recording Zones

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
  = 28.61 GB

Disk Operation (Single-Platter View)

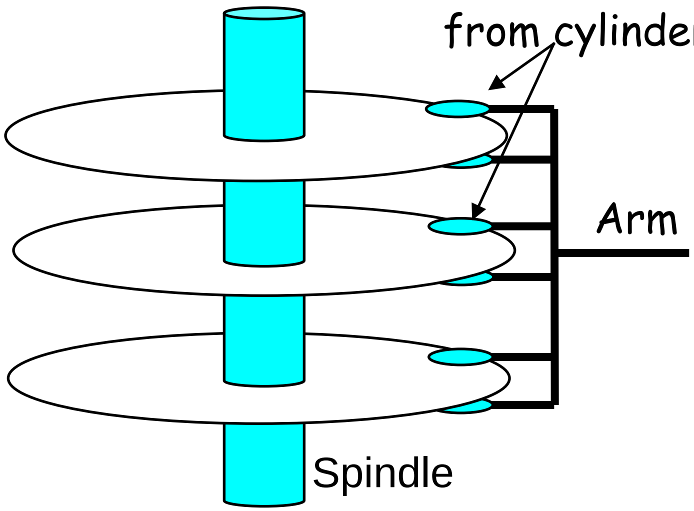
The disk surface spins at a fixed rotational rate



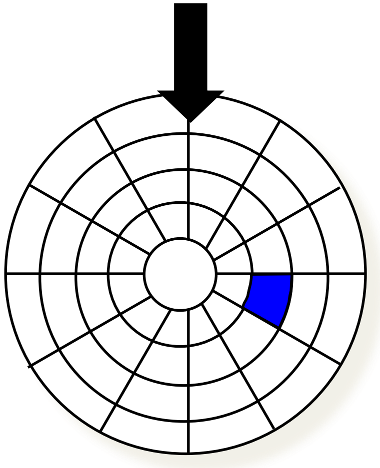
The read/write *head* is attached to the end of the *arm* and flies over the disk surface on a thin cushion of air.

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

Disk Operation (Multi-Platter View)

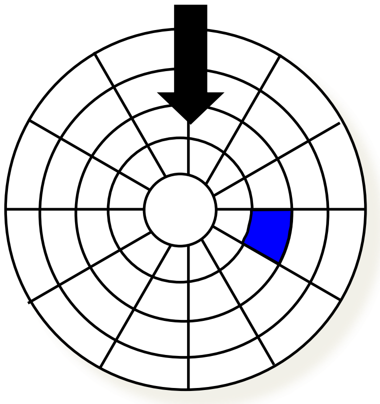


Disk Access



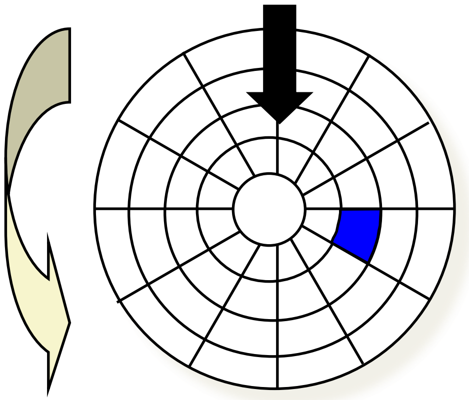
Need to access a sector colored in blue

Disk Access



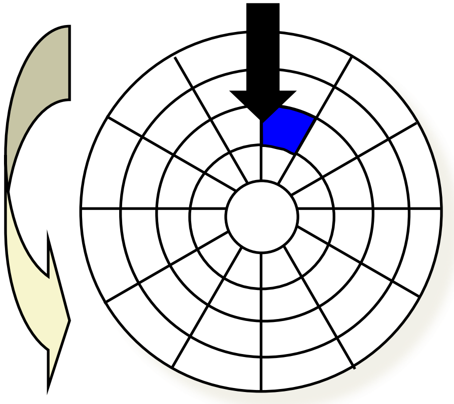
Head in position above a track

Disk Access



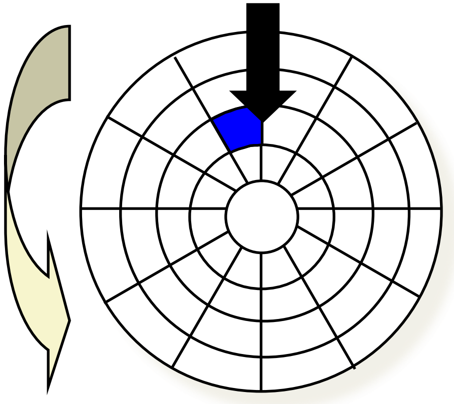
Rotate the platter in counter-clockwise direction

Disk Access – Read



About to read blue sector

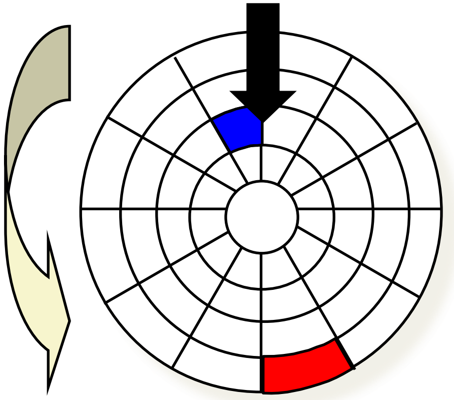
Disk Access – Read



After BLUE read

After reading blue sector

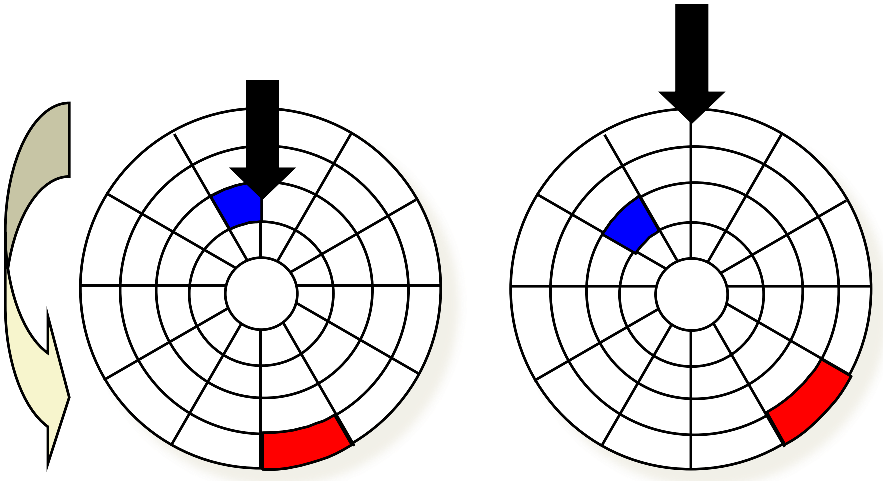
Disk Access – Read



After BLUE read

Red request scheduled next

Disk Access – Seek

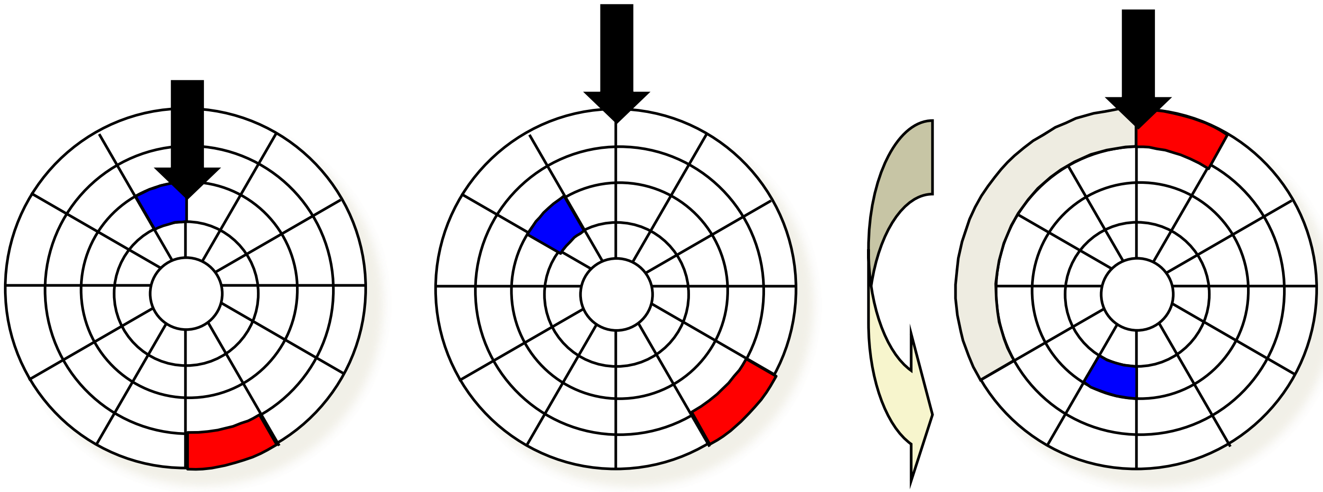


After BLUE read

Seek for RED

Seek to red's track

Disk Access – Rotational Latency



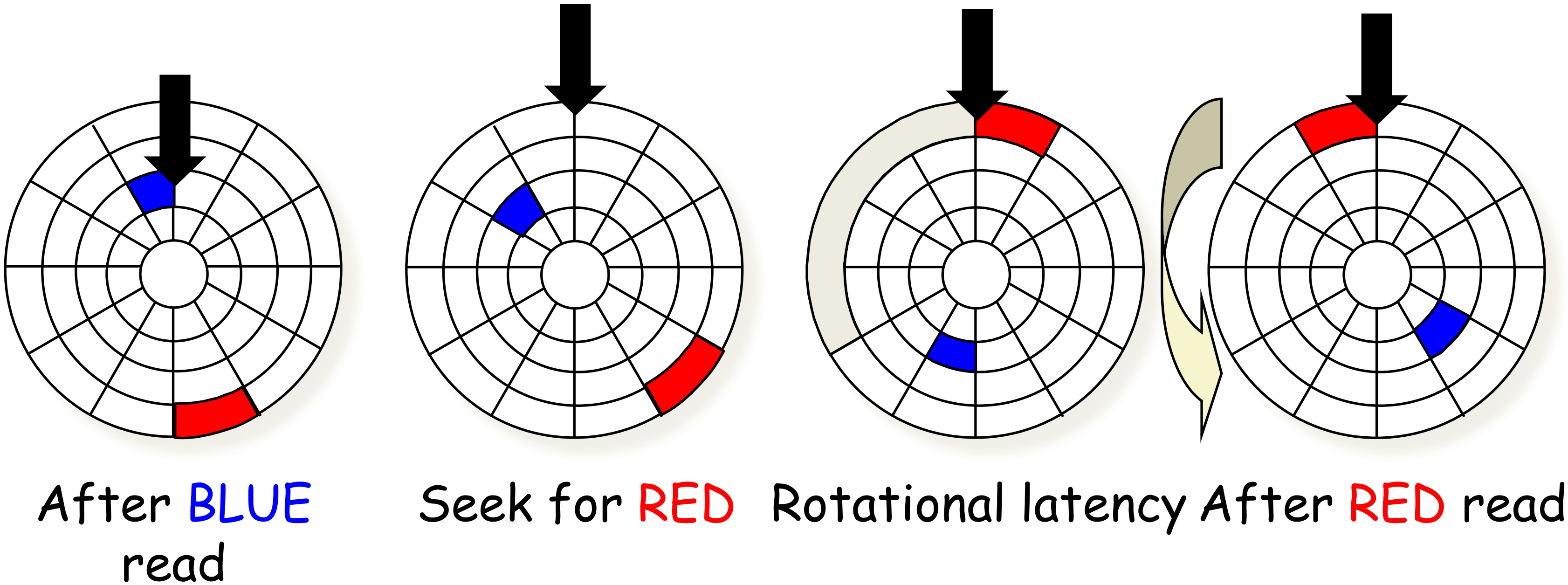
After BLUE read

Seek for RED

Rotational latency

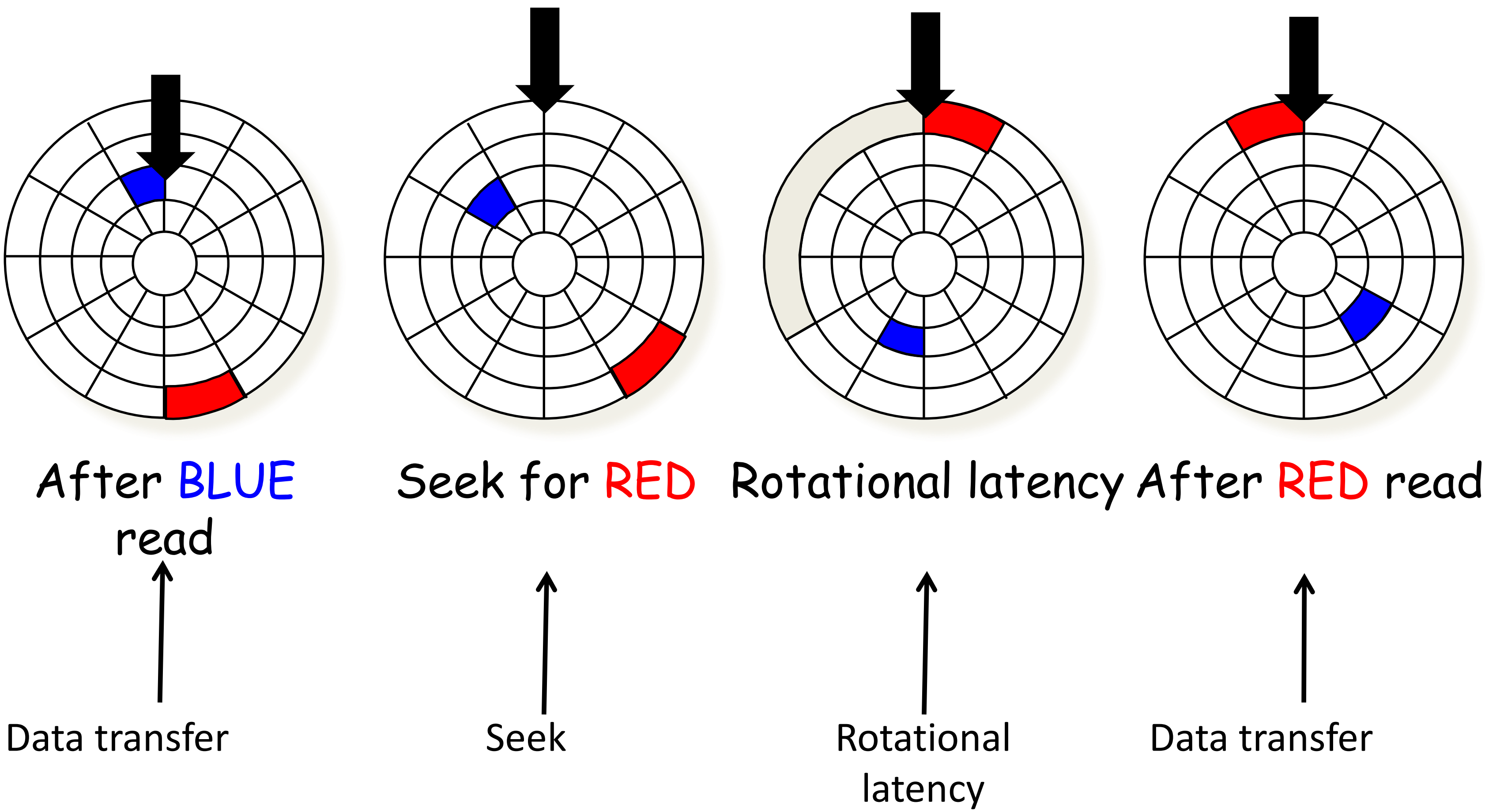
Wait for red sector to rotate around

Disk Access – Read



Complete read of red

Disk Access – Access Time Components



Disk Access Time

- Average time to access some target sector given by :
  - Taccess = Tavg seek + Tavg rotation + Tavg 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.
  - Tavg rotation = 1/2r , where r is rotation Speed in **revolution per Second**
  - Typical Tavg rotation = 7200 RPMs = 7200/60 RPS
- Transfer time (Tavg transfer)
  - Time to read the bits in the target sector.
  - Tavg transfer = b/rN, where b is the number of bytes to be transferred and N is the average number of bytes on a track

Disk Access Time Example

Given:
- Rotational rate = 7,200 RPM
- Average seek time = 9 ms.
- Avg # sectors/track = 400.
- 512 bytes per sector

Derived:
- Tavg rotation = 1/2r = 1/2 x (60 secs/7200 RPM)
  = 0.00416 = 4.16ms.
- Tavg transfer =b/rN
  = 512 x 60/7200 x 1/(400*512)
  = 0.02 ms
- Taccess = 9 ms + 4.16ms + 0.02 ms=13.18ms

Contd..

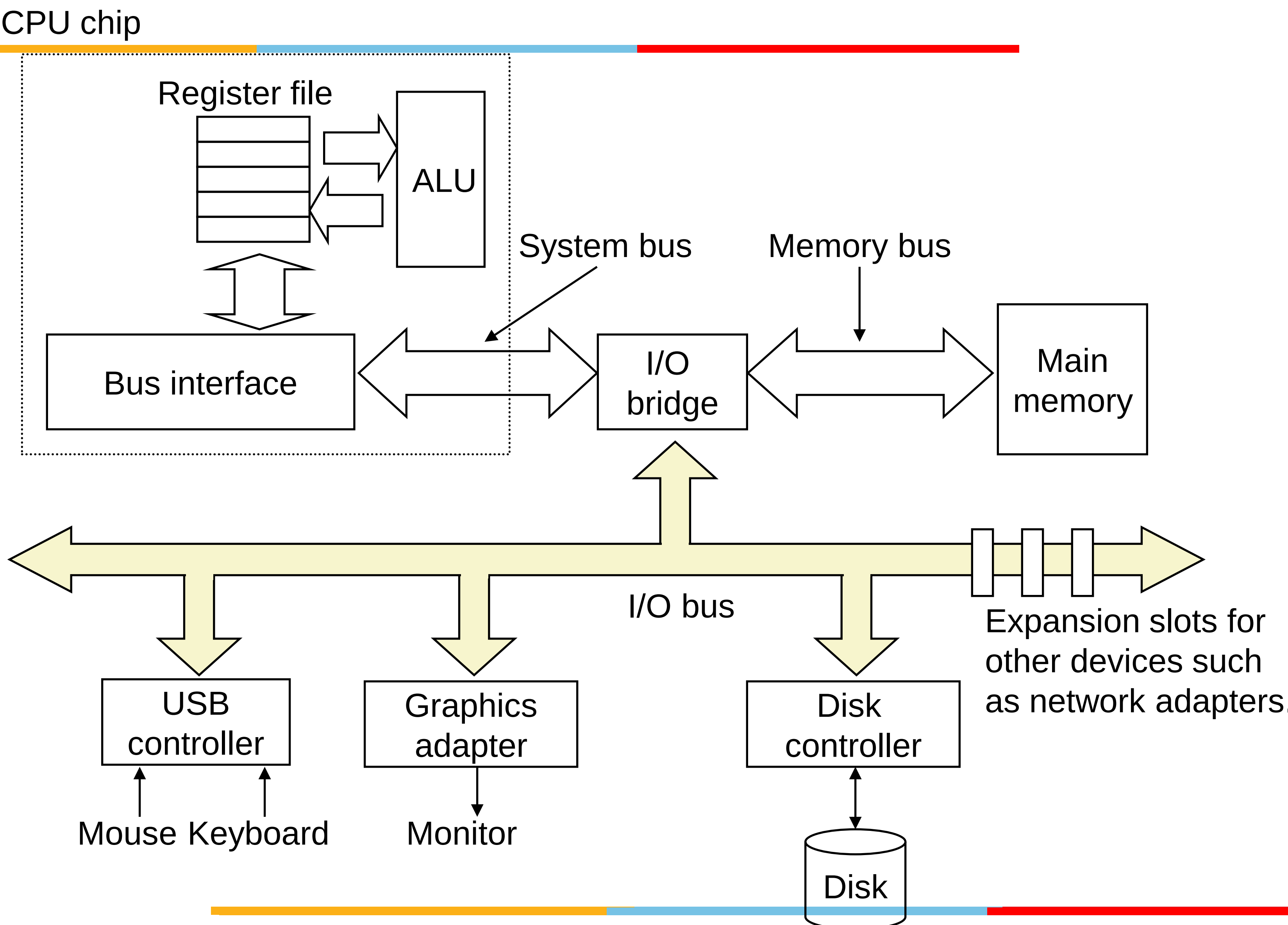
Important points:

- 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 then DRAM.

Logical Disk Blocks

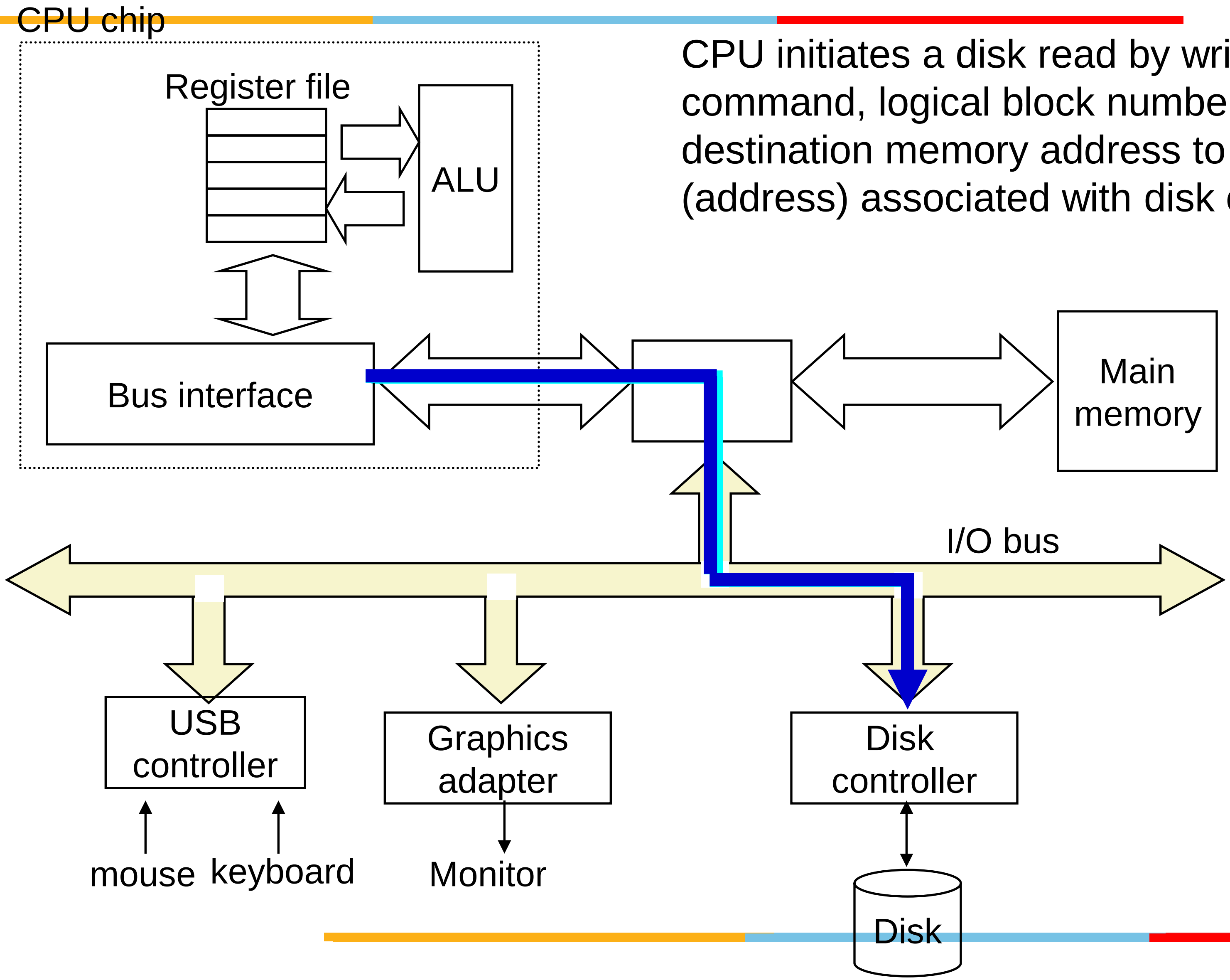
- Modern disks present a simpler abstract view of the complex sector geometry:
  – The set of available sectors is modeled as a sequence of b-sized logical blocks (0, 1, 2, ...)
- Mapping between logical blocks and actual (physical) sectors
  – Maintained by hardware/firmware device called disk controller.
  – 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



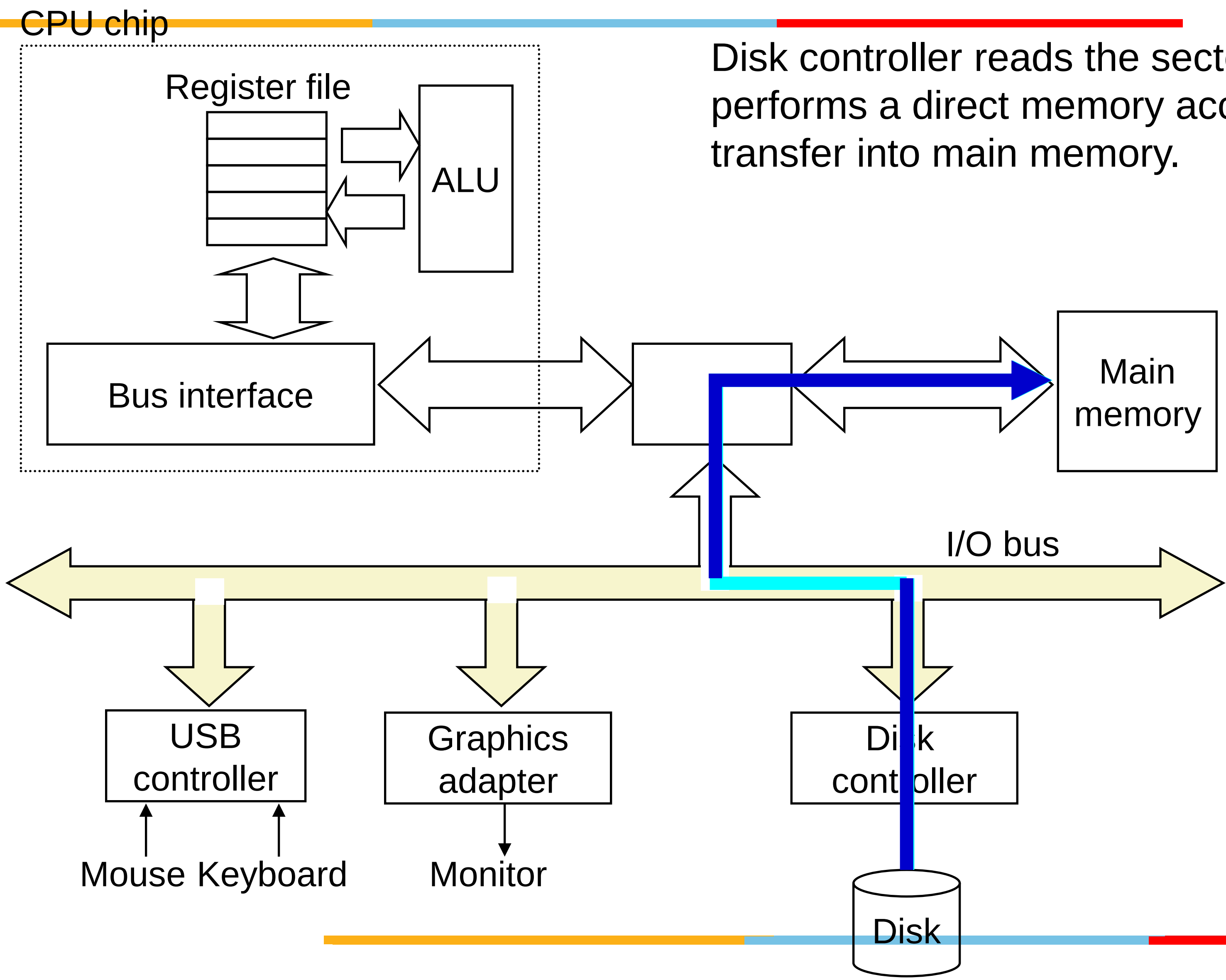
CPU chip
Register file
ALU
System bus
Memory bus
Bus interface
I/O bridge
Main memory
I/O bus
Expansion slots for other devices such as network adapters.
USB controller
Graphics adapter
Disk controller
Mouse
Keyboard
Monitor
Disk

Reading a Disk Sector (1)



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

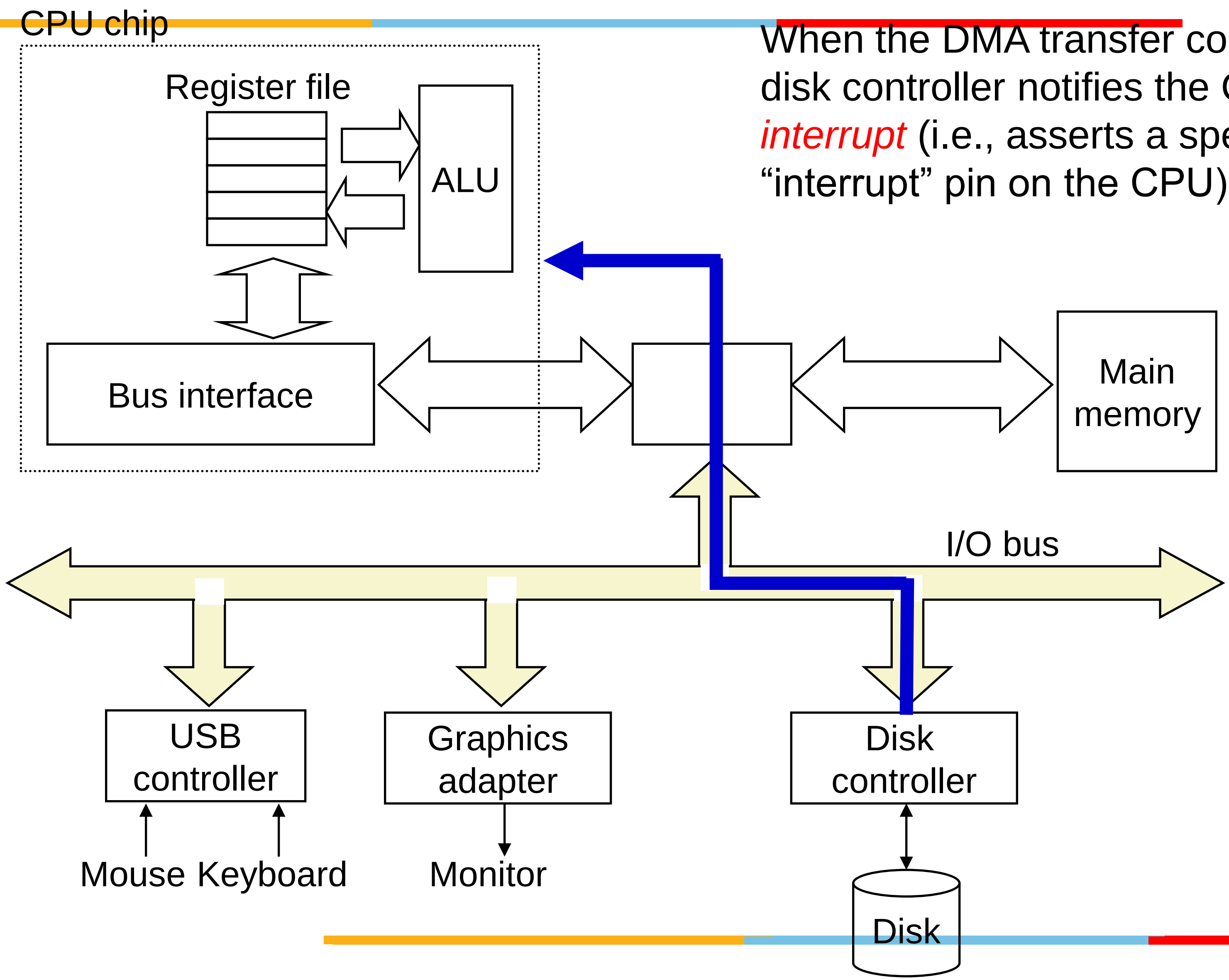
Reading a Disk Sector (2)



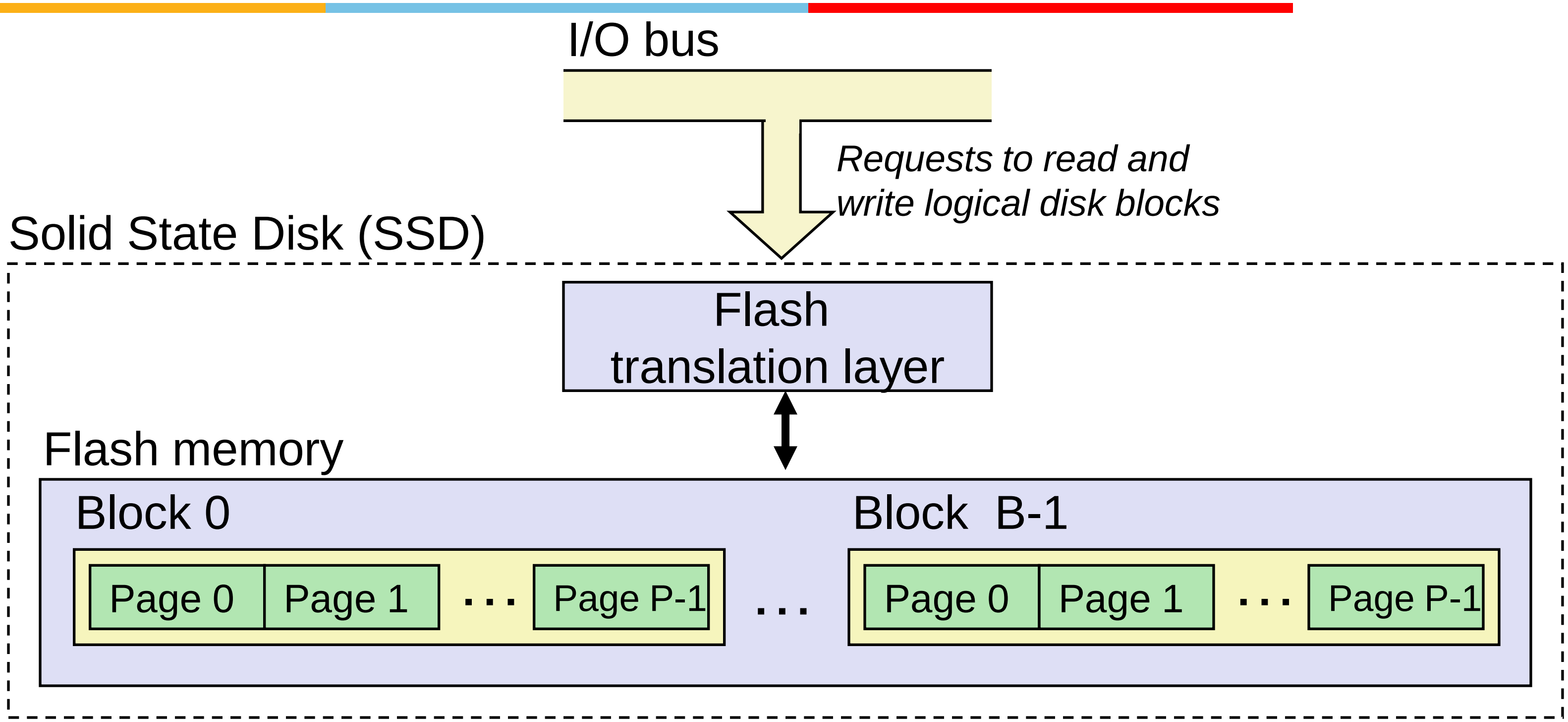
Disk controller reads the sector and performs a direct memory access (DMA) transfer into main memory.

Reading a Disk Sector (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)



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 input	550 MB/s	Sequential write output	470 MB/s
Random read input	365 MB/s	Random write output	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
  - 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 x 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