R

بسم الله الرحمن الرحیم

سیستم عامل

جلسه بیست و یکم – حافظه‌ی جانبی (۱)

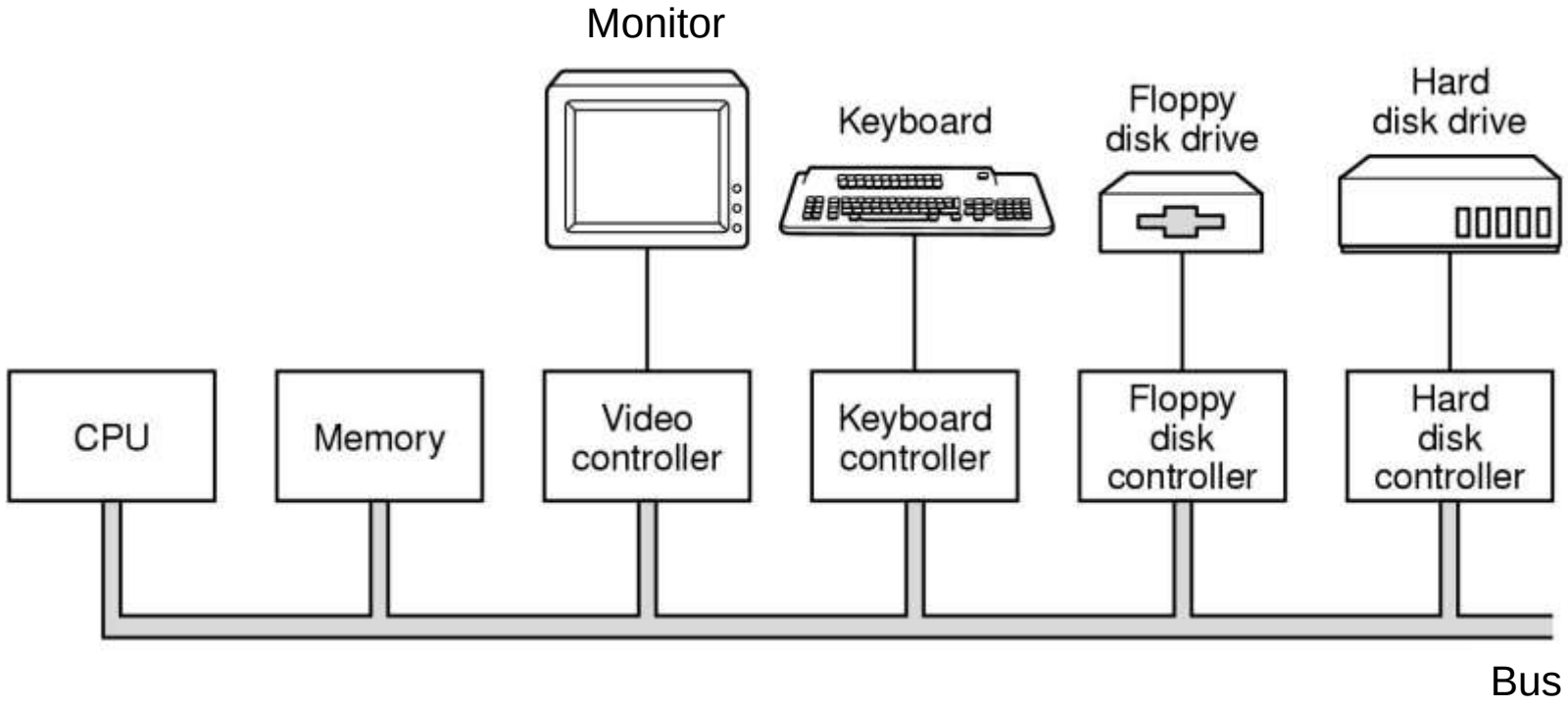
جلسه‌ی گذشته

Device Terminology

- Device (mechanical hardware)
- Device controller (electrical hardware)
- Device driver (software)

Devices & Controllers

- Components of a simple personal computer



پرسش؟ این دیوایس کنترلر دقیقا کجاست؟

- مثلا الآن یه مانیتور وصل کنیم، دیوایس کنترلر توی مانیتور هست؟ یا قبلا توی مادربردمون قرار گرفته؟ موس و پرینتر و ... چطور؟
- جواب: هر کدام یه جوری هستند!

Location Of Device Controller

Device	Location of Device Controller
Mouse/Keyboard	**On the motherboard** (USB or PS/2 controller)
Speakers	**Sound card** (or **integrated motherboard audio controller**)
Hard Disk (HDD/SSD)	**Inside the drive** (HDD controller or SSD controller)
Printer	**Inside the printer** (Printer controller)
Network Card (NIC)	**On the network card** (Ethernet or Wi-Fi adapter controller)
USB Devices	**On the motherboard** (USB host controller) and **inside each USB device**

Device Controllers

- The Device vs. its Controller
- Some duties of a device controller:
  - *Interface between CPU and the Device*
  - *Start/Stop device activity*
  - *Convert serial bit stream to a block of bytes*
  - *Deal with error detection/correction*
  - *Move data to/from main memory*
- Some controllers may handle several (similar) devices

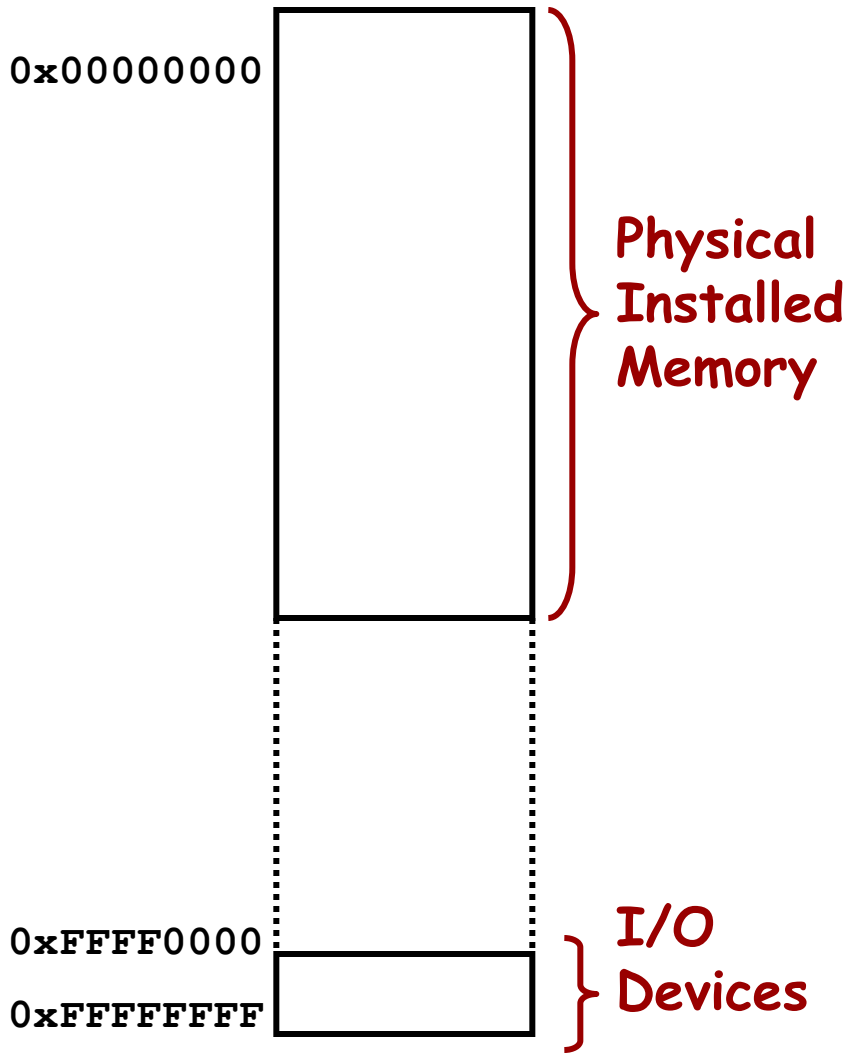
I/O Ports

- Each port has a separate number.
- CPU has special I/O instructions
  - `in  r4,3`
  - `out  3,r4`

- Port numbers form an “address space”... separate from main memory
- Contrast with
  - `load  r4,3`
  - `store  3,r4`

Memory-Mapped I/O

- One address space for
  - *main memory*
  - *I/O devices*
- CPU has no special instructions
  - `load r4,addr`
  - `store addr,r4`
- I/O devices are “mapped” into
  - *very high addresses*



I/O Device Speed

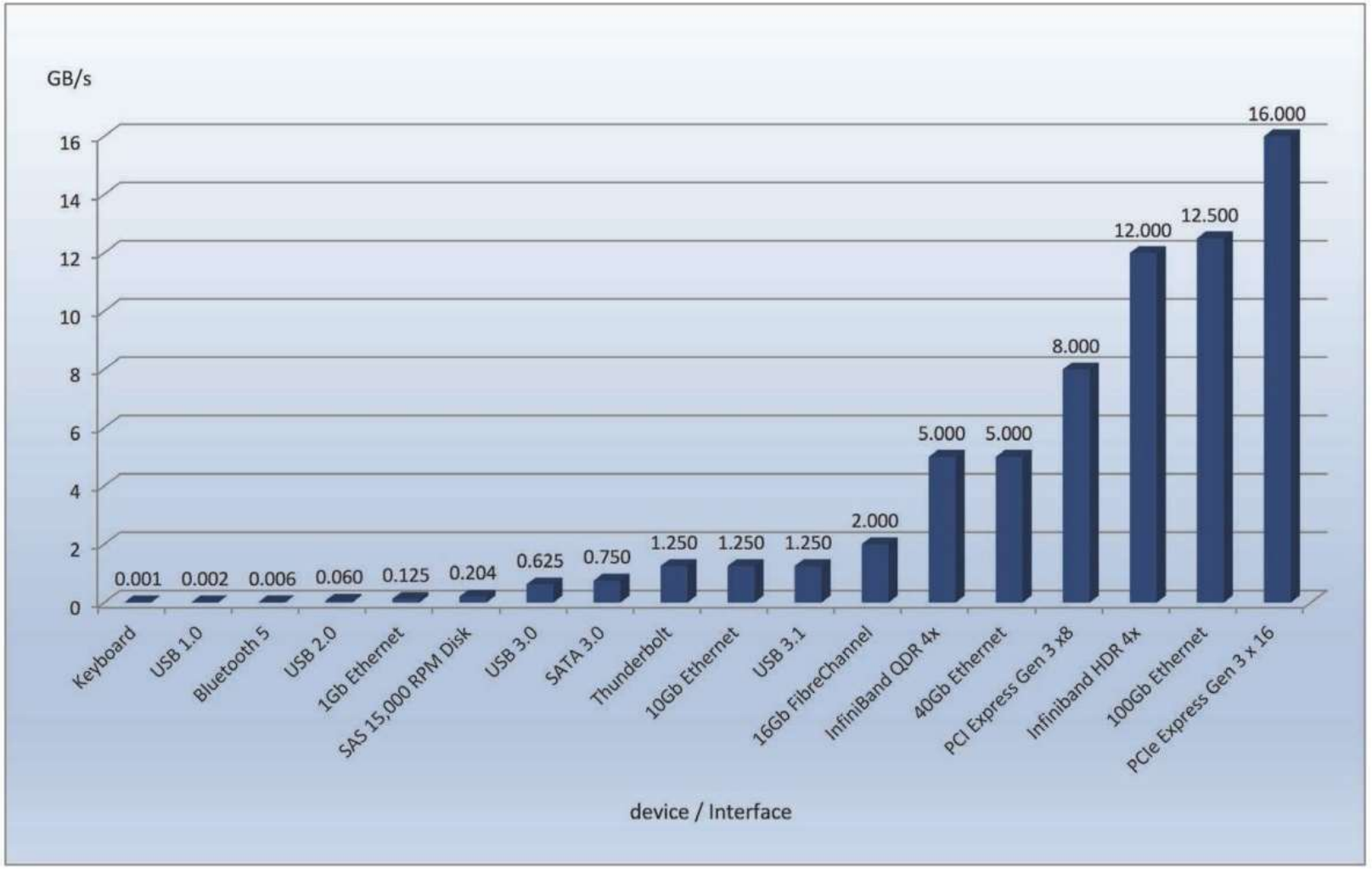
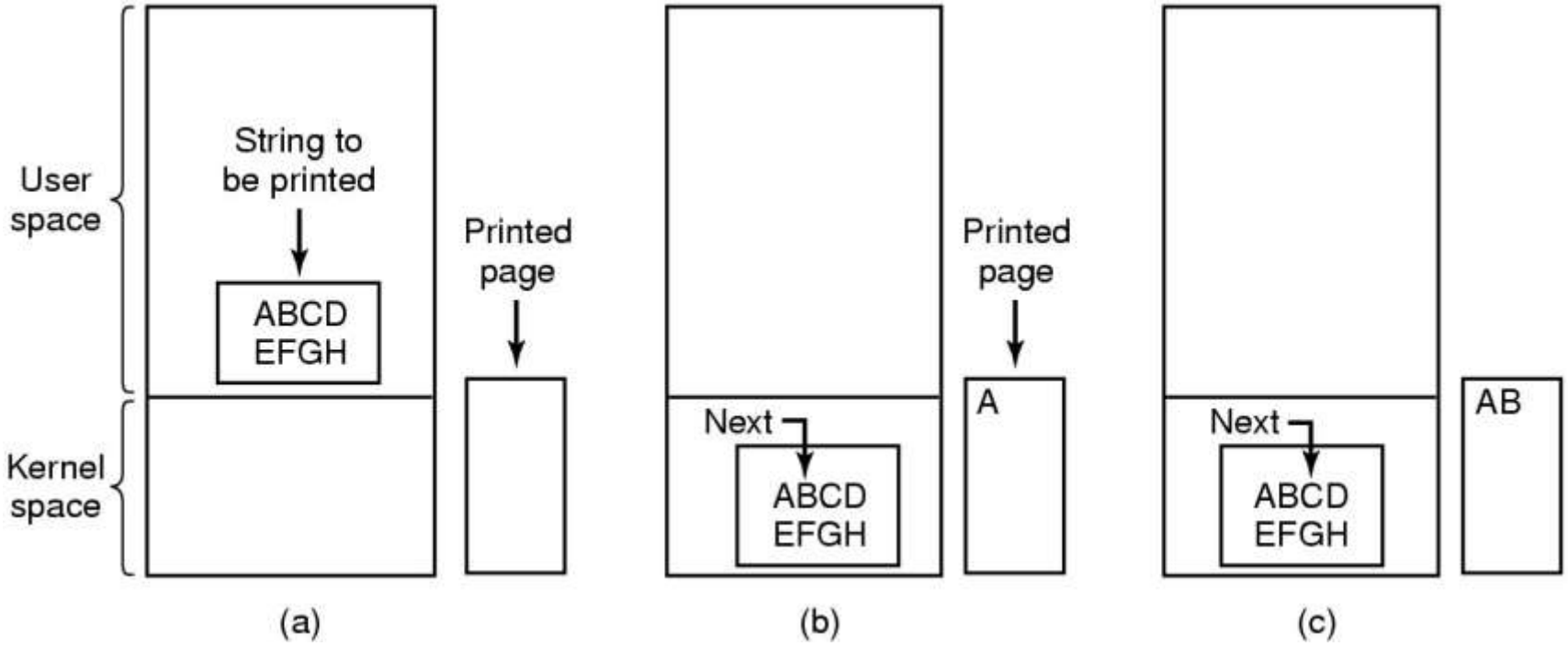


Figure 12.11 Common PC and data-center I/O device and interface speeds.

Programmed I/O



Steps in printing a string

Interrupt-Driven I/O

- Getting the I/O started:

```
CopyFromUser(virtAddr, kernelBuffer,
byteCount)
EnableInterrupts()
while *serialStatusReg != READY
endWhile
*serialDataReg = kernelBuffer[0]
Sleep ()
```

Interrupt-Driven I/O

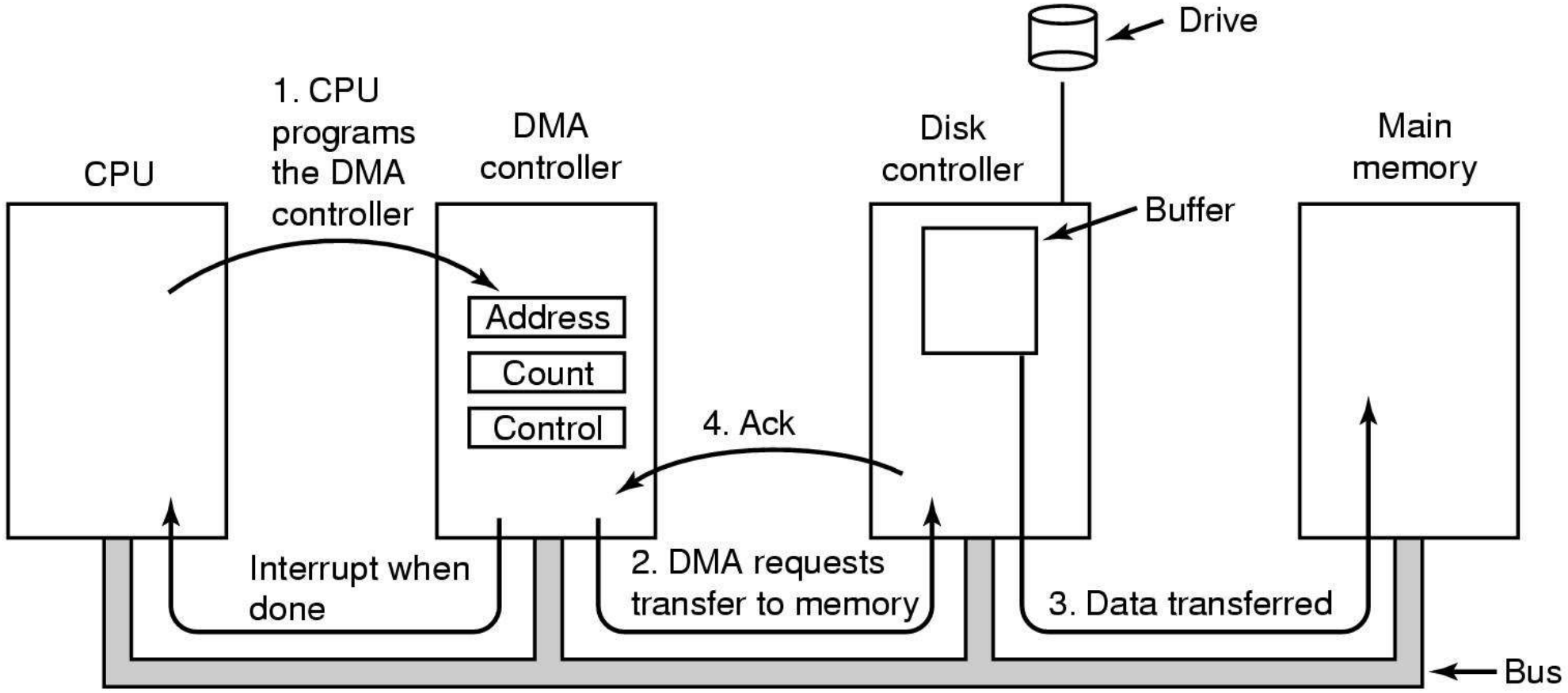
- The Interrupt Handler:

```
if i == byteCount
   Wake up the user process
else
   *serialDataReg = kernelBuffer[i]
   i = i + 1
endIf
Return from interrupt
```

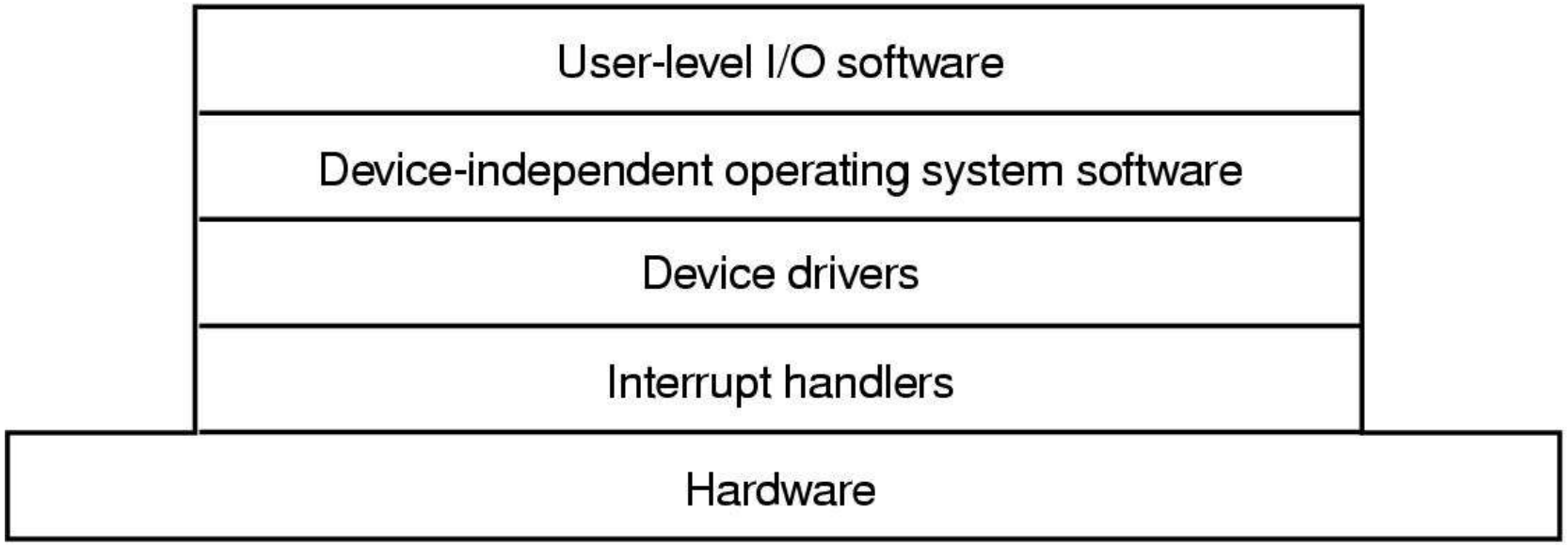
Direct Memory Access (DMA)

- Data transferred from device straight to/from memory
- CPU not involved
- *The DMA controller:*
  - *Does the work of moving the data*
  - *CPU sets up the DMA controller (“programs it ”)*
  - *CPU continues*
  - *The DMA controller moves the bytes*

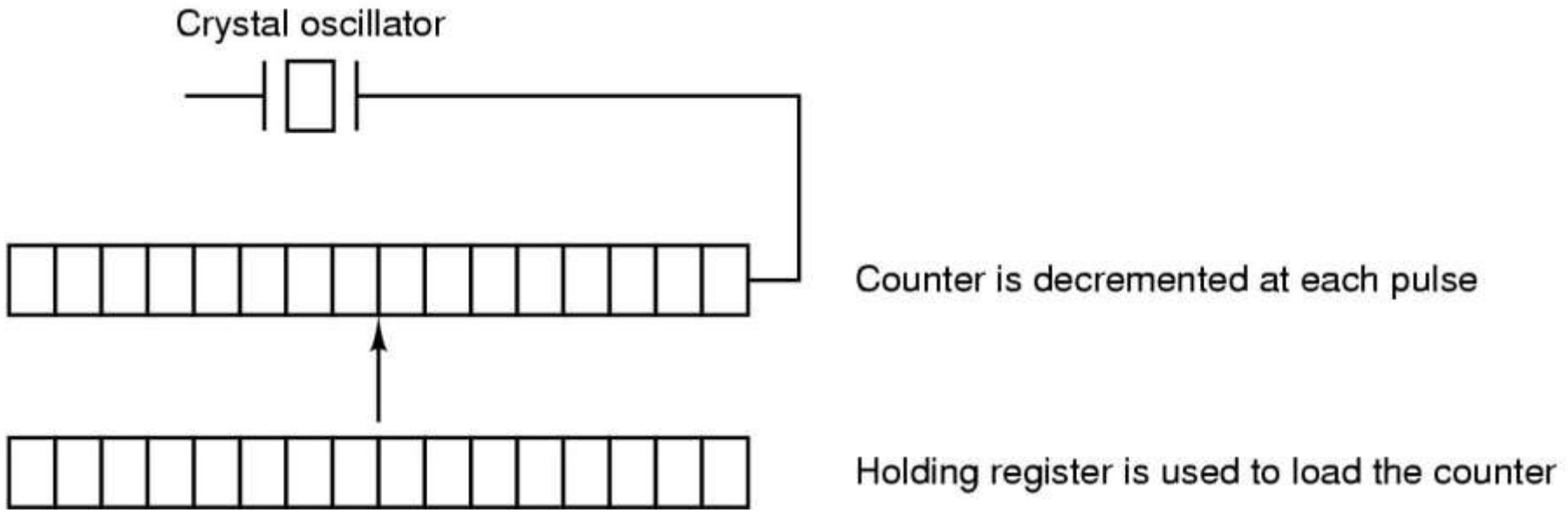
Direct Memory Access (DMA)



I/O Software Layers



Programmable Timer

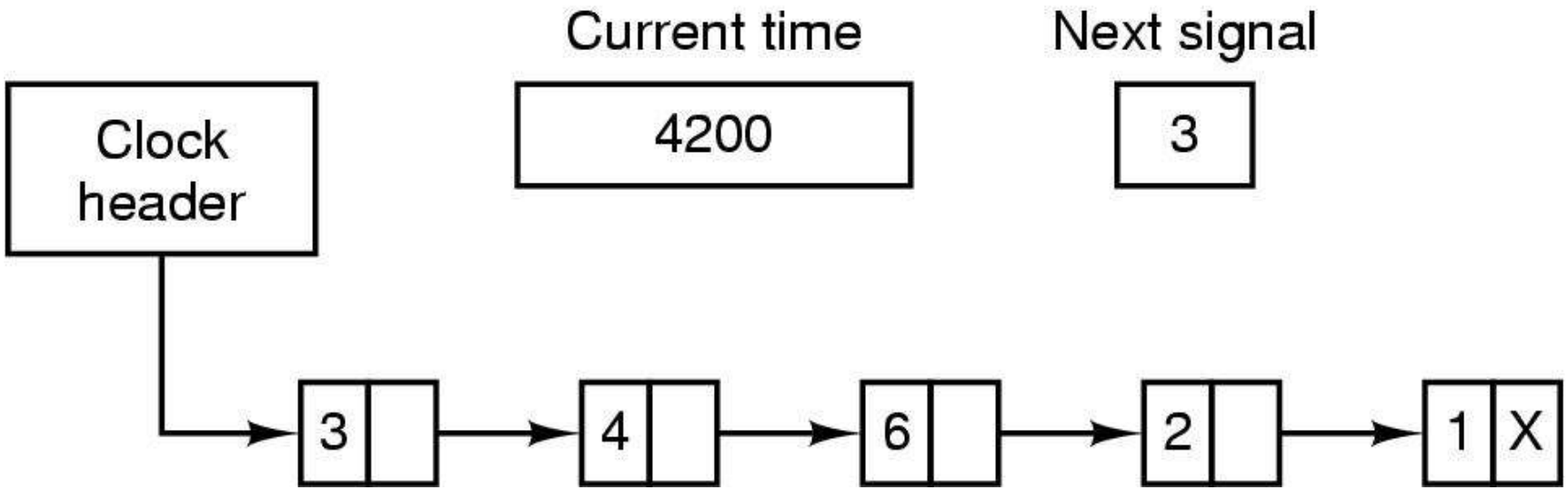


- One-shot mode:
  - *Counter initialized then decremented until zero*
  - *At zero a single interrupt occurs*
- Square wave mode:
  - *At zero the counter is reinitialized with the same value*
  - *Periodic interrupts (called "clock ticks") occur*

Goals of Timer Software

- Maintain time of day
  - - *Must update the time-of-day every tick*
- Prevent processes from running too long
- Account for CPU usage
  - - *Separate timer for every process*
  - - *Charge each tick to the current process*
- Handling the “Alarm” syscall
  - - *User programs ask to be sent a signal at a given time*
- Providing watchdog timers for the OS itself
  - - *When to stop the disk, switch to low power mode, etc*
- Doing profiling, monitoring, and statistics gathering

Software Timers



- Alarms set for 4203, 4207, 4213, 4215 and 4216.
- Each entry tells how many ticks past the previous entry.
- On each tick, decrement the “NextSignal”.
- When it gets to 0, then signal the process.

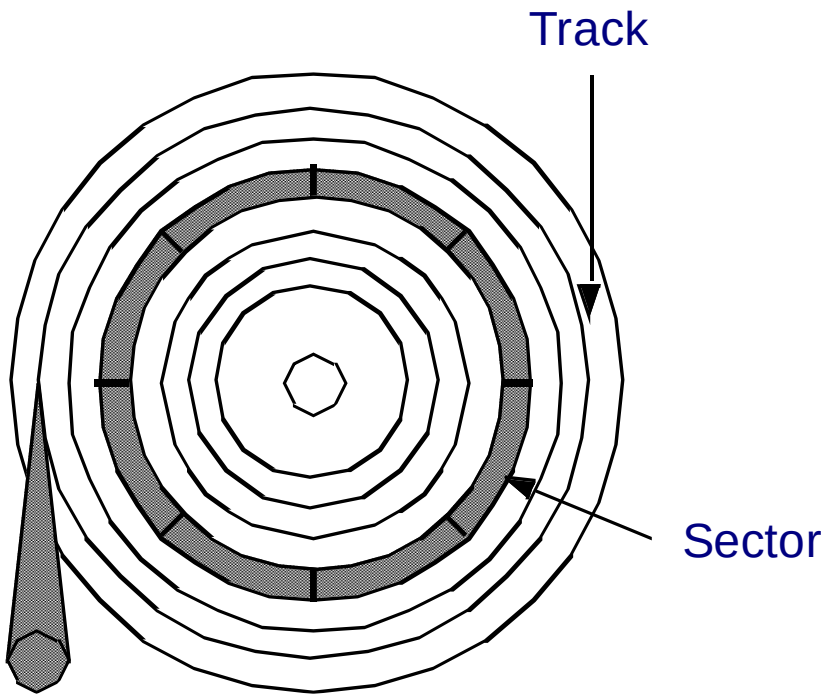
جلسه جدید

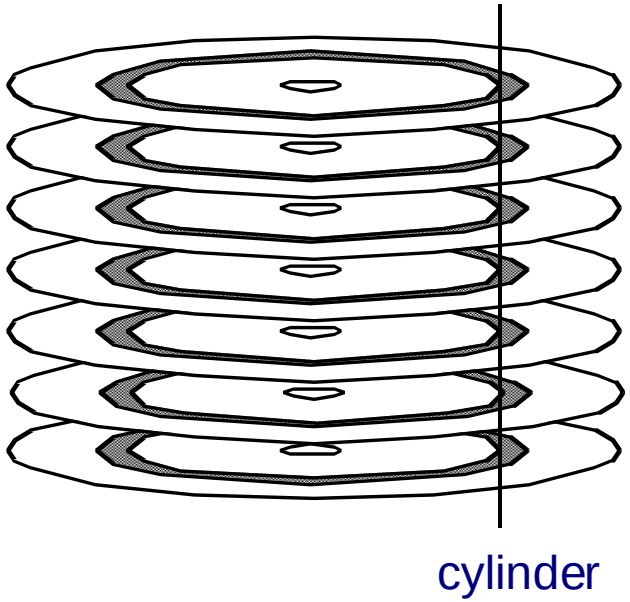
Overview of Mass Storage Structure

- Bulk of secondary storage for modern computers is **hard disk drives** (**HDDs**) and **nonvolatile memory** (**NVM**) devices
- **HDDs** spin platters of magnetically-coated material under moving read-write heads
  - *Drives rotate at 60 to 250 times per second*
  - ***Transfer rate*** *is rate at which data flow between drive and computer*
  - ***Positioning time*** *(**random-access time**) is time to move disk arm to desired cylinder (**seek time**) and time for desired sector to rotate under the disk head (**rotational latency**)*
  - ***Head crash*** *results from disk head making contact with the disk surface -- That's bad*
- Disks can be removable

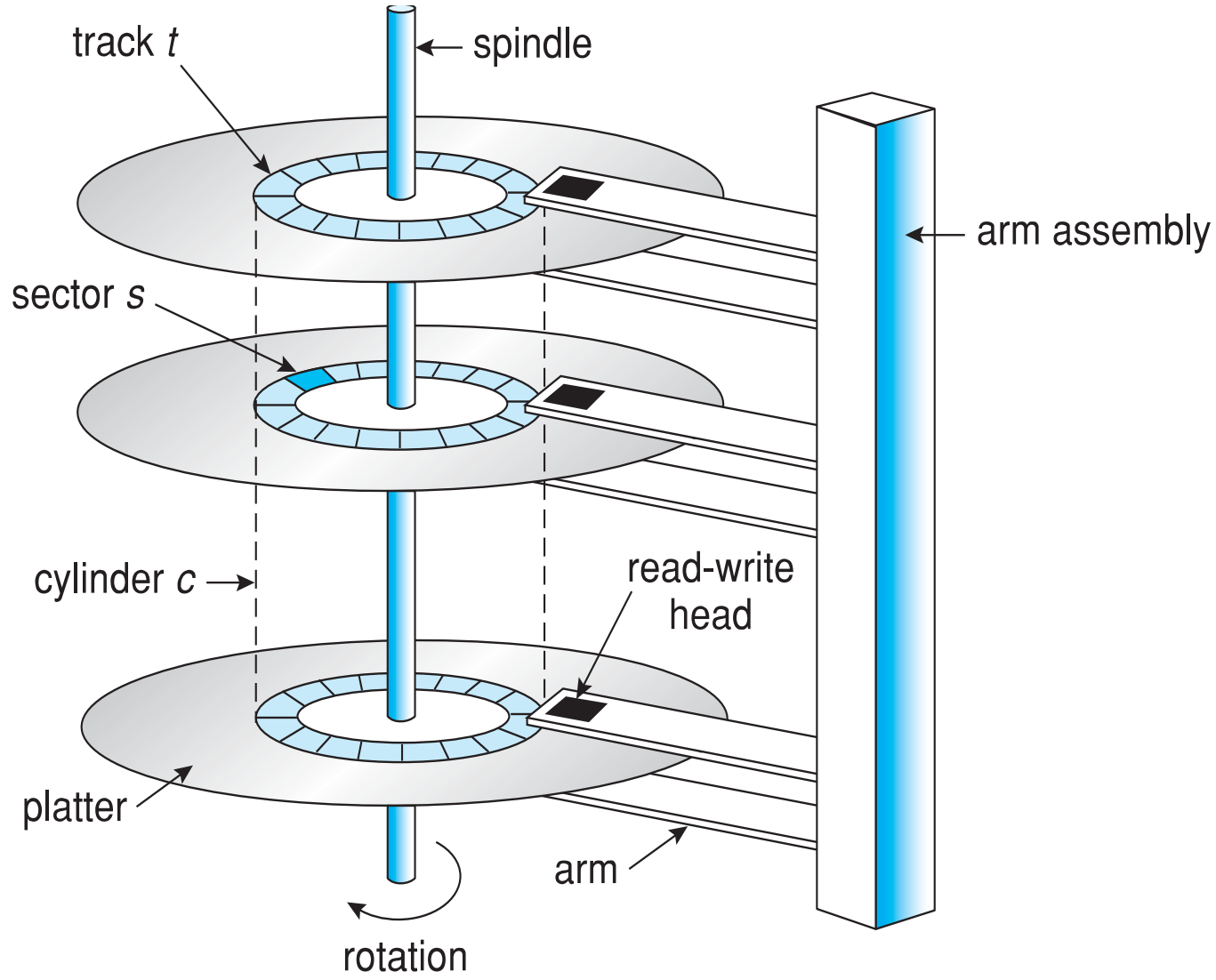
Disk Geometry

- Disk head, surfaces, tracks, sectors …





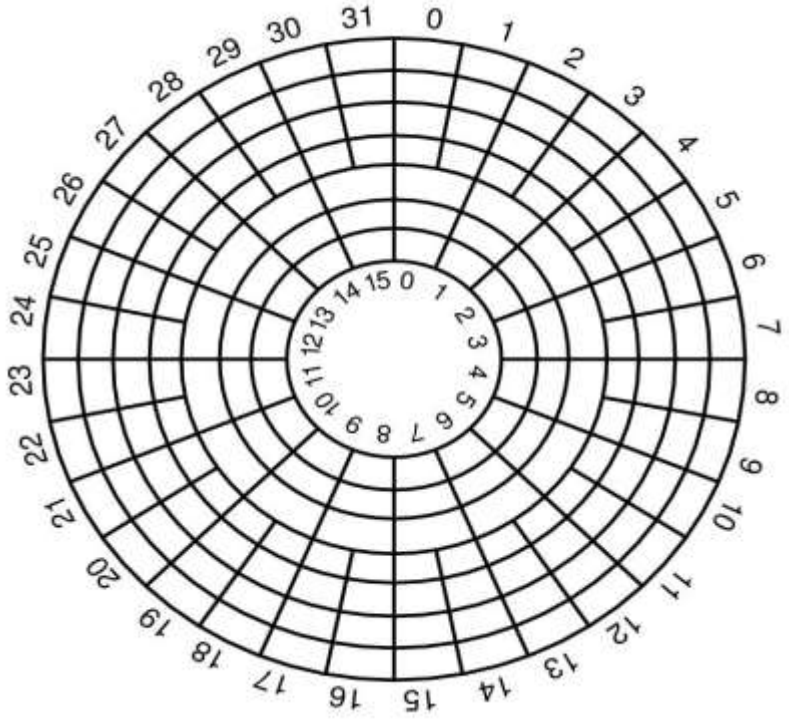
Disk Geometry



Example Disk Characteristics

Parameter	IBM 360-KB floppy disk	WD 18300 hard disk
Number of cylinders	40	10601
Tracks per cylinder	2	12
Sectors per track	9	281 (avg)
Sectors per disk	720	35742000
Bytes per sector	512	512
Disk capacity	360 KB	18.3 GB
Seek time (adjacent cylinders)	6 msec	0.8 msec
Seek time (average case)	77 msec	6.9 msec
Rotation time	200 msec	8.33 msec
Motor stop/start time	250 msec	20 sec
Time to transfer 1 sector	22 msec	17 µsec

Disk Surface Geometry



Constant rotation speed

- Want constant bit density

Inner tracks:

- Fewer sectors per track

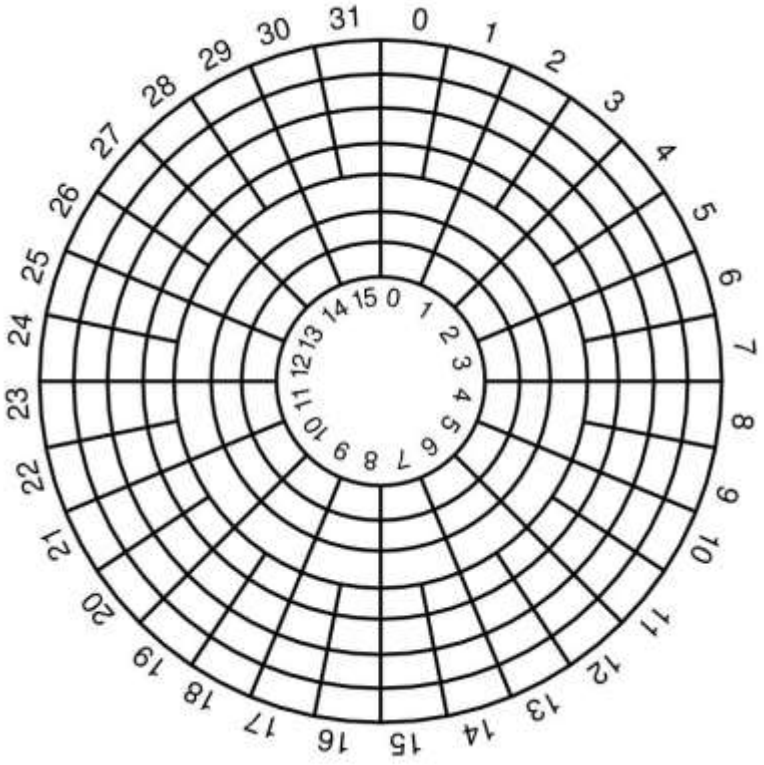
Outer tracks:

- More sectors per track

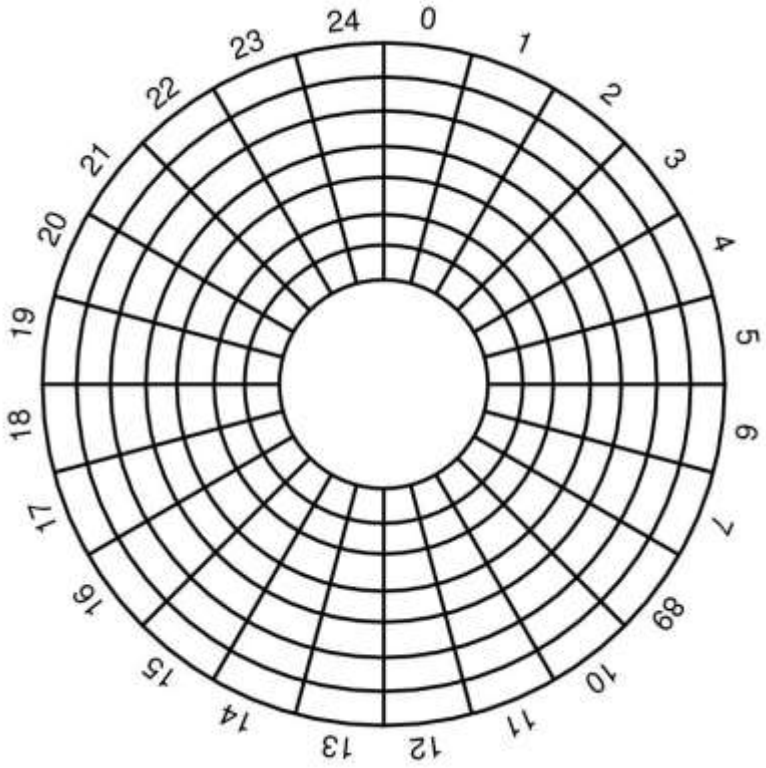
Virtual Geometry

- Physical Geometry
  - *The actual layout of sectors on the disk may be complicated*
  - *The disk controller does the translation*
  - *The CPU sees a “virtual geometry”.*

Virtual Geometry



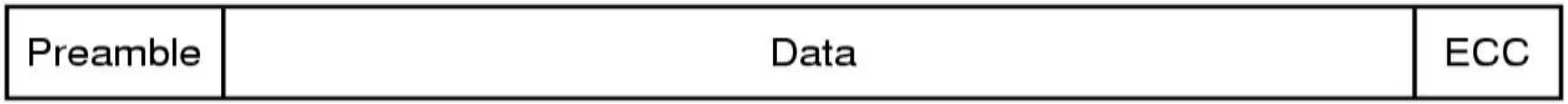
physical geometry



virtual geometry

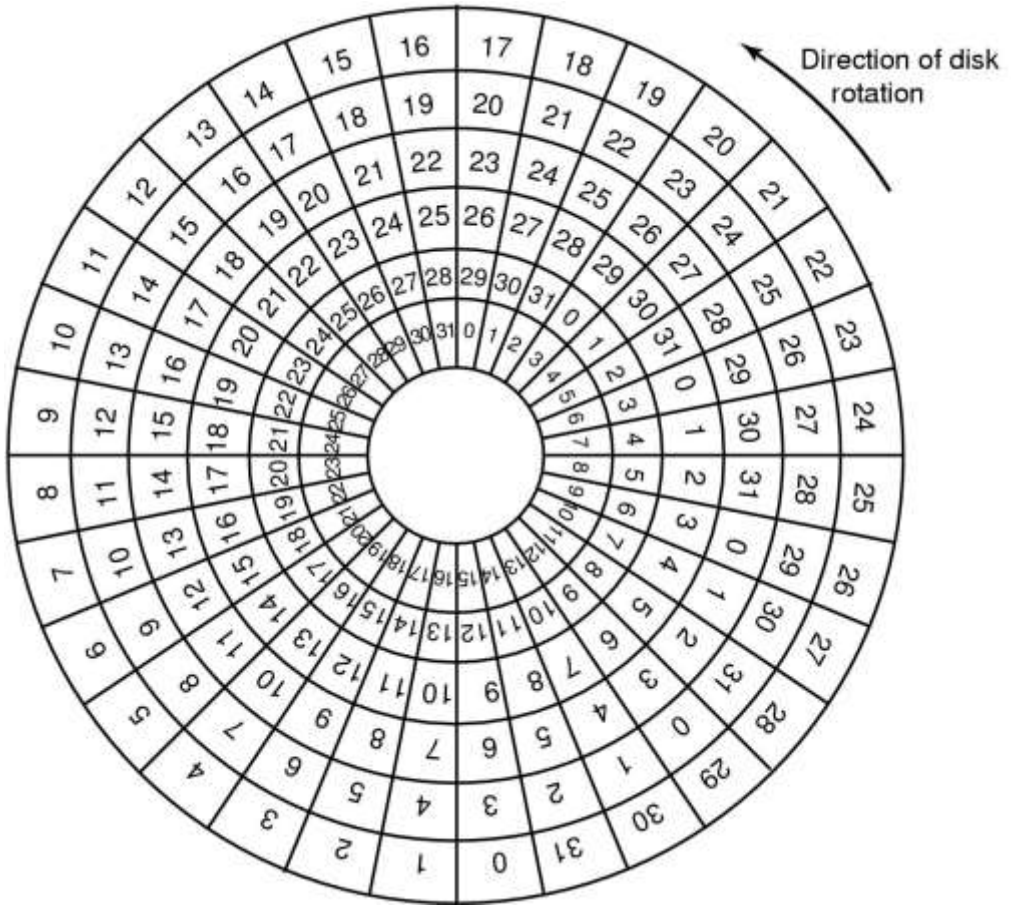
Sector Formatting

- A disk sector

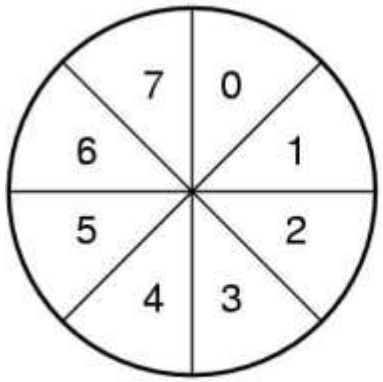


- Typically
  - *512 bytes / sector by 2010, (now about 4KB)*
  - *ECC = 16 bytes*

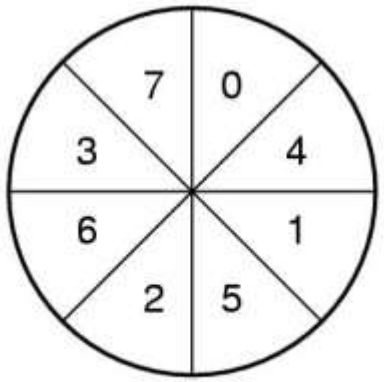
Cylinder Skew



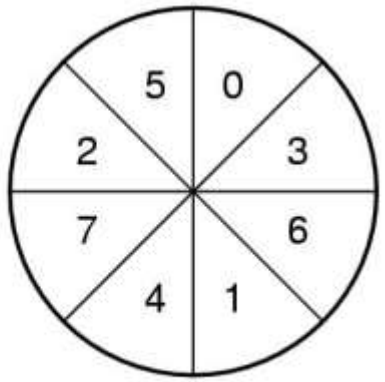
Sector Interleaving



No Interleaving



Single Interleaving

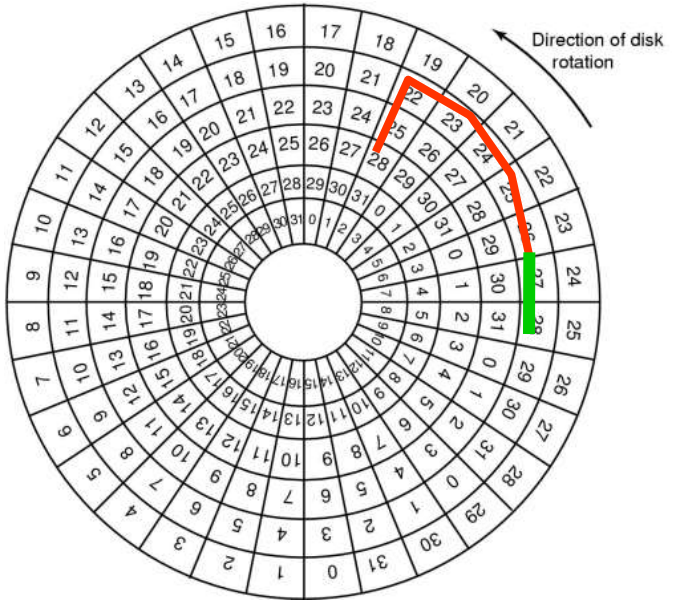


Double Interleaving

Disk Scheduling Algorithms

- Time required to read or write a disk block determined by 3 factors
  - *Seek time*
  - *Rotational delay*
  - *Actual transfer time*

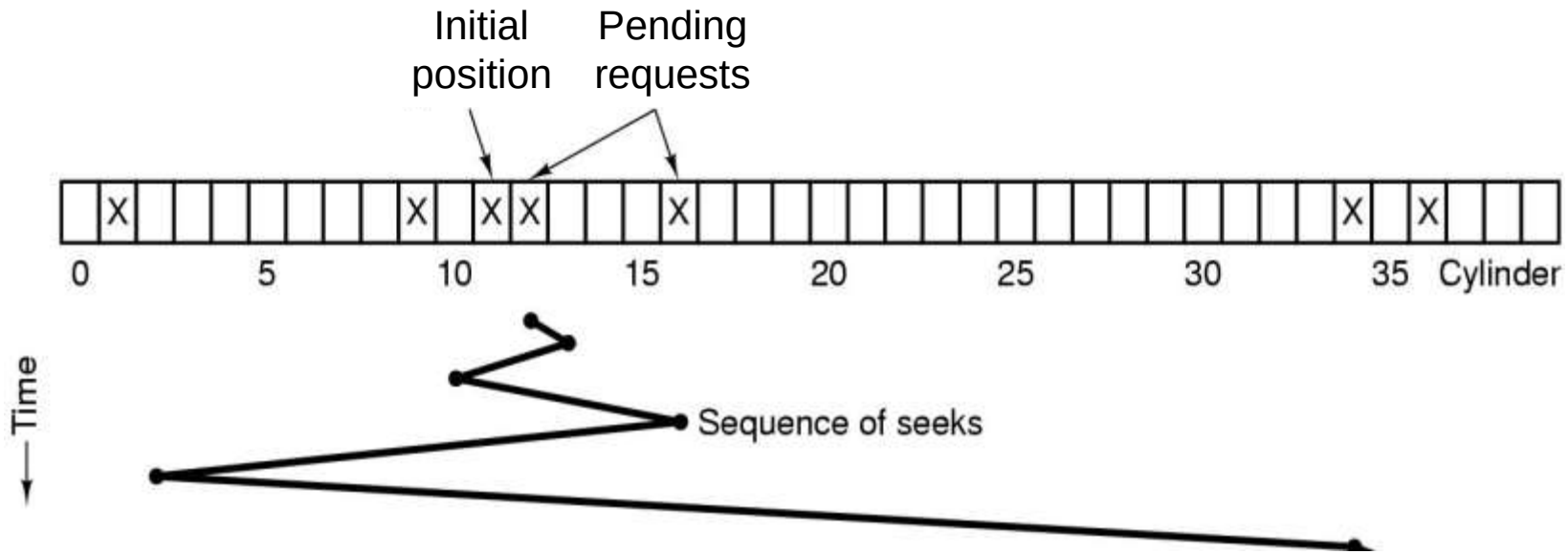
- Seek time dominates
  - *Schedule disk heads to minimize it!*



Disk Scheduling Algorithms

- First-come first serve
- Shortest seek time first
- Scan → back and forth to ends of disk
- C-Scan → only one direction
- Look → back and forth to last request
- C-Look → only one direction

Shortest Seek First (SSF)



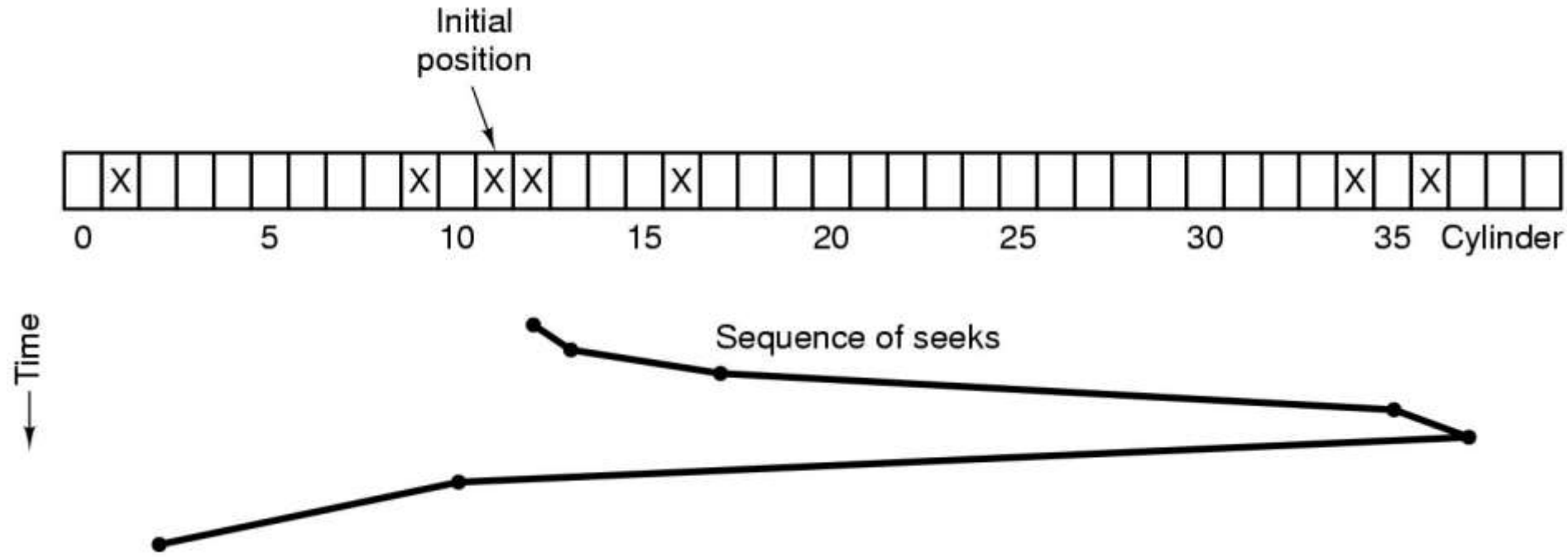
Shortest Seek First (SSF)

- Cuts arm motion in half
- Fatal problem:
  - *Starvation is possible!*

The Elevator Algorithm

- Use one bit to track which direction the arm is moving
  - *Up*
  - *Down*
- Keep moving in that direction
- Service the next pending request in that direction
- When there are no more requests in the current direction, reverse direction

The Elevator Algorithm (SCAN)



Other Algorithms

- First-come first serve
- Shortest seek time first
- Scan → back and forth to ends of disk
- C-Scan → only one direction
- Look → back and forth to last request
- C-Look → only one direction