Chapter 5

Input/Output

5.1 Principles of I/O hardware
5.2 Principles of I/O software
5.3 I/O software layers
5.4 I/O devices: Disks, Character-oriented terminals, Clocks

5.1 Principles of I/O hardware

Principles of I/O Hardware
Types of I/O devices

Two main groups: Block and Character Devices

- Block devices include disk drives
  - Commands include read, write, seek
  - Raw I/O or file-system access
  - Memory-mapped file access possible
- Character devices include keyboards, mice, serial ports
  - Commands include `get, put`
  - Libraries layered on top allow line editing

Principles of I/O Hardware
Types of I/O devices

Some typical device, network, and data base rates

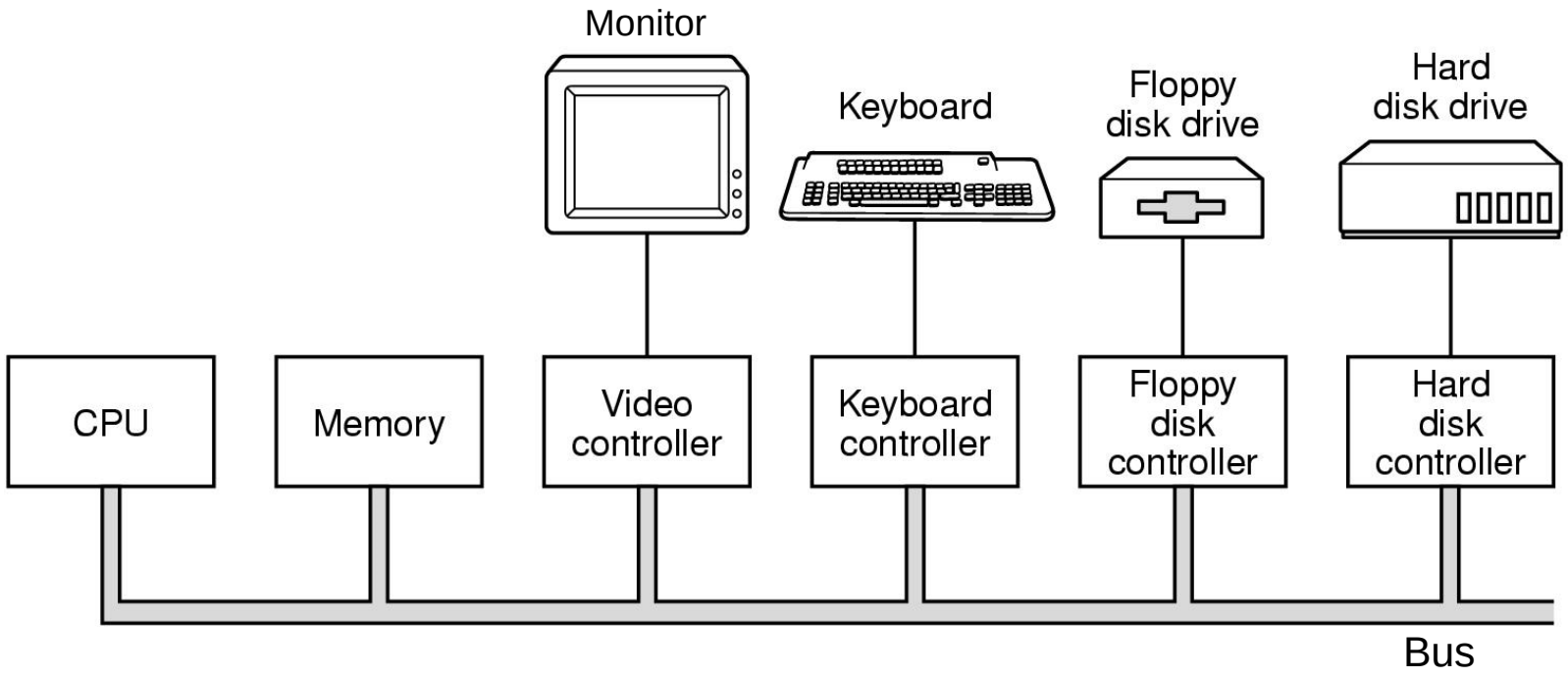
Device	Data rate
Keyboard	10 bytes/sec
Mouse	100 bytes/sec
56K modem	7 KB/sec
Telephone channel	8 KB/sec
Dual ISDN lines	16 KB/sec
Laser printer	100 KB/sec
Scanner	400 KB/sec
Classic Ethernet	1.25 MB/sec
USB (Universal Serial Bus)	1.5 MB/sec
Digital camcorder	4 MB/sec
IDE disk	5 MB/sec
40x CD-ROM	6 MB/sec
Fast Ethernet	12.5 MB/sec
ISA bus	16.7 MB/sec
EIDE (ATA-2) disk	16.7 MB/sec
FireWire (IEEE 1394)	50 MB/sec
XGA Monitor	60 MB/sec
SONET OC-12 network	78 MB/sec
SCSI Ultra 2 disk	80 MB/sec
Gigabit Ethernet	125 MB/sec
Ultrium tape	320 MB/sec
PCI bus	528 MB/sec
Sun Gigaplane XB backplane	20 GB/sec

Principles of I/O Hardware
Common concepts

Common concepts

- I/O Device Controller
- I/O Port
- I/O Bus

Principles of I/O Hardware
Device Controllers



- Components of a simple personal computer

Principles of I/O Hardware

Device Controllers

- I/O devices have components:
  - electromechanical component
  - electronic component
- The electronic component is the device controller
  - may be able to handle multiple devices
- Controller's tasks (Disk)
  - convert serial bit stream to block of bytes
  - perform error correction as necessary
  - make available to main memory

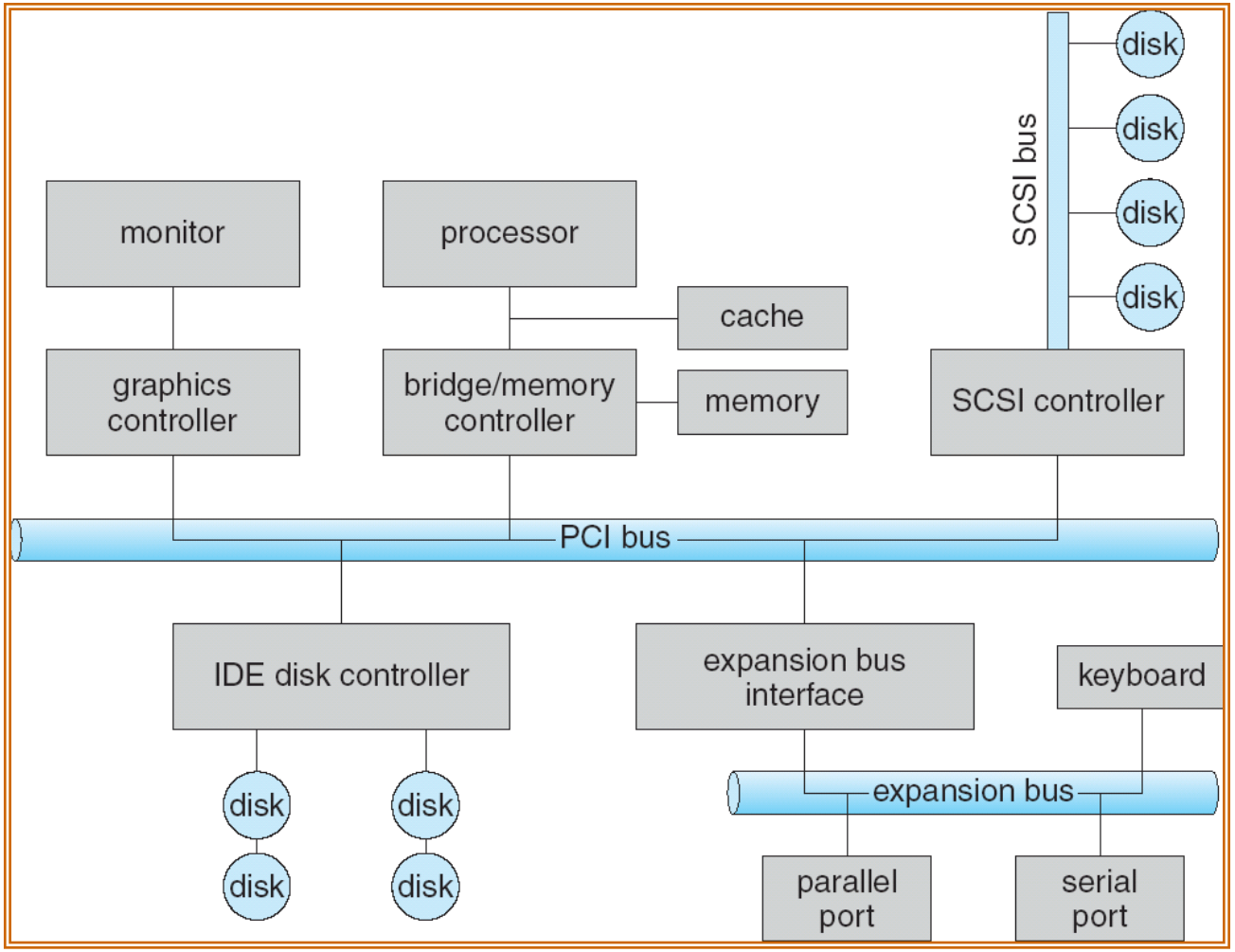
Principles of I/O Hardware
I/O Port

I/O Port is a register in device interface. Example: Device I/O Port Locations on PCs (partial)

I/O address range (hexadecimal)	device
000–00F	DMA controller
020–021	interrupt controller
040–043	timer
200–20F	game controller
2F8–2FF	serial port (secondary)
320–32F	hard-disk controller
378–37F	parallel port
3D0–3DF	graphics controller
3F0–3F7	diskette-drive controller
3F8–3FF	serial port (primary)

Principles of I/O Hardware

A Typical PC Bus Structure

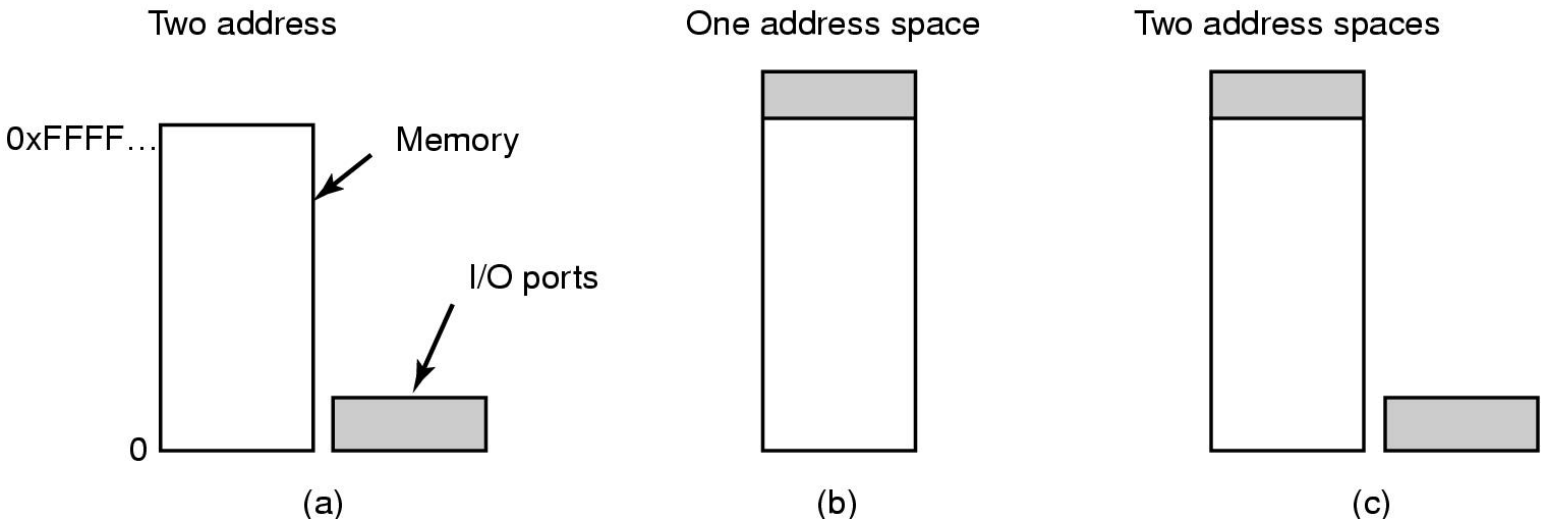


Principles of I/O Hardware

I/O address

- I/O instructions control devices
- Devices have addresses, used by
  - Direct I/O instructions
  - Memory-mapped I/O

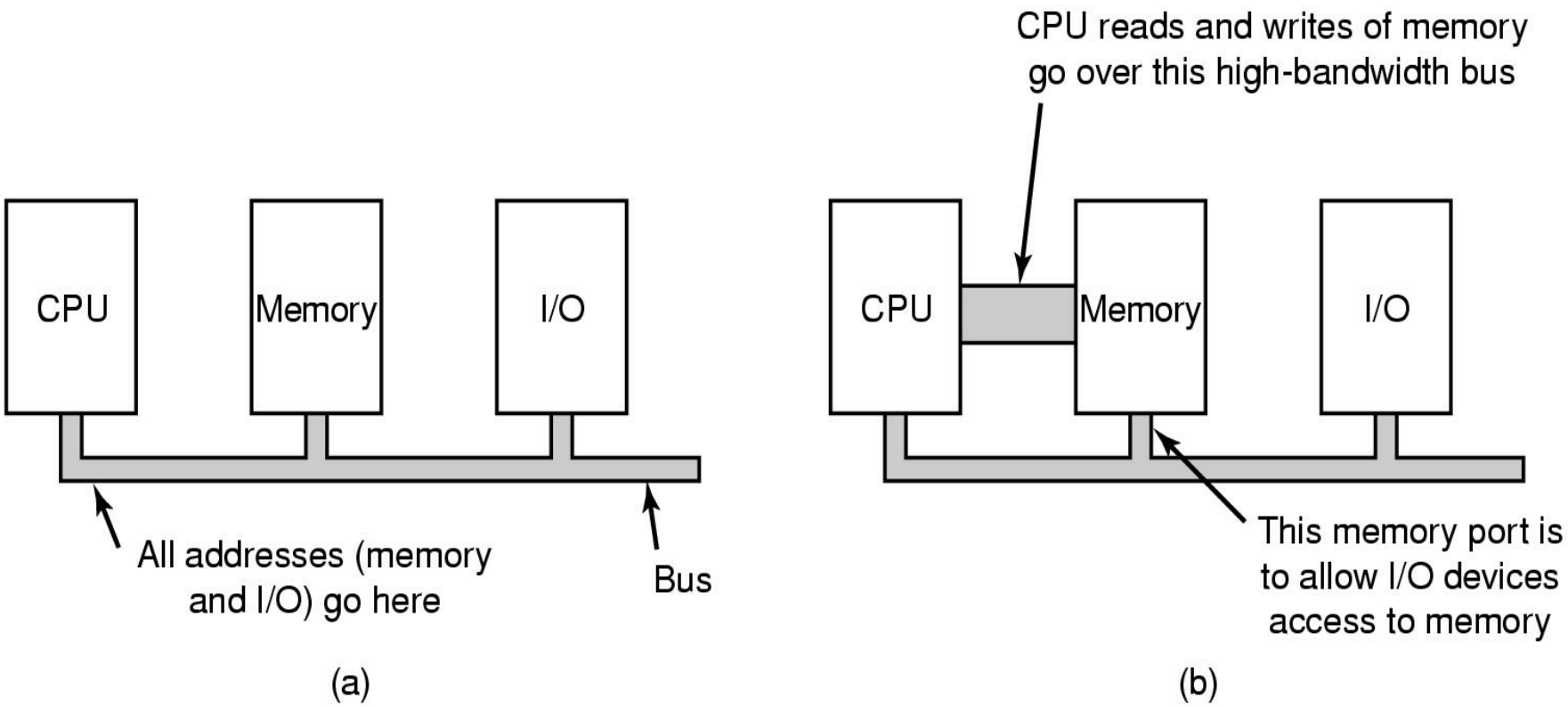
Principles of I/O Hardware
Memory-Mapped I/O (1)



- Separate I/O and memory space
- Memory-mapped I/O
- Hybrid

Principles of I/O Hardware

Memory-Mapped I/O (2)



(a) A single-bus architecture

(b) A dual-bus memory architecture

Principles of I/O Hardware

Data transfer Method between CPU and I/O device

Three Data I/O transfer Methods:

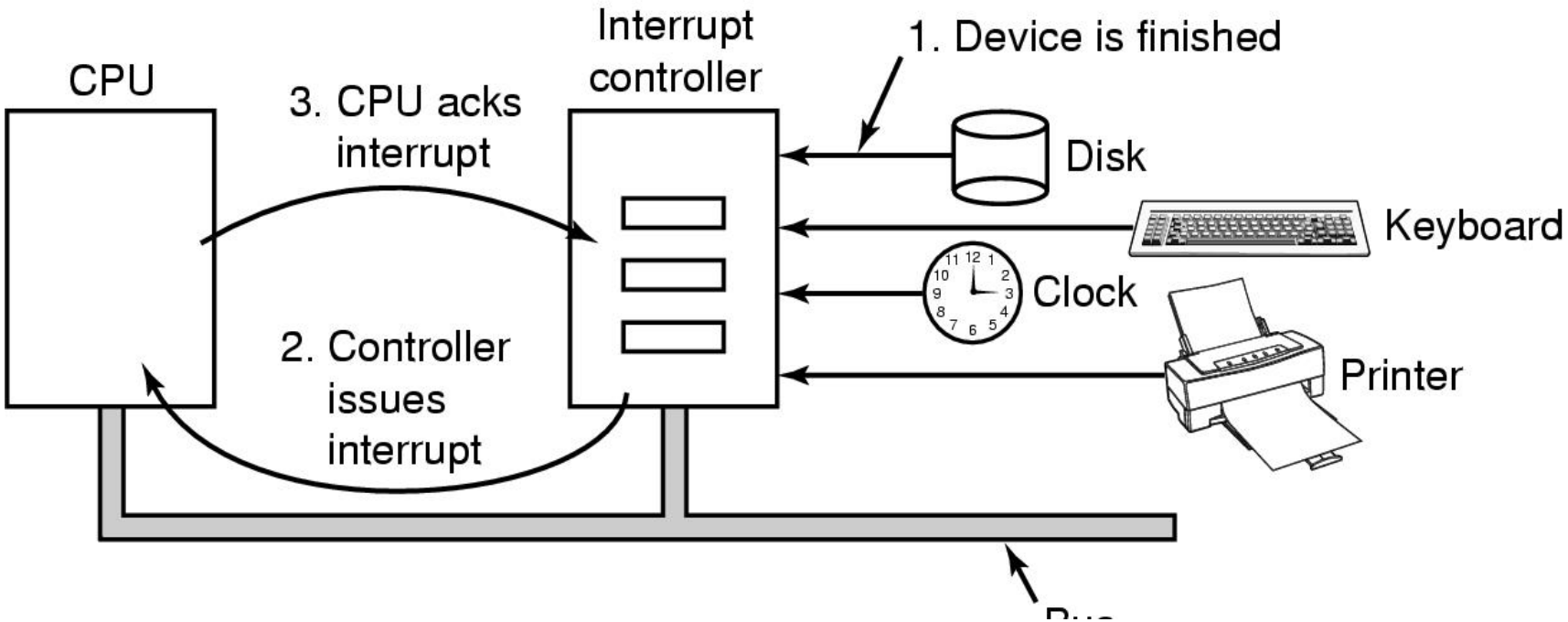
- Programmed I/O
- Interrupt-Driven I/O
- Direct Memory Access

Principles of I/O Hardware

Programmed I/O, Polling

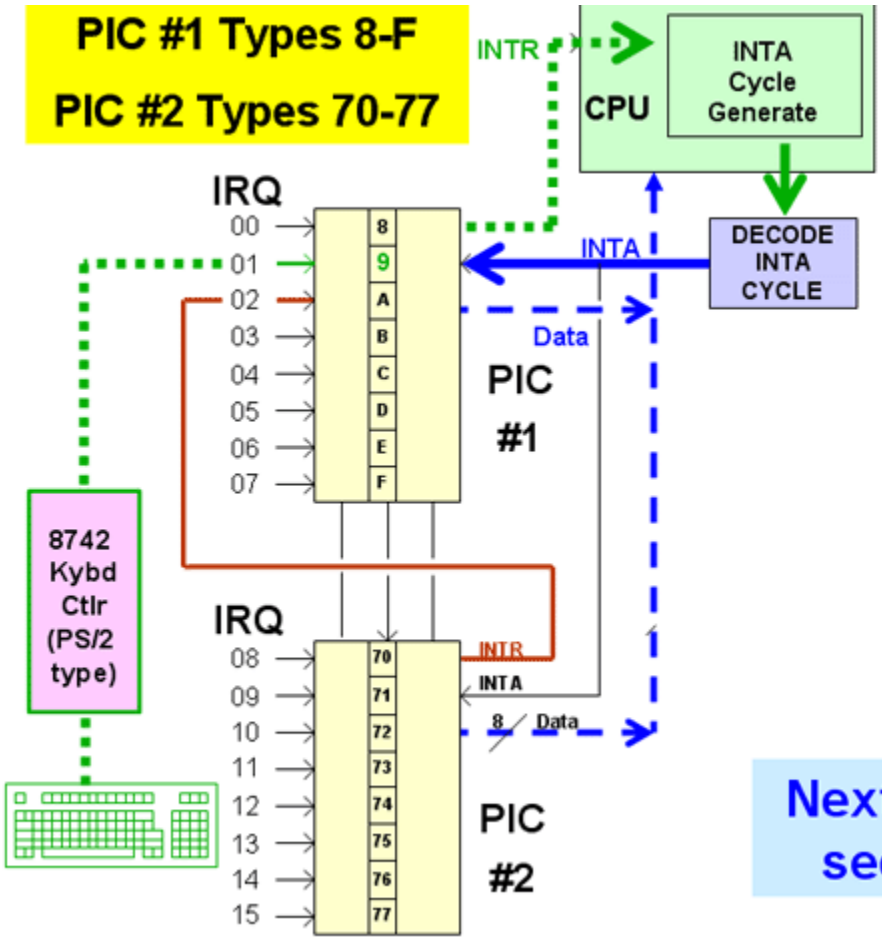
- Determines state of device
  - ready
  - busy
  - Error
- **Busy-wait** cycle to wait for I/O from device

Principles of I/O Hardware
Interrupt-Driven I/O



How interrupts happens. Connections between devices and interrupt controller actually use interrupt lines on the bus rather than dedicated wires

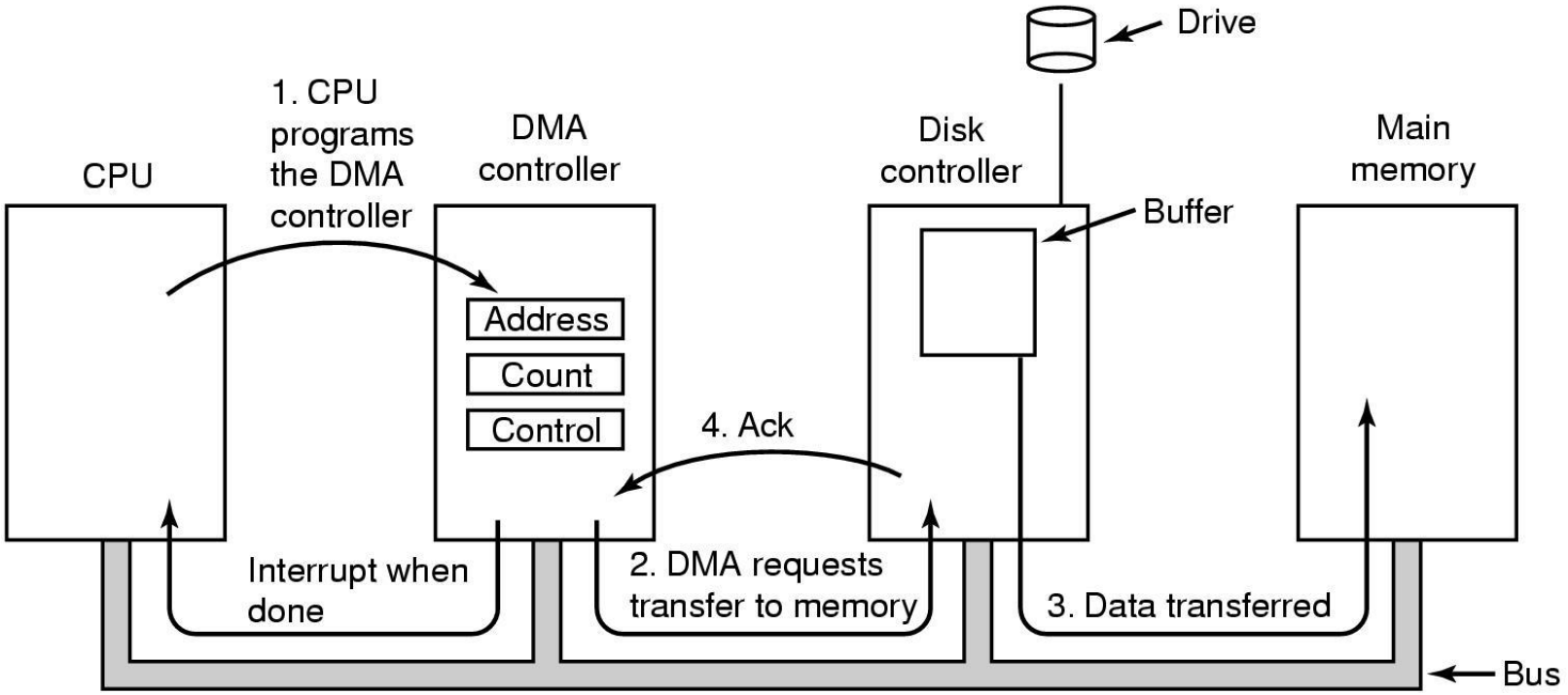
Principles of I/O Hardware
Example: Interrupts of PC computer



INT. NO	Interrupt Source
NMI	Parity Error Detected
IRQ0	Interval Timer (PIT)
IRQ1	Keyboard Output Buffer
IRQ2	PIC 2 (Cascade)
IRQ8	Real-time Clock Int
IRQ9	S/W Redirected To Int 0ah (IRQ2)--free
IRQ10	Free
IRQ11	Free
IRQ12	Free (PS/2 Mouse)
IRQ13	reserved--math coprocessor
IRQ14	Pri EIDE Fixed Disk
IRQ15	Sec EIDE Fixed Disk
IRQ3	COM2
IRQ4	COM1 (Primary)
IRQ5	Free (Sound Card, LPT2)
IRQ6	Floppy Disk Controller
IRQ7	LPT1 (Primary) Or Free

Next two slides show sequence of steps

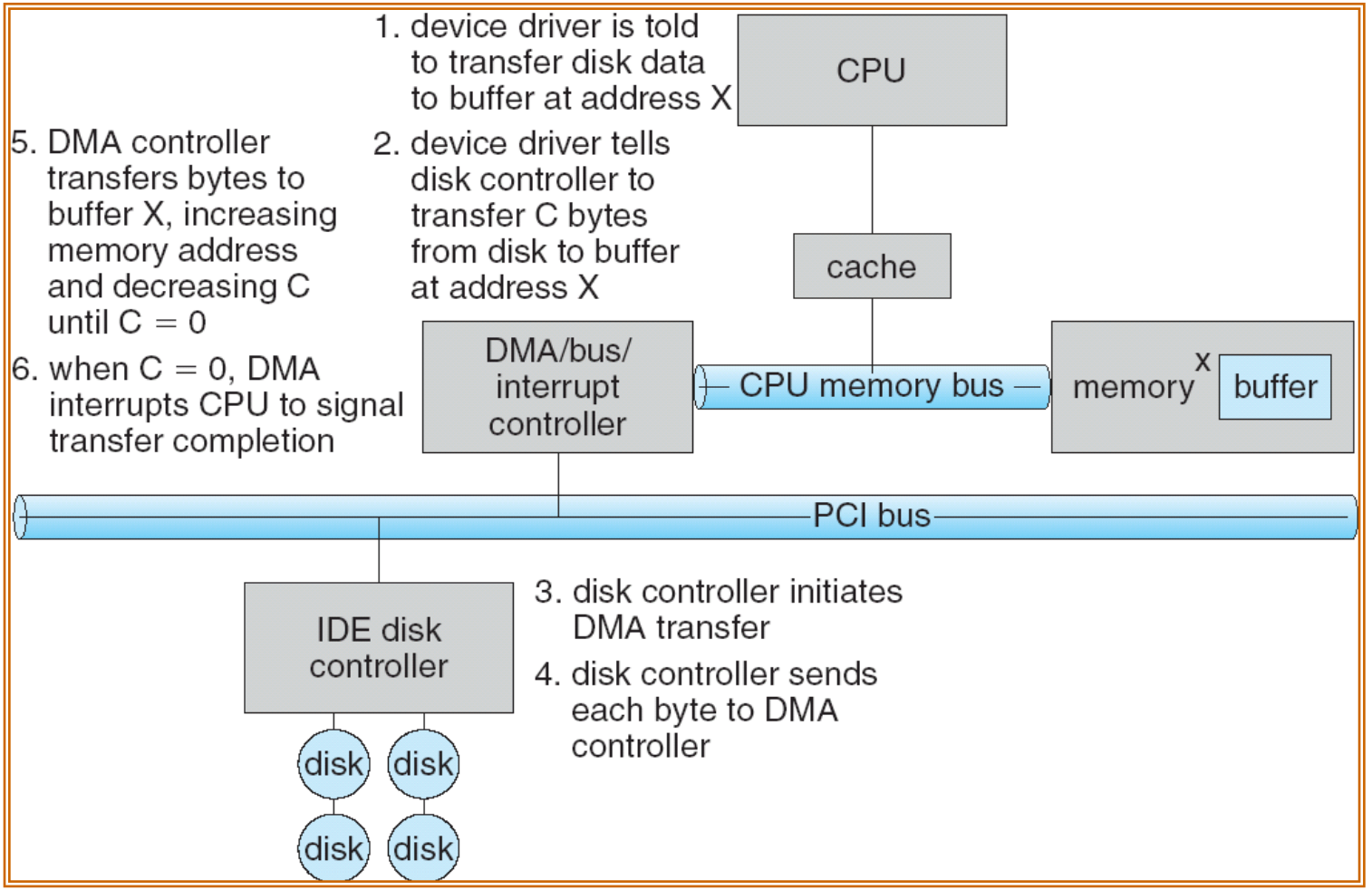
Principles of I/O Hardware
Direct Memory Access (DMA)



Operation of a DMA transfer

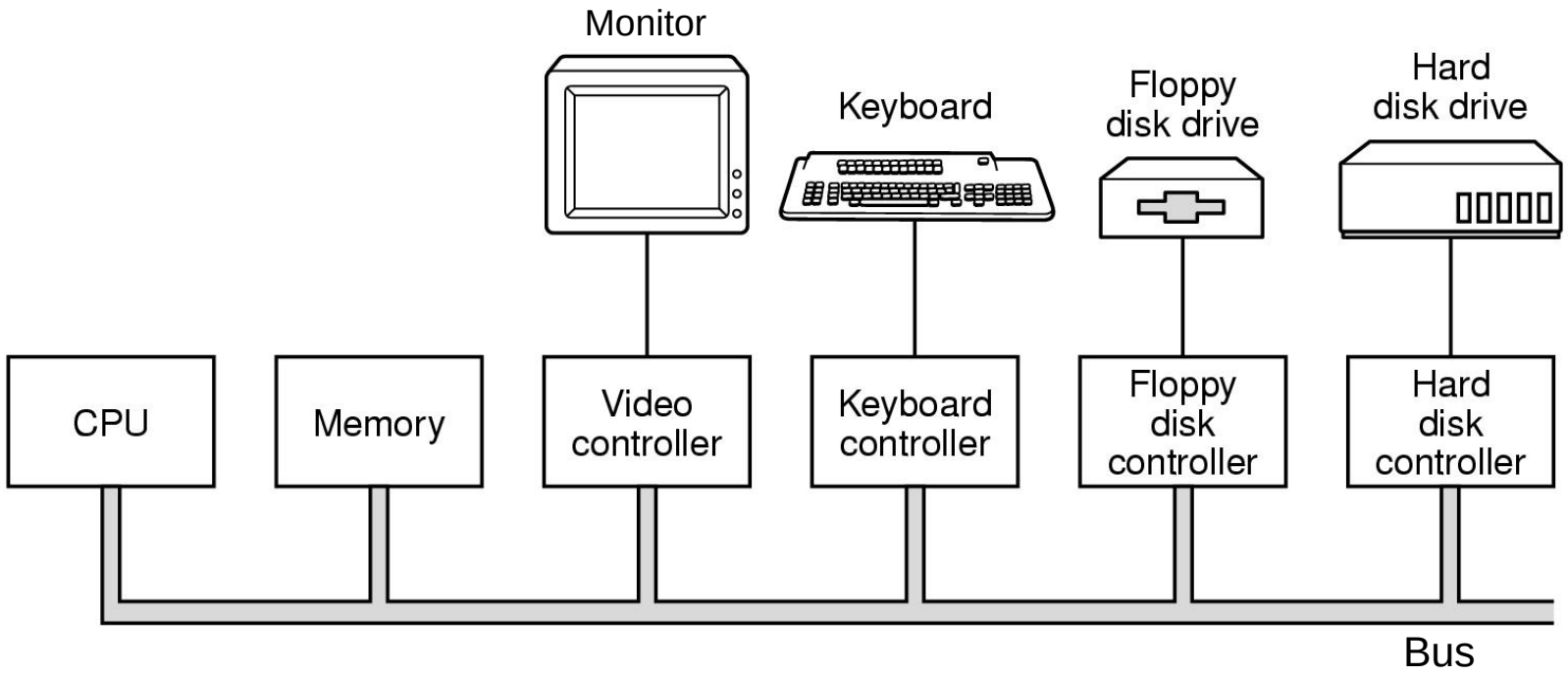
Principles of I/O Hardware

Example: DMA Transfer in PC Computer



5.2 Principles of I/O software

Principles of I/O Hardware
Device Controllers



Principles of I/O Software
Goals of I/O Software (1)

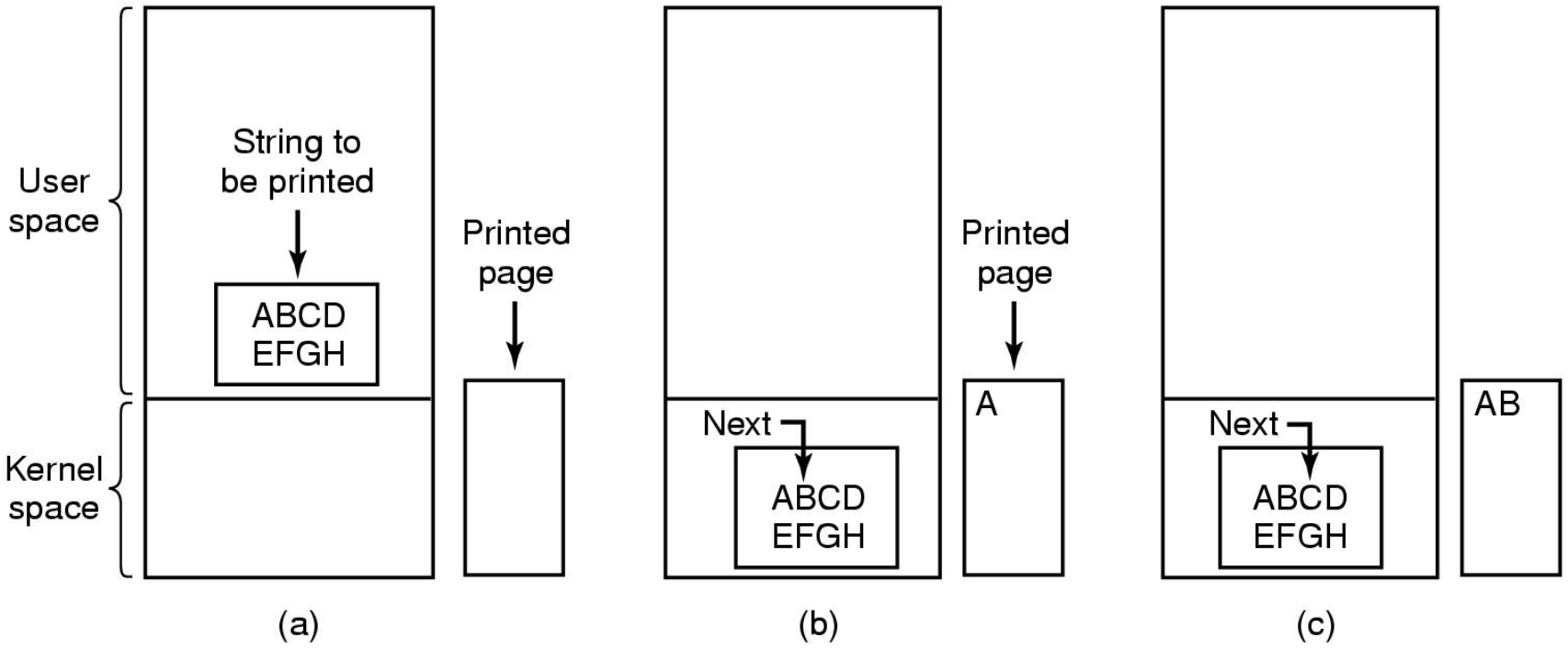
- Device independence
  - programs can access any I/O device
  - without specifying device in advance
    (floppy, hard drive, or CD-ROM)
- Uniform naming
  - name of a file or device is a string or an integer
  - not depending on which machine
- Error handling
  - handle as close to the hardware as possible

Principles of I/O Software
Goals of I/O Software (2)

- Synchronous vs. asynchronous transfers
  - blocked transfers vs. interrupt-driven
- Buffering
  - data coming off a device cannot be stored in final destination
- Sharable vs. dedicated devices
  - disks are sharable
  - tape drives would not be

Principles of I/O Software

Programmed I/O (1)



Steps in printing a string

Principles of I/O Software
Programmed I/O (2)

```
copy_from_user(buffer, p, count);                /* p is the kernel bufer */
for (i = 0; i < count; i++) {                    /* loop on every character */
        while (*printer_status_reg != READY) ;   /* loop until ready */
        *printer_data_register = p[i];           /* output one character */
}
return_to_user( );
```

Writing a string to the printer using programmed I/O

Principles of I/O Software
Interrupt-Driven I/O

```
copy_from_user(buffer, p, count);
enable_interrupts( );
while (*printer_status_reg != READY) ;
*printer_data_register = p[0];
scheduler( );
```

(a)

```
if (count == 0) {
    unblock_user( );
} else {
    *printer_data_register = p[i];
    count = count – 1;
     i = i + 1;
}
acknowledge_interrupt( );
return_from_interrupt( );
```

(b)

- Writing a string to the printer using interrupt-driven I/O
  - Code executed when print system call is made
  - Interrupt service procedure

Principles of I/O Software
I/O Using DMA

```
copy_from_user(buffer, p, count);
set_up_DMA_controller( );
scheduler( );
```

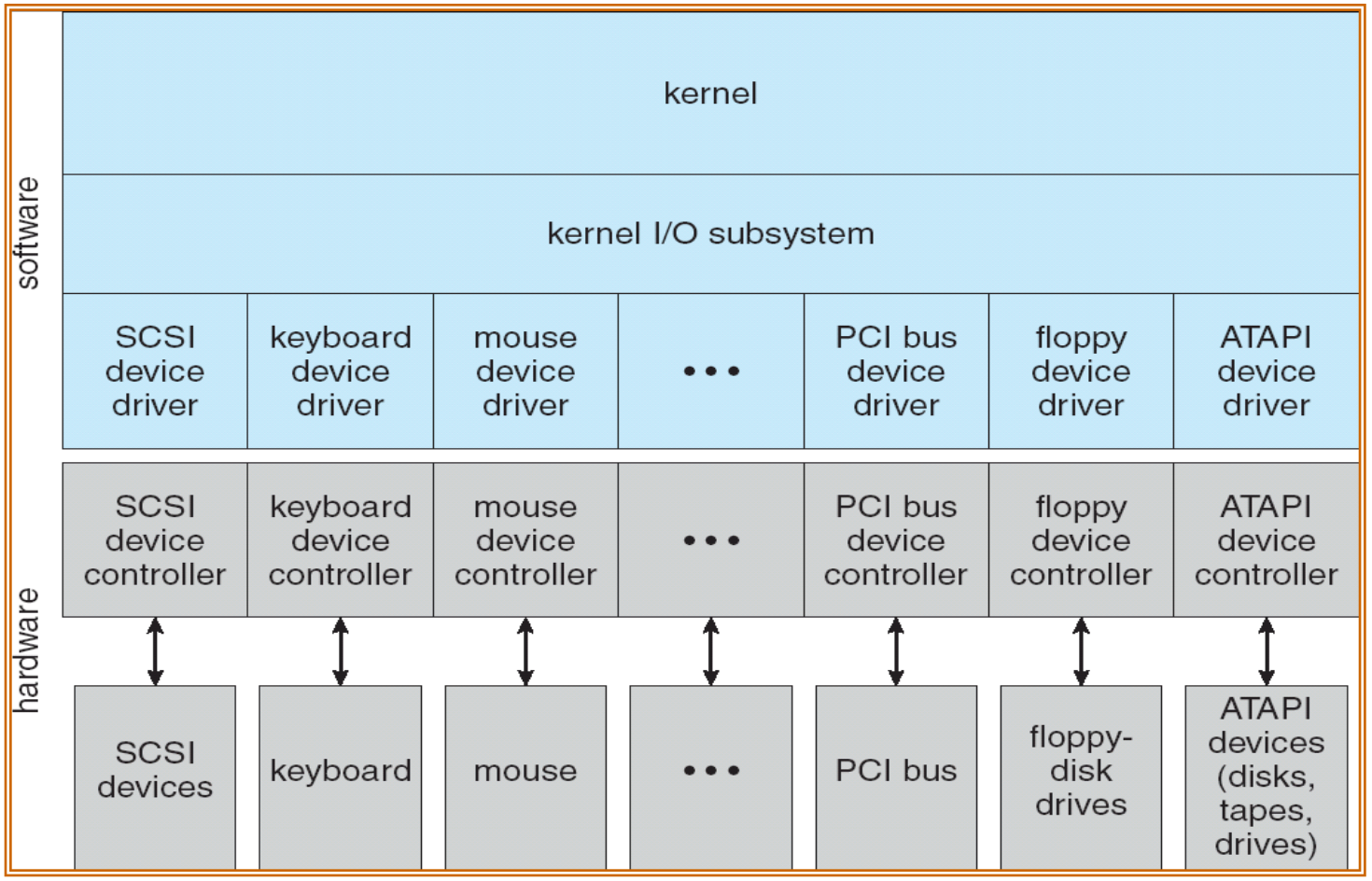
(a)

```
acknowledge_interrupt( );
unblock_user( );
return_from_interrupt( );
```

(b)

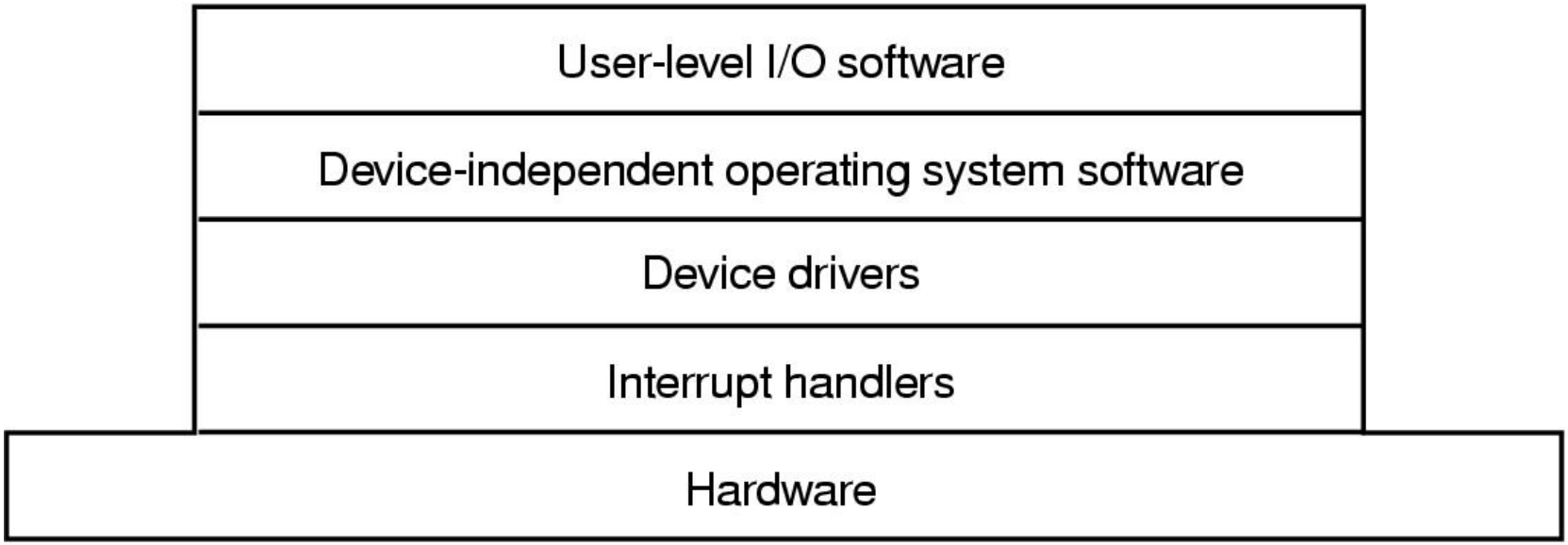
- Printing a string using DMA
  - code executed when the print system call is made
  - interrupt service procedure

A Kernel I/O Structure: Hardware And Software



5.3 I/O software layers

I/O Software Layers



Layers of the I/O Software System

I/O Software Layers
Interrupt Handlers (1)

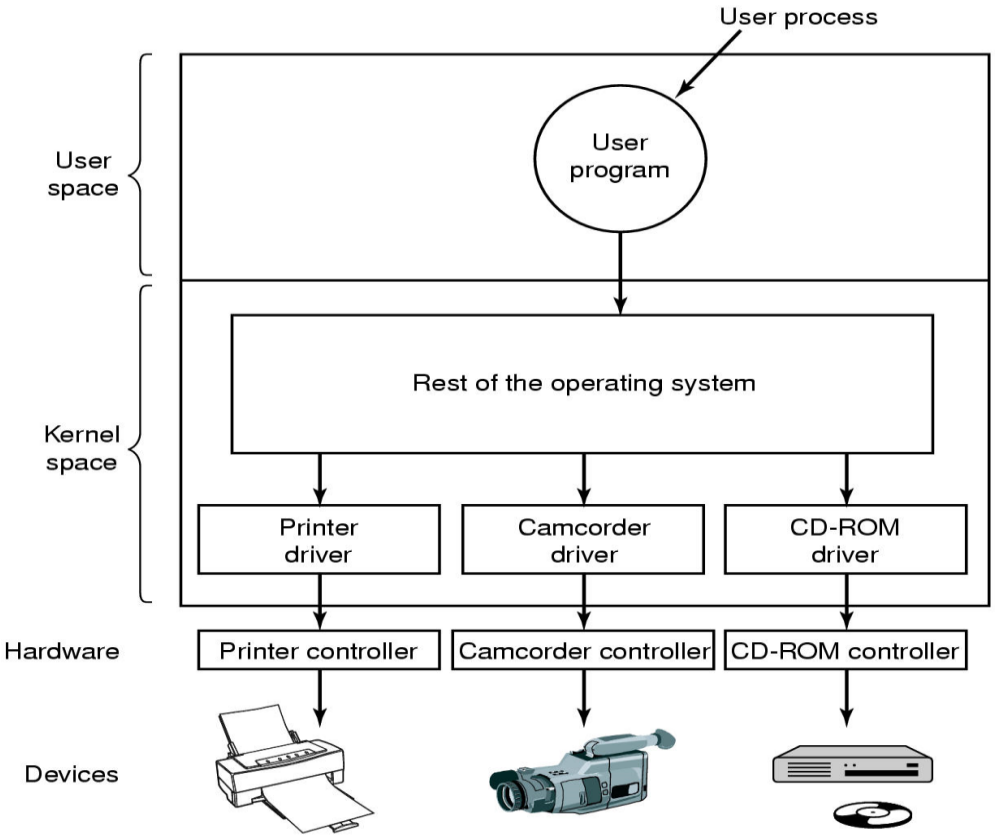
- Interrupt handlers are best hidden
  - have driver starting an I/O operation block until interrupt notifies of completion
- Interrupt procedure does its task
  - then unblocks driver that started it
- Steps must be performed in software after interrupt completed
  1. Save regs not already saved by interrupt hardware
  2. Set up context for interrupt service procedure

I/O Software Layers
Interrupt Handlers (2)

3. Set up stack for interrupt service procedure
4. Ack interrupt controller, reenable interrupts
5. Copy registers from where saved to process table
6. Run service procedure
7. Set up MMU context for process to run next
8. Load new process' registers
9. Start running the new process

I/O Software Layers

Device Drivers



- Logical position of device drivers is shown here
- Communications between drivers and device controllers goes over the bus

I/O Software Layers

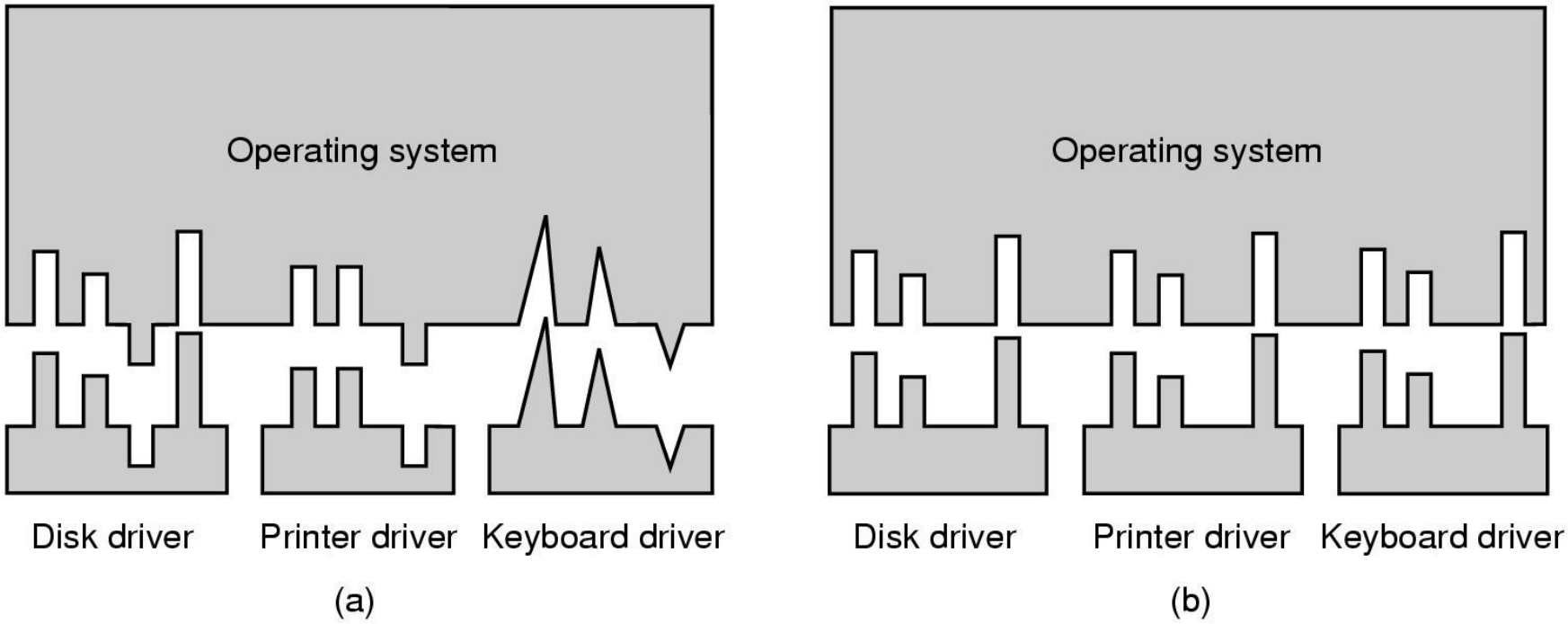
Device-Independent I/O Software (1)

Uniform interfacing for device drivers
Buffering
Error reporting
Allocating and releasing dedicate devices
Providing a device-independent block size

Functions of the device-independent I/O software

I/O Software Layers

Device-Independent I/O Software (2)

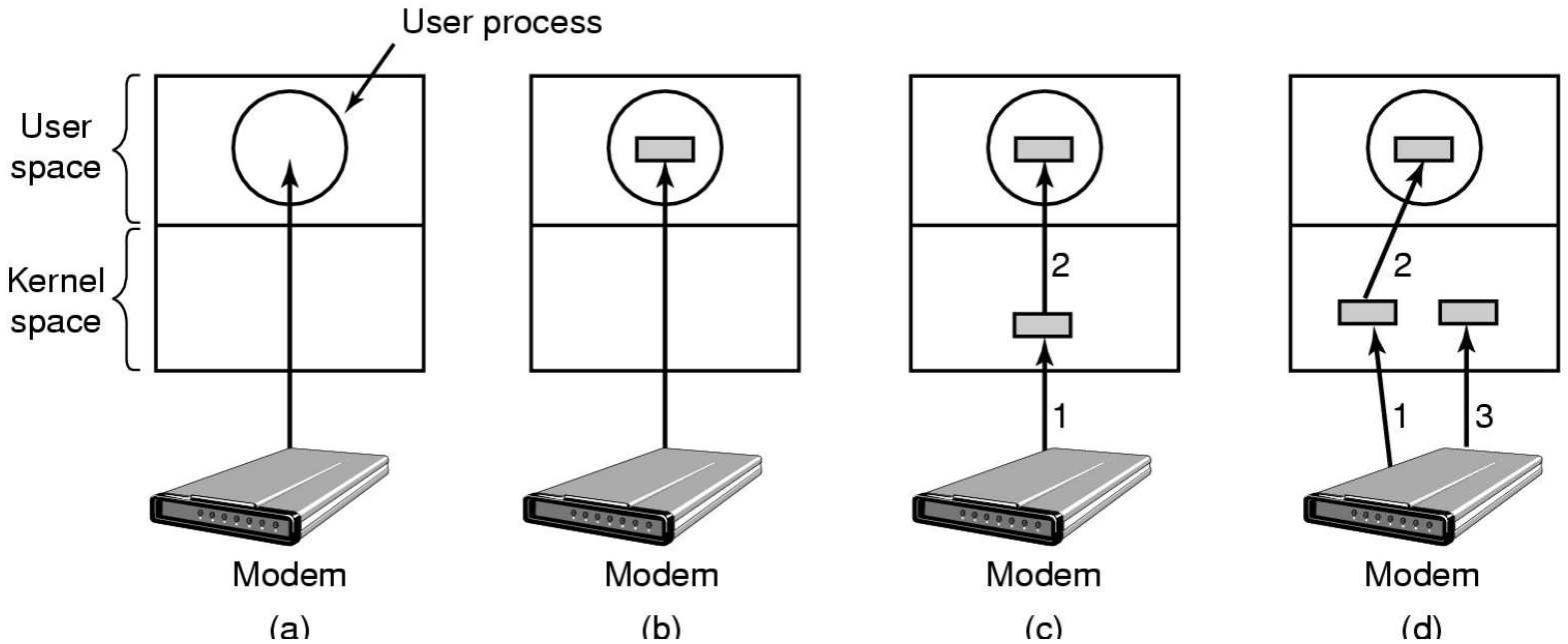


(a) Without a standard driver interface

(b) With a standard driver interface

I/O Software Layers

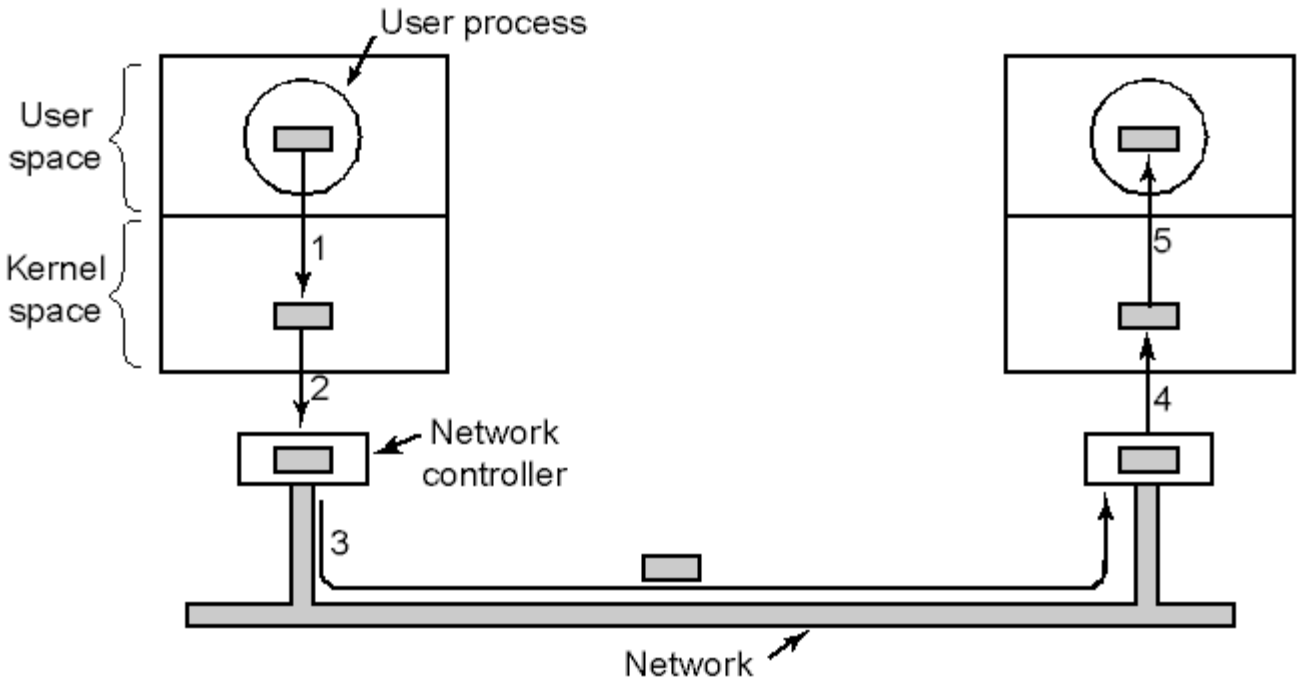
Device-Independent I/O Software (3)



(a) Unbuffered input
(b) Buffering in user space
(c) Buffering in the kernel followed by copying to user space
(d) Double buffering in the kernel

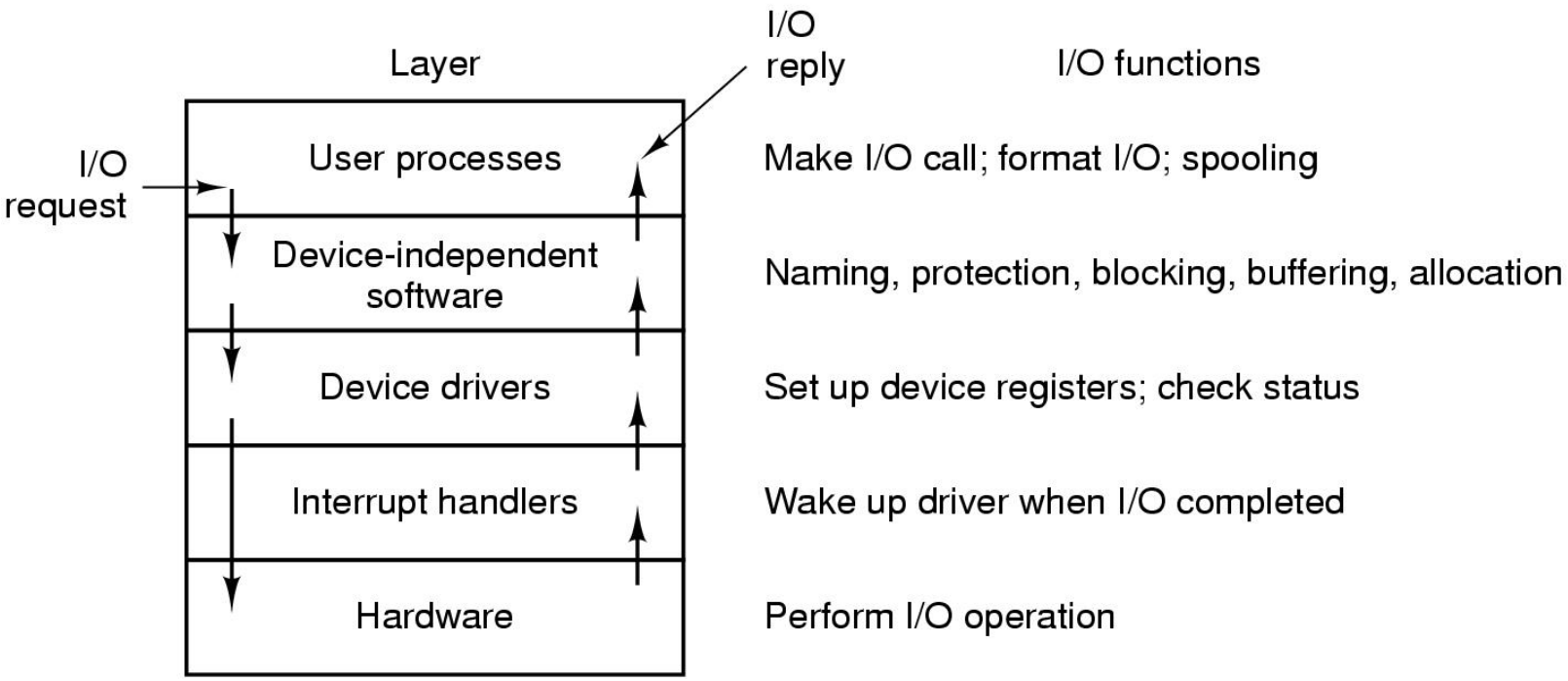
I/O Software Layers

Device-Independent I/O Software (4)



Networking may involve many copies

User-Space I/O Software



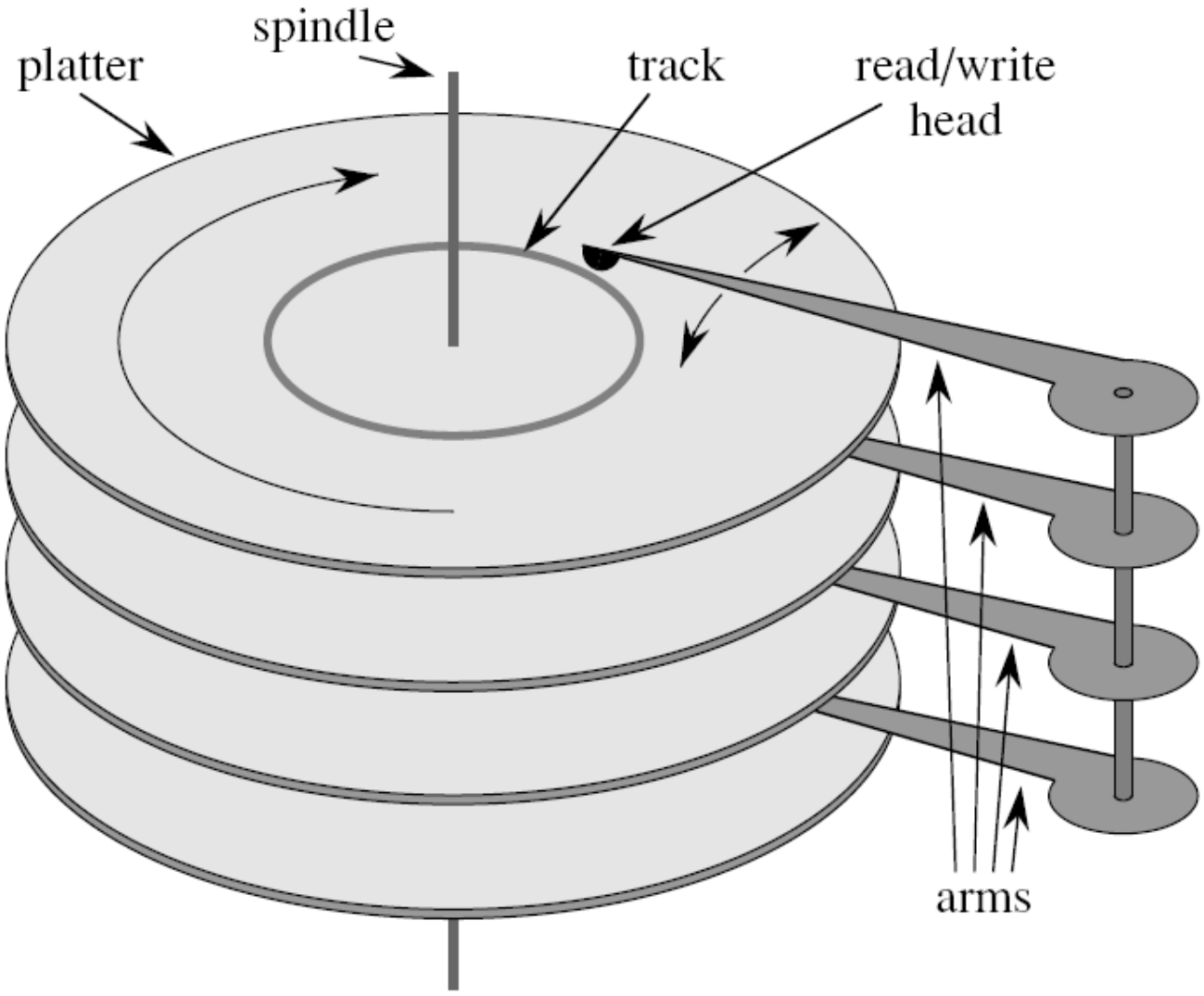
Layers of the I/O system and the main functions of each layer

5.4 I/O devices

I/O devices

- Storage devices: Hard Disks, CD-ROM, CD-R, DVD…
- Display devices: Character-oriented terminals, Graphical user interfaces
- Clocks

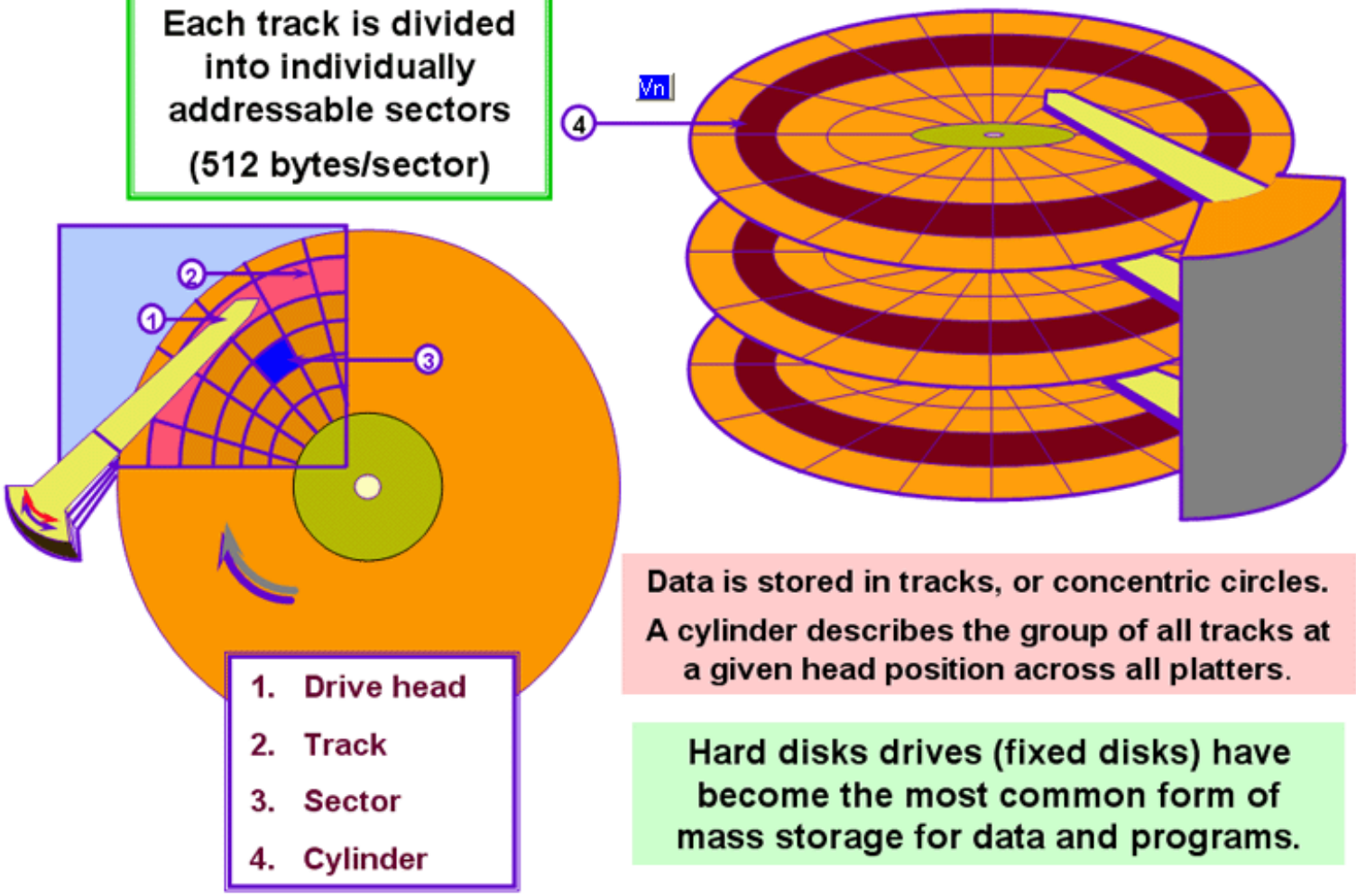
Hard Disk Drive



Disk Hardware (2)

Each track is divided into individually addressable sectors

(512 bytes/sector)



1. Drive head
2. Track
3. Sector
4. Cylinder

Data is stored in tracks, or concentric circles.

A cylinder describes the group of all tracks at a given head position across all platters.

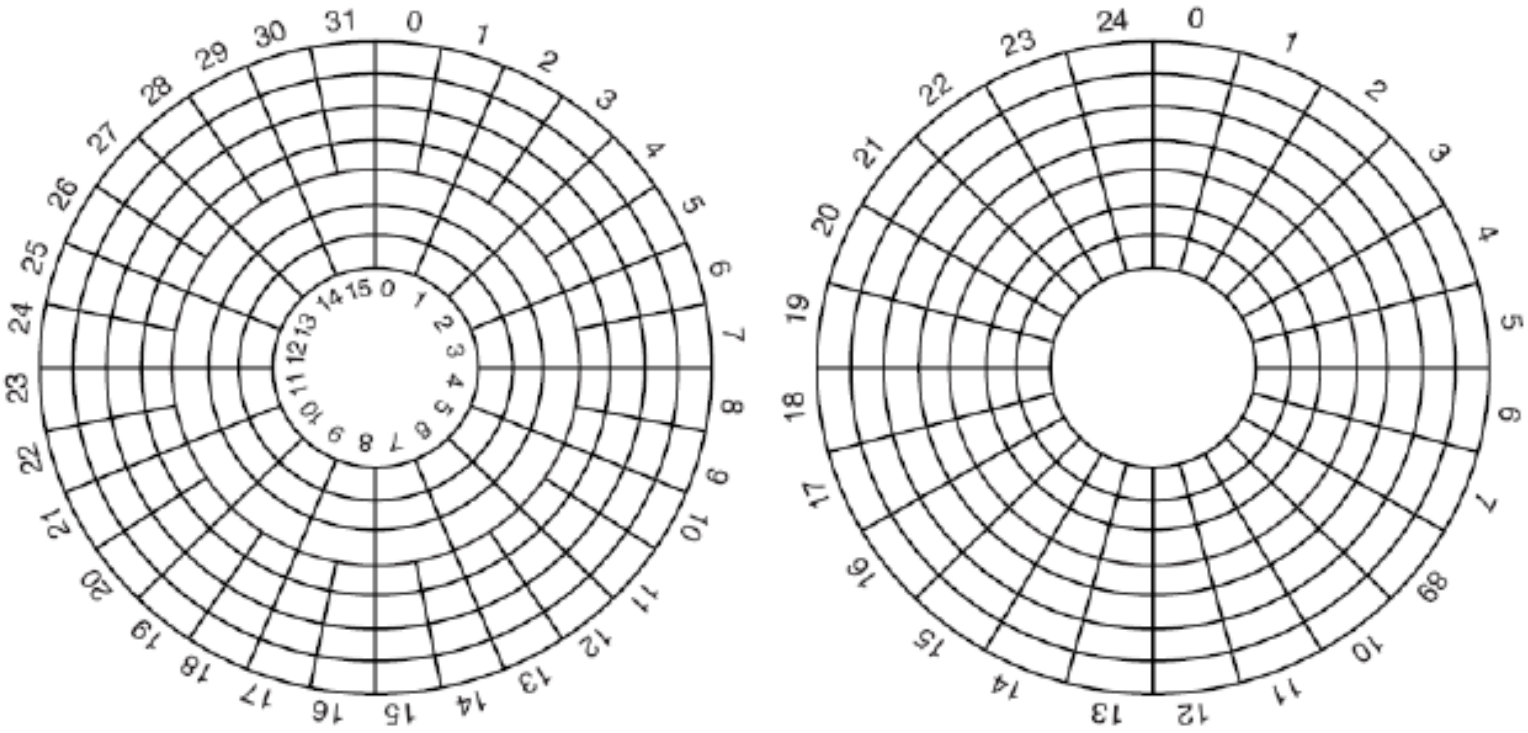
Hard disks drives (fixed disks) have become the most common form of mass storage for data and programs.

Magnetic Disks (1)

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 parameters for the original IBM PC 360-KB floppy disk and a Western Digital WD 18300 hard disk.

Magnetic Disks (2)



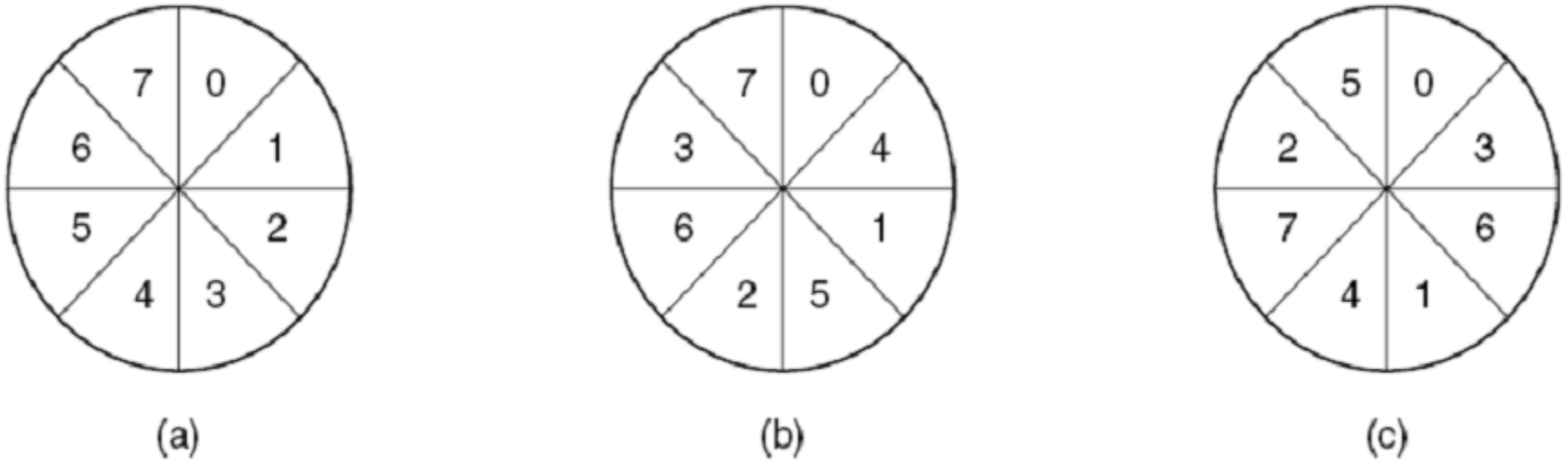
(a) Physical geometry of a disk with two zones.

(b) A possible virtual geometry for this disk.

Disk Formatting (1)

A disk sector.

Disk Formatting (3)

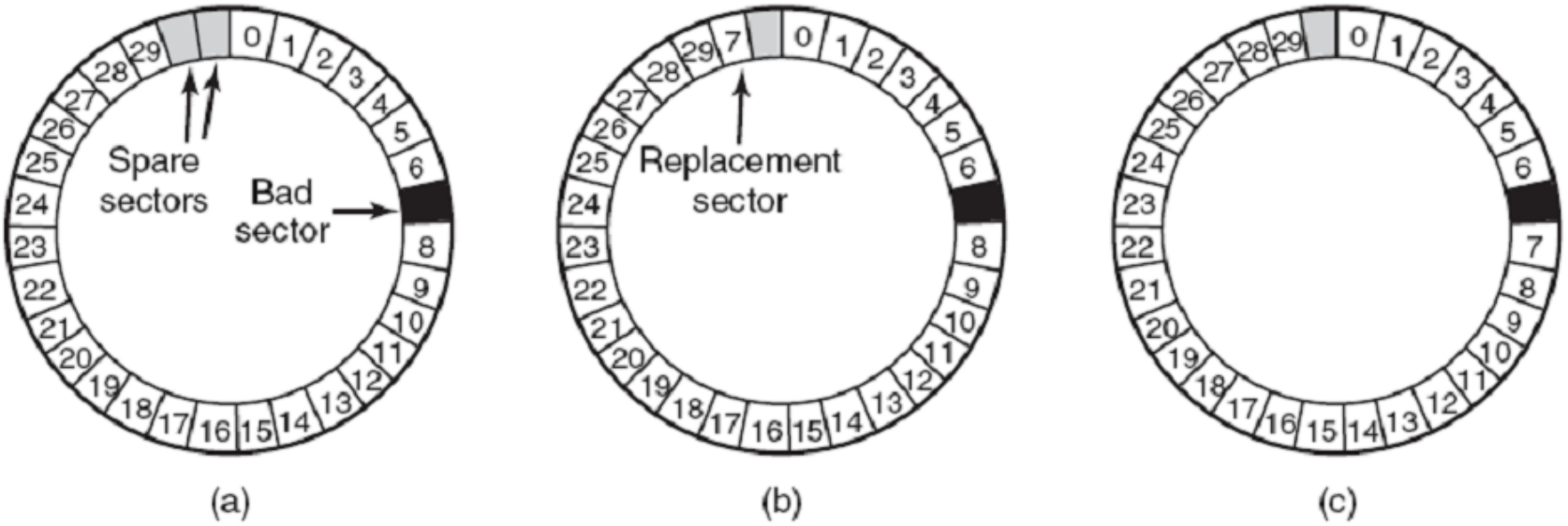


(a) No interleaving.

(b) Single interleaving.

(c) Double interleaving.

Error Handling



(a) A disk track with a bad sector.
(b) Substituting a spare for the bad sector.
(c) Shifting all the sectors to bypass the bad one.

Stable Storage (1)

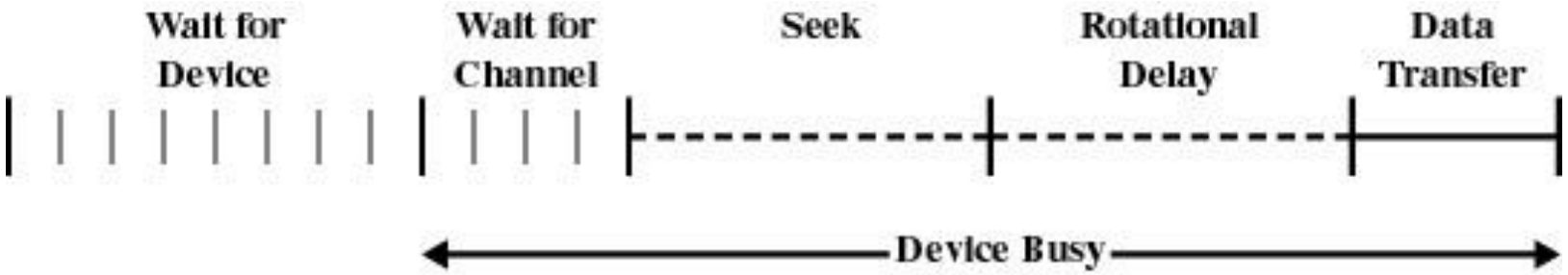
Operations for stable storage using identical disks:

1. Stable writes
2. Stable reads
3. Crash recovery

Disk Performance Parameters

- To read or write, disk head must be positioned at desired track and at beginning of desired sector
- Read/write time factors:
  1. Seek time: time it takes to position the head at desired track
  2. Rotational delay or rotational latency: time its takes for beginning of sector to reach head
  3. Actual data transfer time

Timing of a Disk I/O Transfer



Disk Performance Parameters

- Transfer time
  - data transfer occurs as the disk moves under head
  - depends on rotational speed of disk

$$Tt = b/rN$$

T_t = transfer time

b = no of bytes to transfer

N = no of bytes on a track

r = rotational speed (rps)

HDD Access Time

- Sum of seek time, rotational delay and transfer time

$$Ta = Ts + Tr + Tt$$

- Average seek time → *Ts*
- Time for half a revolution → *Tr*

$$Ta = Ts + 0.5/r + b/rN$$

assuming that all data to be read is on the same track, i.e.
no head re-positioning is required

Disk Scheduling Algorithms

- For a single disk there will be a number of I/O requests
- If requests are selected randomly we will get the worst possible performance
  - seek time will dominate
- FCFS will not be entirely random due to locality

Disk Scheduling Algorithms

Name	Description	Remarks
	Selection according to requestor	
RSS	Random scheduling	For analysis and simulation
FIFO	First in first out	Fairest of them all
PRI	Priority by process	Control outside of disk queue management
LIFO	Last in first out	Maximize locality and resource utilization
	Selection according to requested item:	
SSTF	Shortest service time first	High utilization, small queues
SCAN	Back and forth over disk	Better service distribution
C-SCAN	One way with fast return	Lower service variability
N-step-SCAN	SCAN of N records at a time	Service guarantee
FSCAN	N-step-SCAN with N = queue size at beginning of SCAN cycle	Load-sensitive

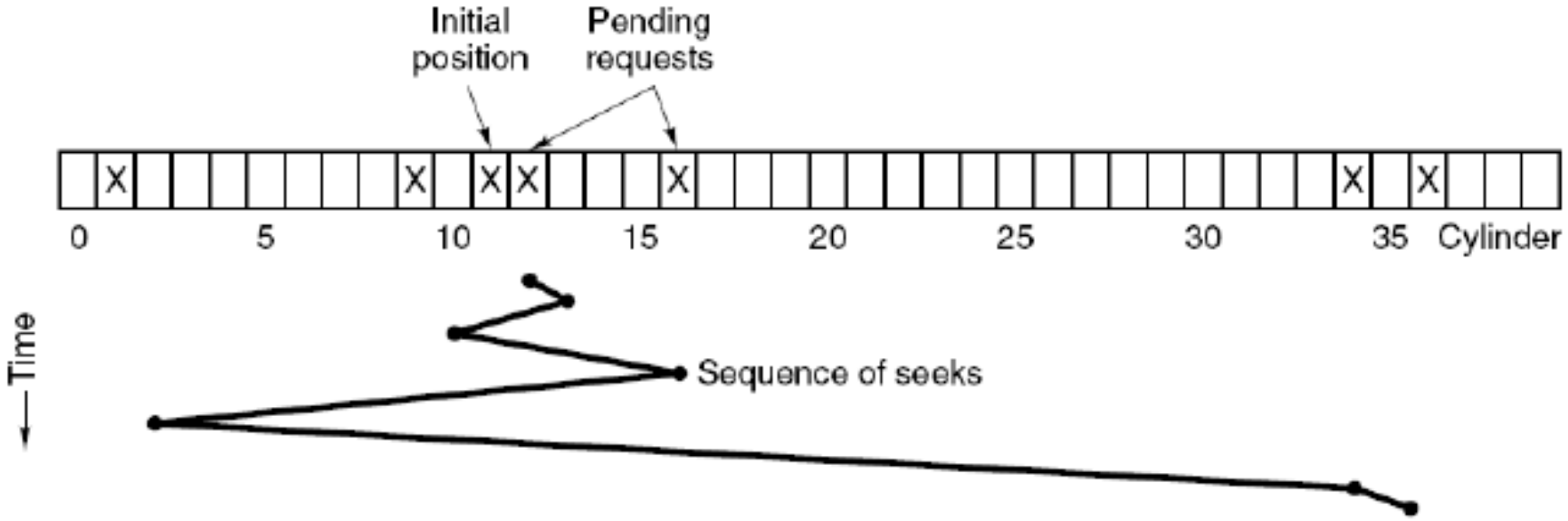
Disk Scheduling Algorithms
First In First Out (FIFO, FCFS)

- simplest form of scheduling
- Each request processed in arrival order
- Advantages
  - simple and fair
  - no starvation
    - unless a single request is extremely long.
- Disadvantages
  - stupid and slow.

Disk Scheduling Algorithms
Shortest Seek Time First (SSTF)

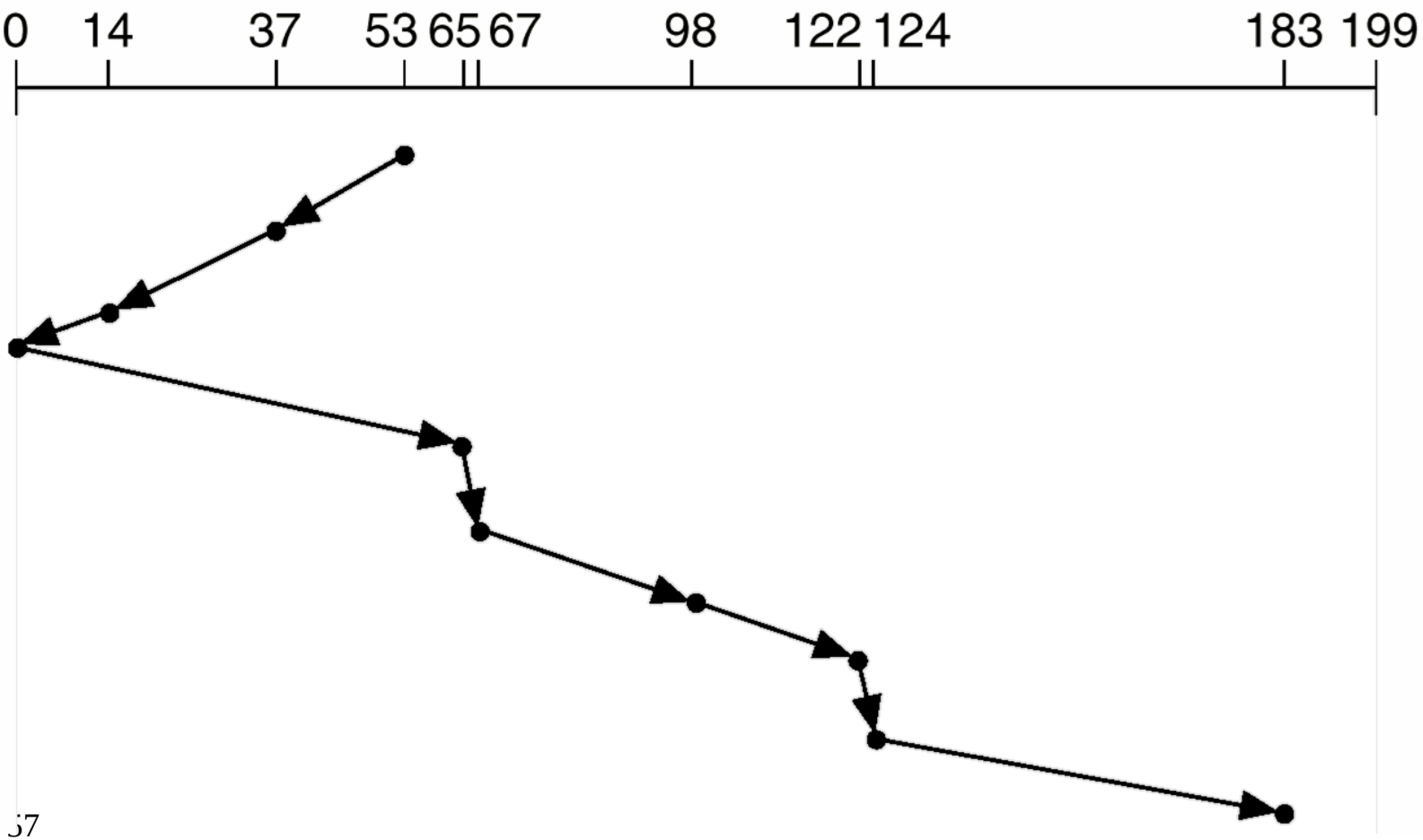
- Request causing least disk arm movement selected
  - a priority based policy
- Advantages
  - Performs better than FIFO
  - offers "locality" -- benefits disk operations close together
- Disadvantages
  - can cause starvation during high request rate
  - No guarantee that average seek time will be minimum over time

Shortest Seek Time First (SSTF)

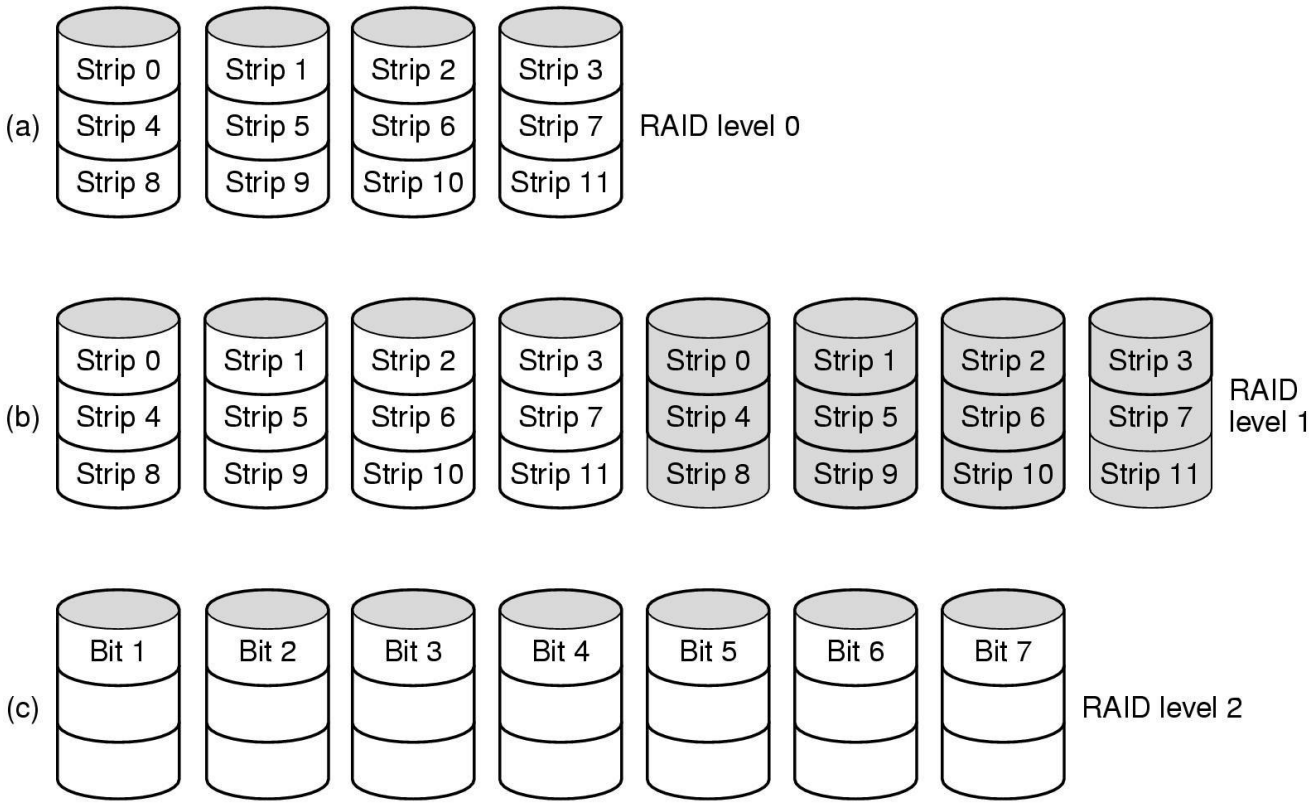


SCAN (elevator algorithm)

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53 and is moving toward cylinder 0

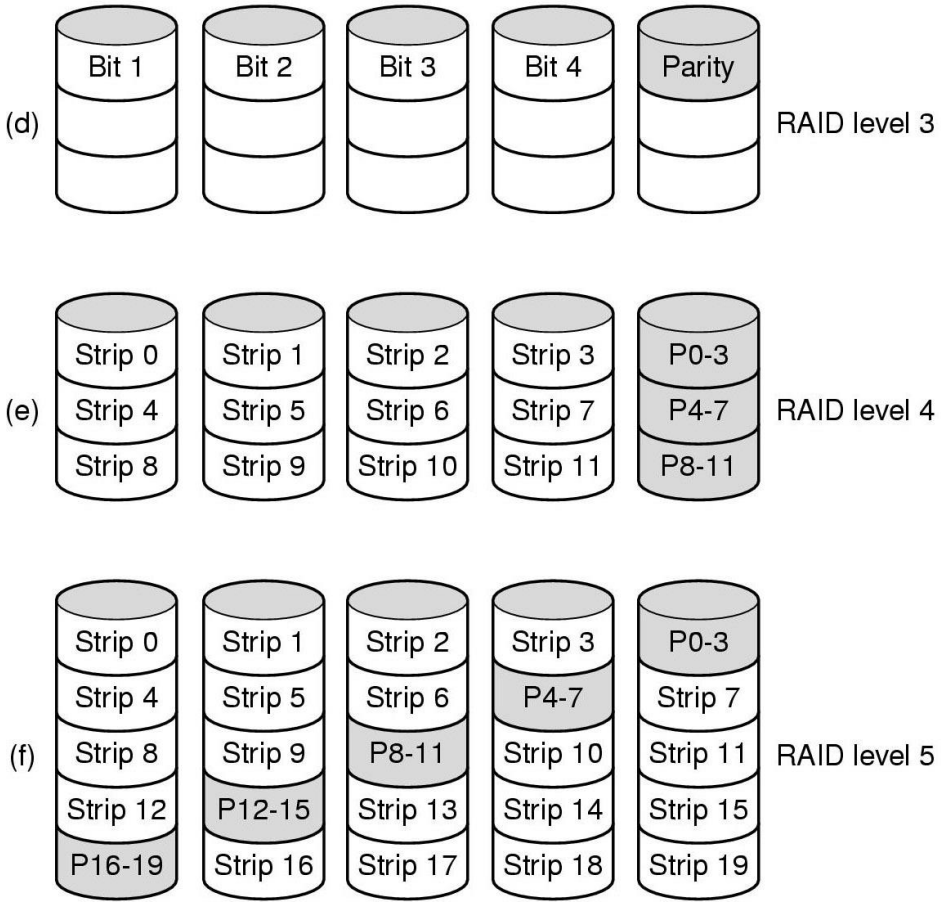


RAID (1)



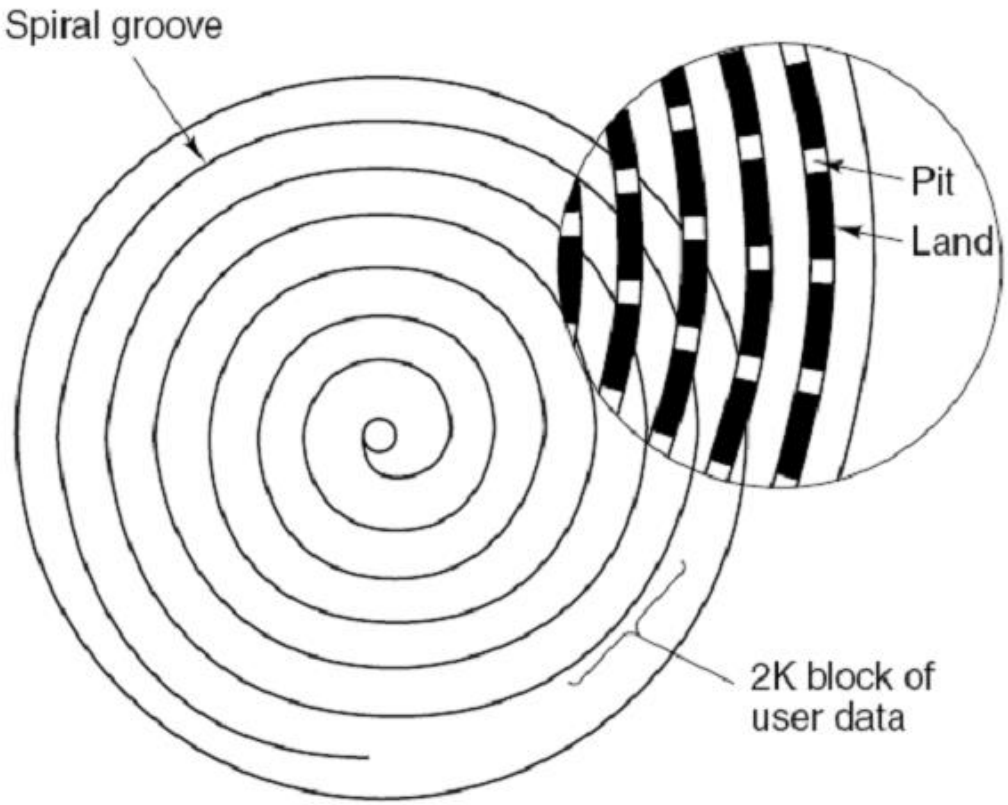
- Raid levels 0 through 2
- Backup and parity drives are shaded

RAID (2)



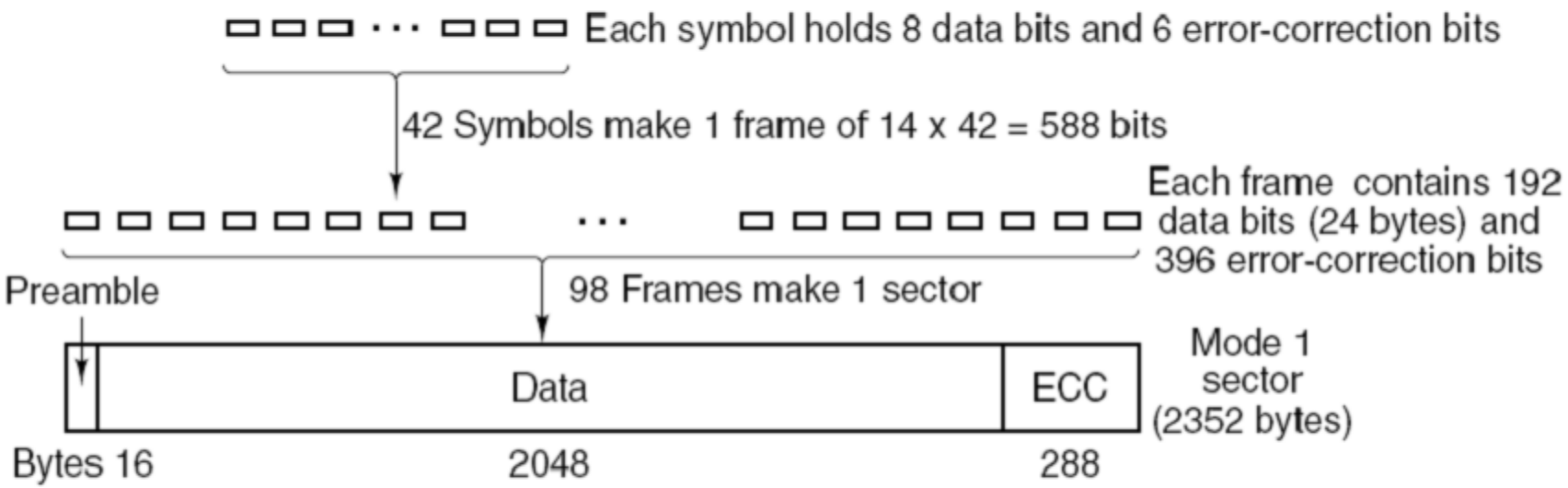
- Raid levels 3 through 5
- Backup and parity drives are shaded

CD-ROMs (1)



Recording structure of a compact disc or CD-ROM.

CD-ROMs (2)



Logical data layout on a CD-ROM.

DVD (1)

DVD Improvements on CDs

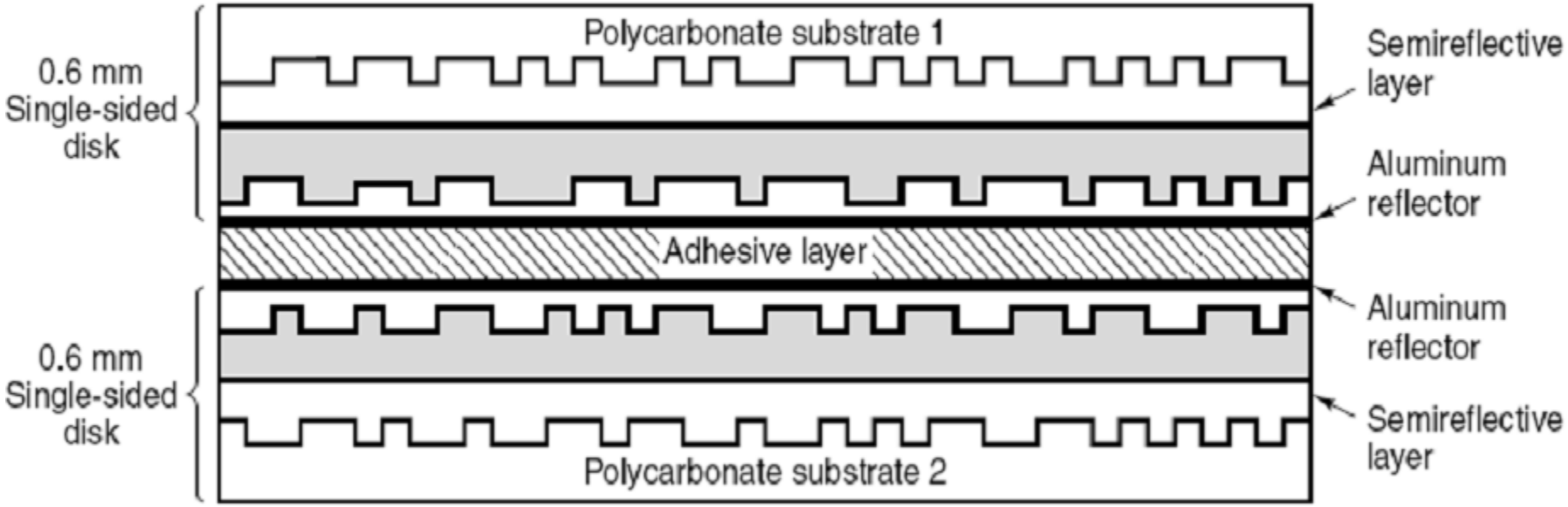
1. Smaller pits
   (0.4 microns versus 0.8 microns for CDs).
2. A tighter spiral
   (0.74 microns between tracks versus 1.6 microns for CDs).
3. A red laser
   (at 0.65 microns versus 0.78 microns for CDs).

DVD (2)

DVD Formats

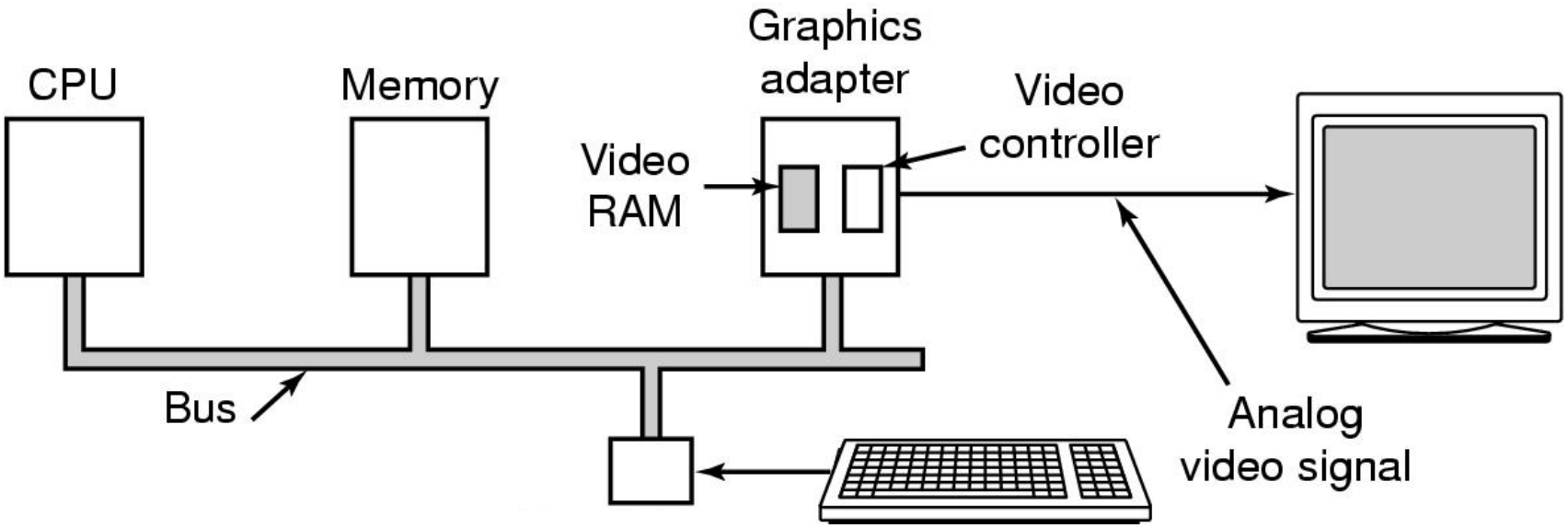
1. Single-sided, single-layer (4.7 GB).
2. Single-sided, dual-layer (8.5 GB).
3. Double-sided, single-layer (9.4 GB).
4. Double-sided, dual-layer (17 GB).

DVD (3)



A double-sided, dual-layer DVD disk.

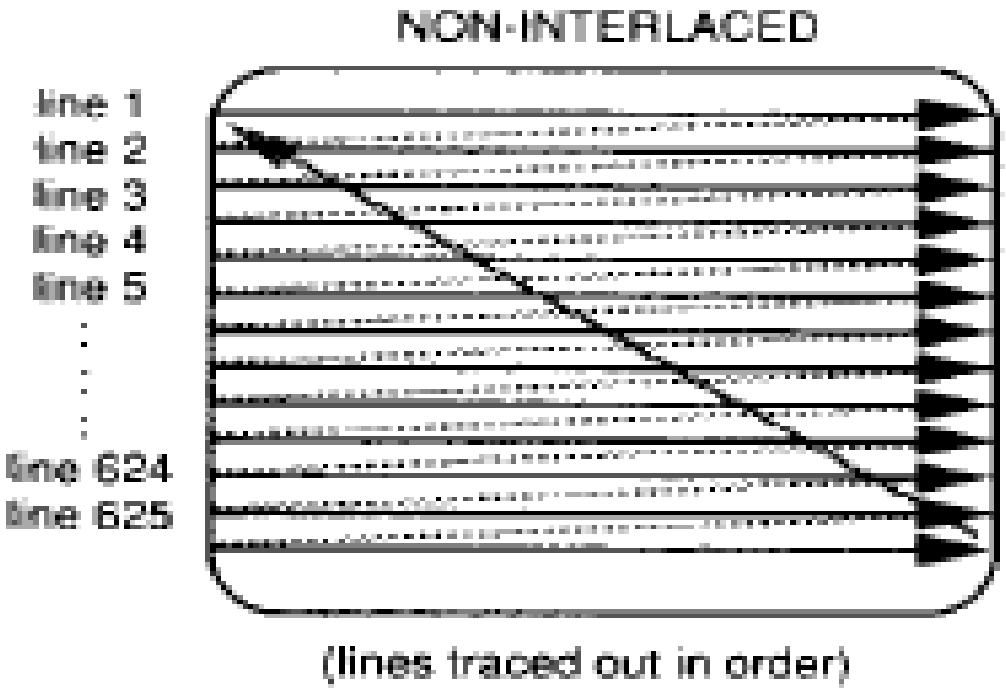
Display Hardware (1)



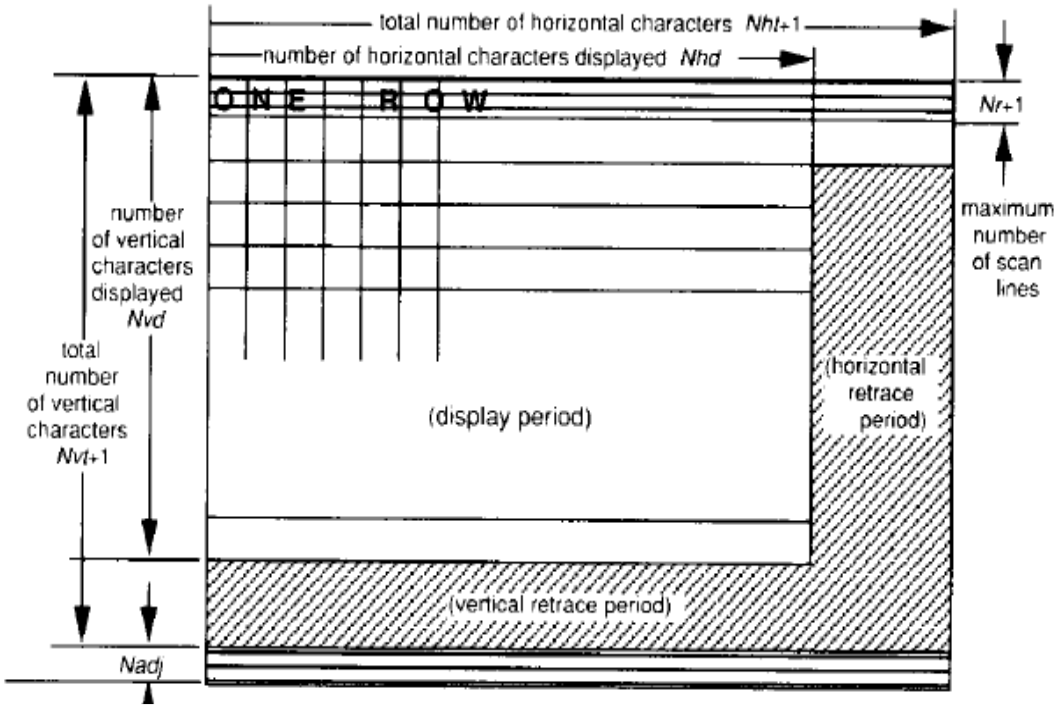
Memory-mapped displays

- driver writes directly into display's video RAM

Display Hardware (2)

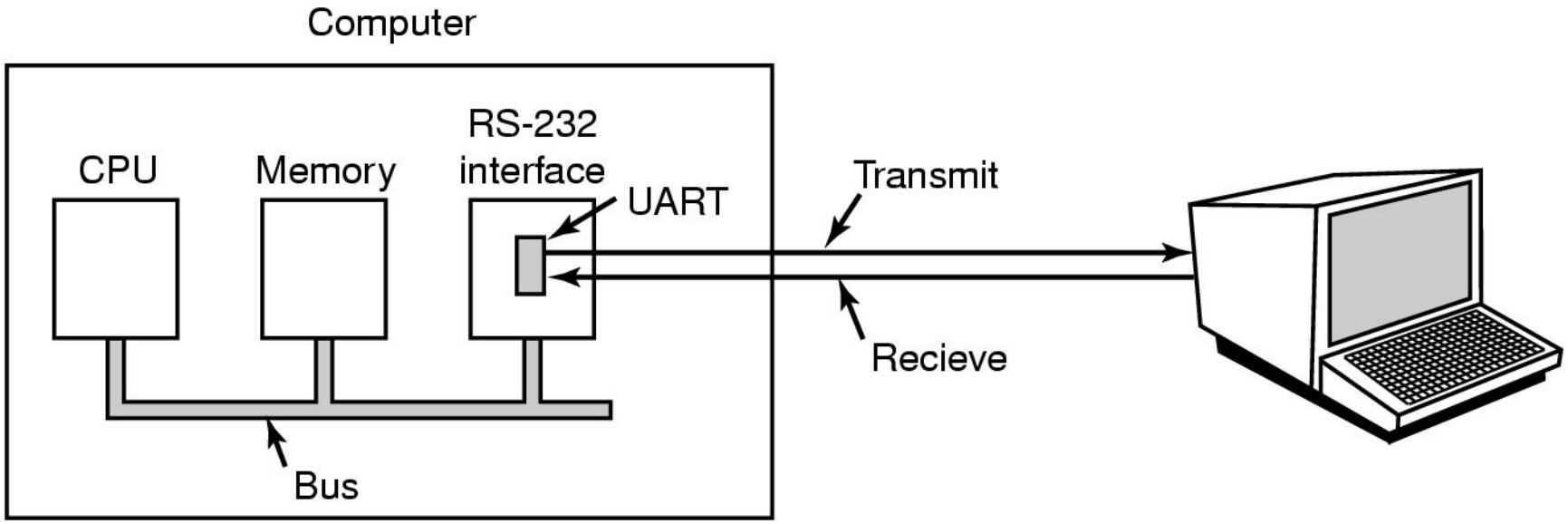


Display Hardware (3)



Character Oriented Terminals

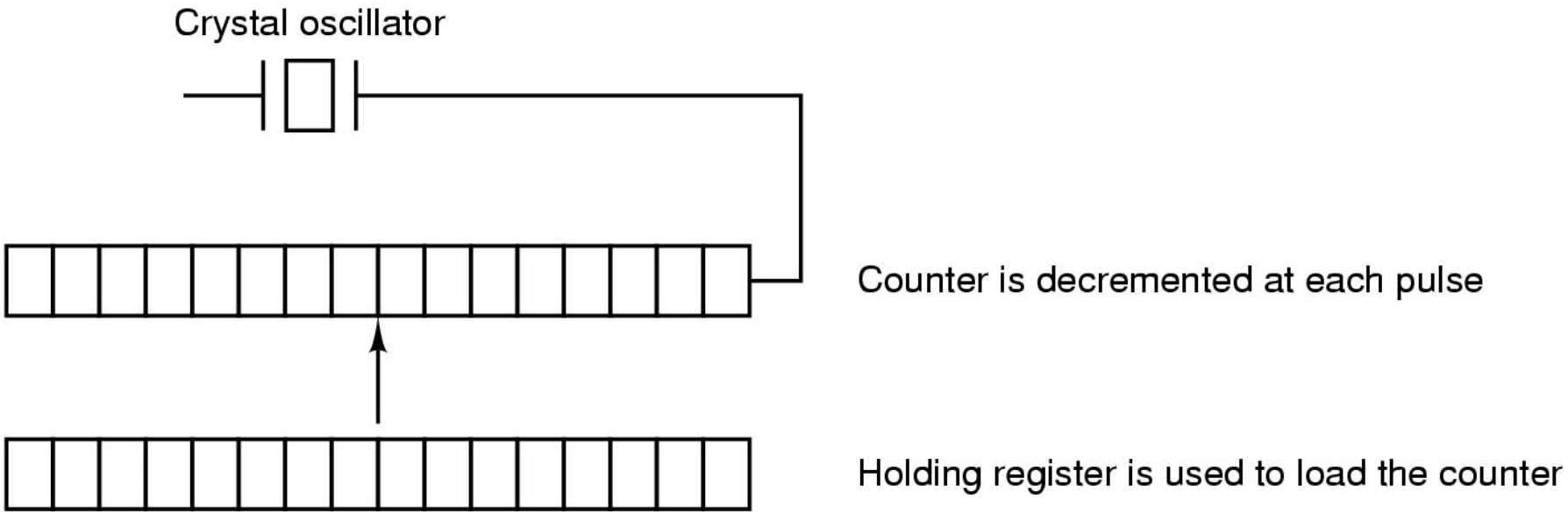
RS-232 Terminal Hardware



- An RS-232 terminal communicates with computer 1 bit at a time
- Called a serial line – bits go out in series, 1 bit at a time
- Windows uses COM1 and COM2 ports, first to serial lines
- Computer and terminal are completely independent

Clocks

Clock Hardware



A programmable clock

Clock Software (1)

Typical duties of a clock driver

1. Maintaining the time of day.
2. Preventing processes from running longer than they are allowed to.
3. Accounting for CPU usage.
4. Handling alarm system call made by user processes.
5. Providing watchdog timers for parts of the system itself.
6. Doing profiling, monitoring, statistics gathering.

- Cấu trúc đĩa CD, DVD: chia vùng ntnao? Lưu trữ dữ liệu theo kiểu gì?
- DVD khác CD ở điểm nào mà dung lượng nhiều hơn?

- HDD, SSD.

- Tìm 3 nhà sản xuất đĩa cứng HDD. Thị phần và khuyến cáo lợi thế, bất lợi của sản phẩm từng công ty.

Western Digital (43%), Segate (41%), Toshiba (16%)…

SATA, ATA, SCSI, EIDE…

16MB, 8MB…

FAT32, NTFS, ext2-3, …

3.5inch, 1.8, 2.5, …

- Xem các thông số của 1 ổ đĩa cứng, giải thích ý nghĩa của từng thông số: tốc độ quay, giao tiếp, bộ nhớ đệm, dung lượng, kích thước…
- Trình bày cấu trúc vật lý của ổ đĩa cứng: track, sector, cylinder, trục quay, đầu đọc/ghi…
- Các định dạng phân vùng dùng được cho HDD

- List 3 HDD manufactures, their market shares, the advantages and disadvantages of each one?
- List all factors of HDD, explain the meaning of each one, such as: rotation speed, interface, buffer, capacity, size,…
- Xem các thông số của 1 ổ đĩa cứng, giải thích ý nghĩa của từng thông số: tốc độ quay, giao tiếp, bộ nhớ đệm, dung lượng, kích thước…
- Trình bày cấu trúc vật lý của ổ đĩa cứng: track, sector, cylinder, trục quay, đầu đọc/ghi…
- Các định dạng phân vùng dùng được cho HDD