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 Disks
5.5 Clocks
5.6 Character-oriented terminals
5.7 Graphical user interfaces
5.8 Network terminals
5.9 Power management

Principles of I/O Hardware

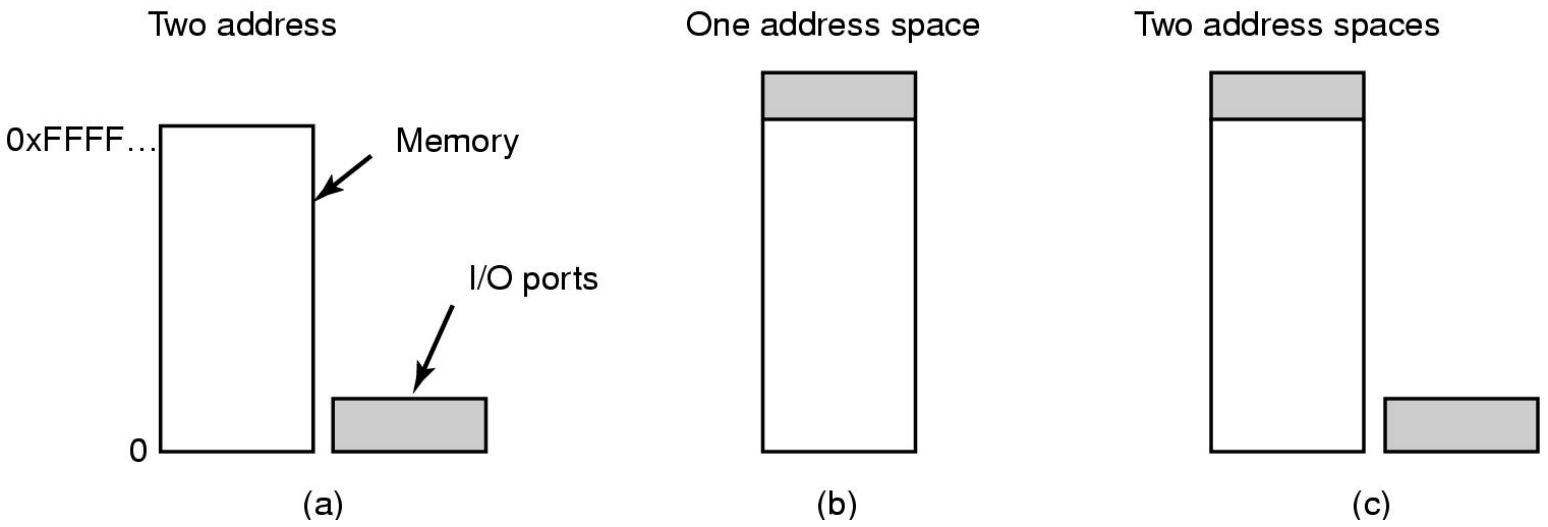
Device	Data rate
Keyboard	10 bytes/sec
Mouse	100 bytes/sec
56K modem	7 KB/sec
Telephone channel	8 KB/sec
Dual ISDN lines	16 KB/sec
Laser printer	100 KB/sec
Scanner	400 KB/sec
Classic Ethernet	1.25 MB/sec
USB (Universal Serial Bus)	1.5 MB/sec
Digital camcorder	4 MB/sec
IDE disk	5 MB/sec
40x CD-ROM	6 MB/sec
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

Some typical device, network, and data base rates

Device Controllers

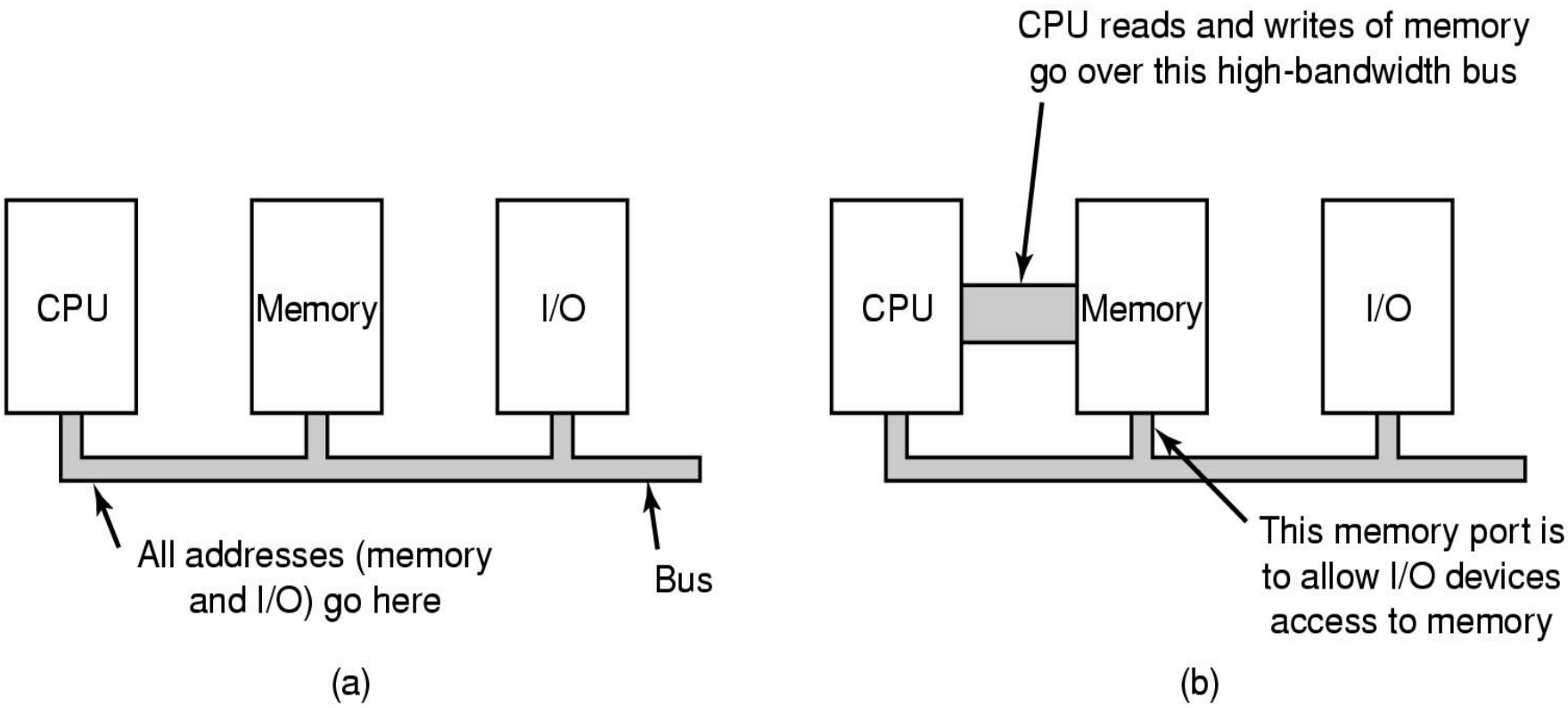
- I/O devices have components:
  - mechanical component
  - electronic component
- The electronic component is the device controller
  - may be able to handle multiple devices
- Controller's tasks
  - convert serial bit stream to block of bytes
  - perform error correction as necessary
  - make available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

Memory-Mapped I/O (1)



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

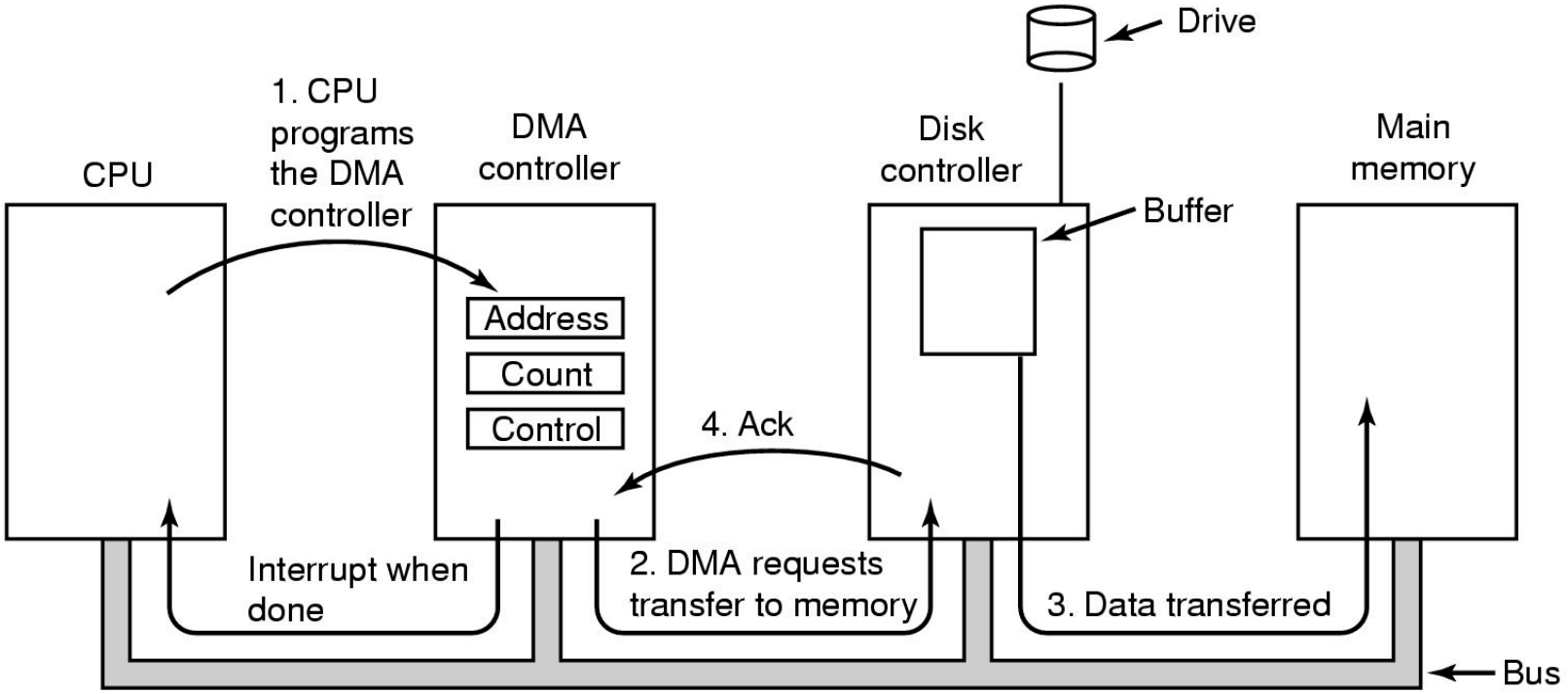
Memory-Mapped I/O (2)



(a) A single-bus architecture

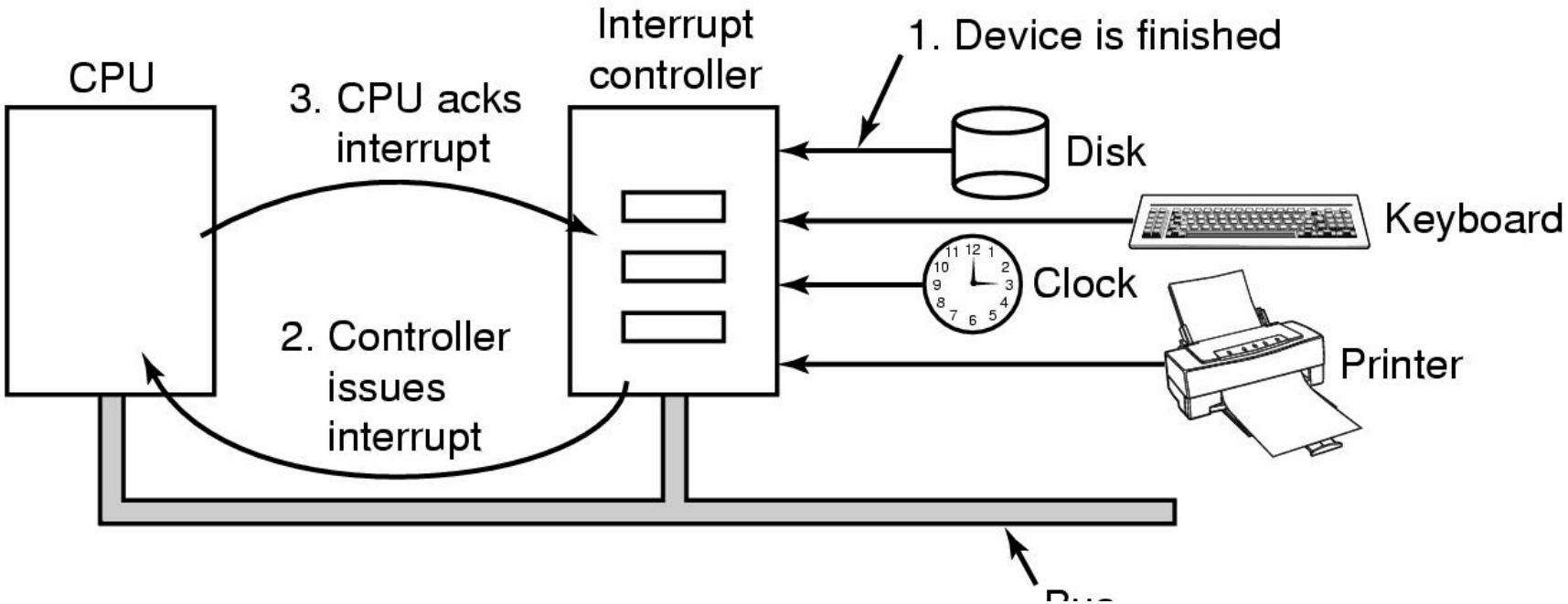
(b) A dual-bus memory architecture

Direct Memory Access (DMA)



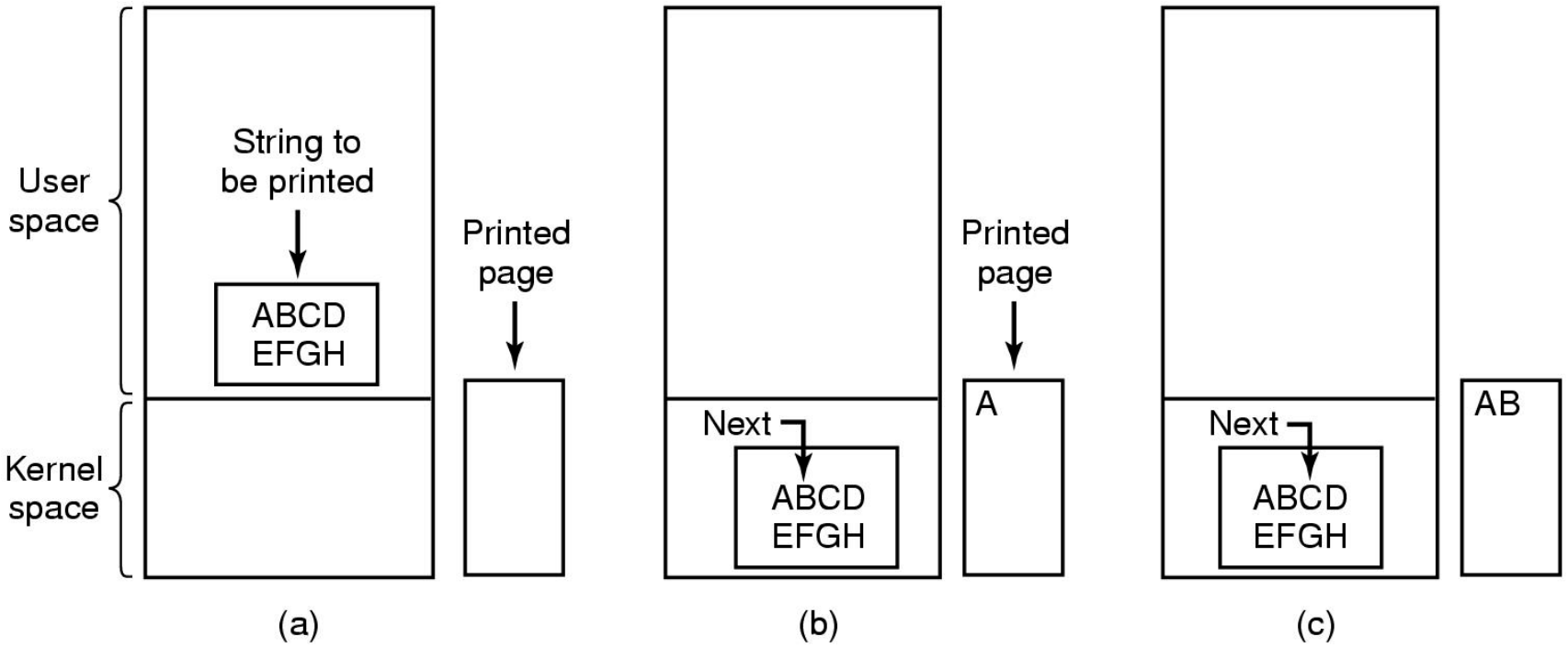
Operation of a DMA transfer

Interrupts Revisited



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

Programmed I/O (1)



Steps in printing a string

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

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

I/O Using DMA

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

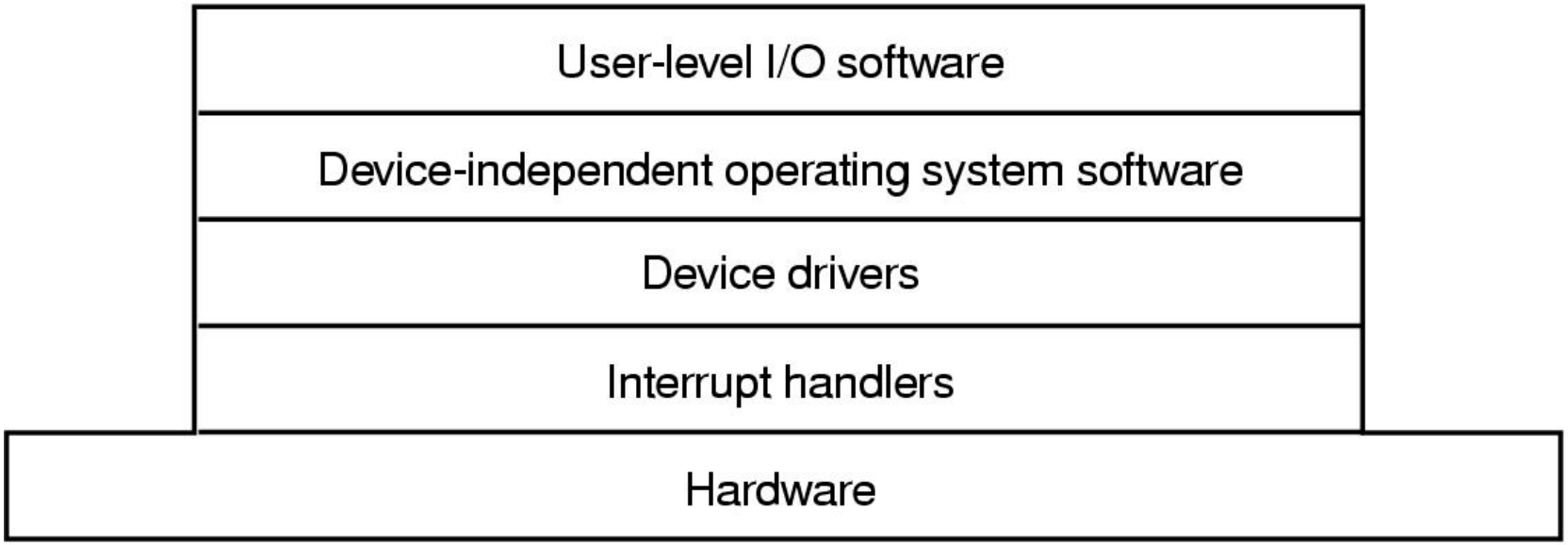
(a)

```
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

I/O Software Layers



Layers of the I/O Software System

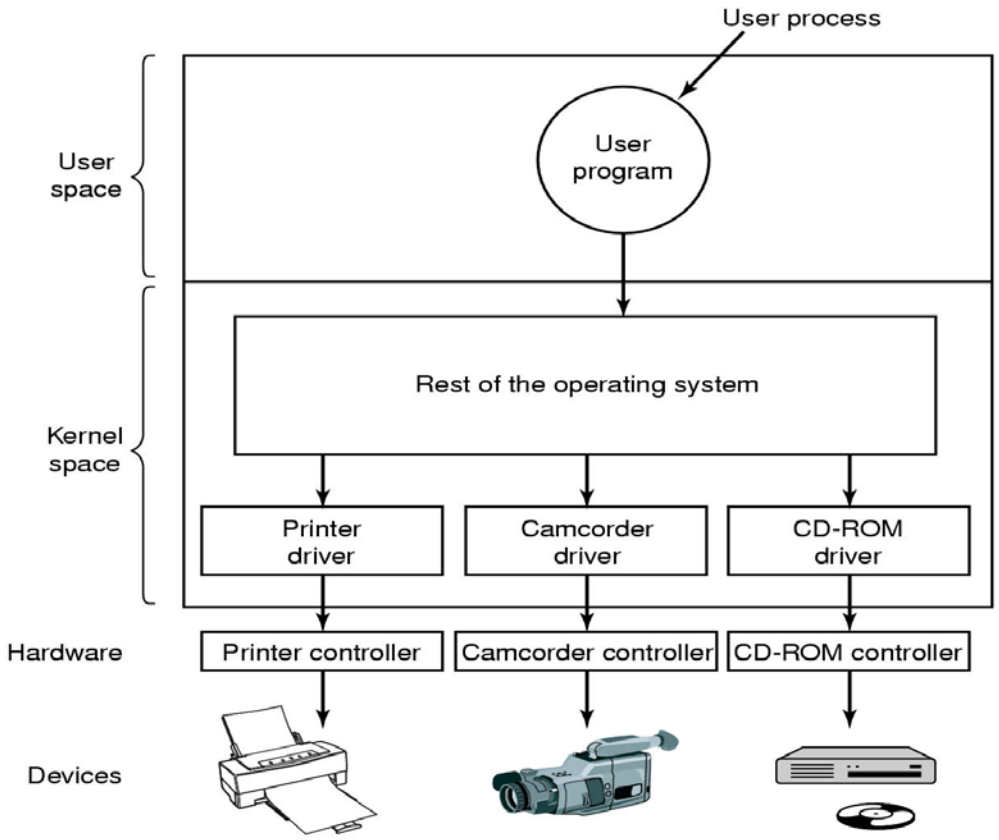
Interrupt Handlers (1)

- 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

Interrupt Handlers (2)

3. Set up stack for interrupt service procedure
4. Ack interrupt controller, reenable interrupts
5. Copy registers from where saved
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

Device Drivers



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

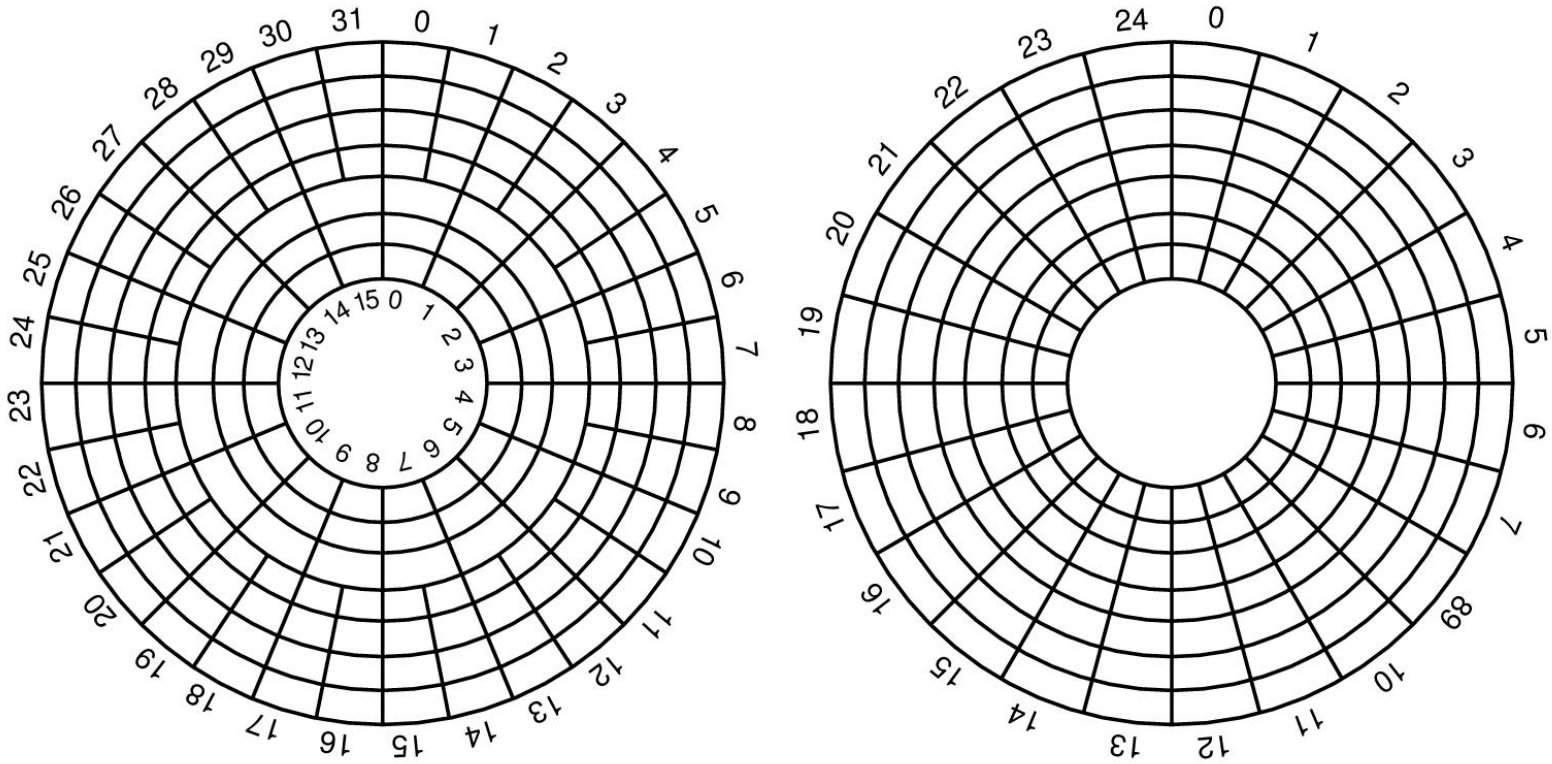
Disks

Disk Hardware (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

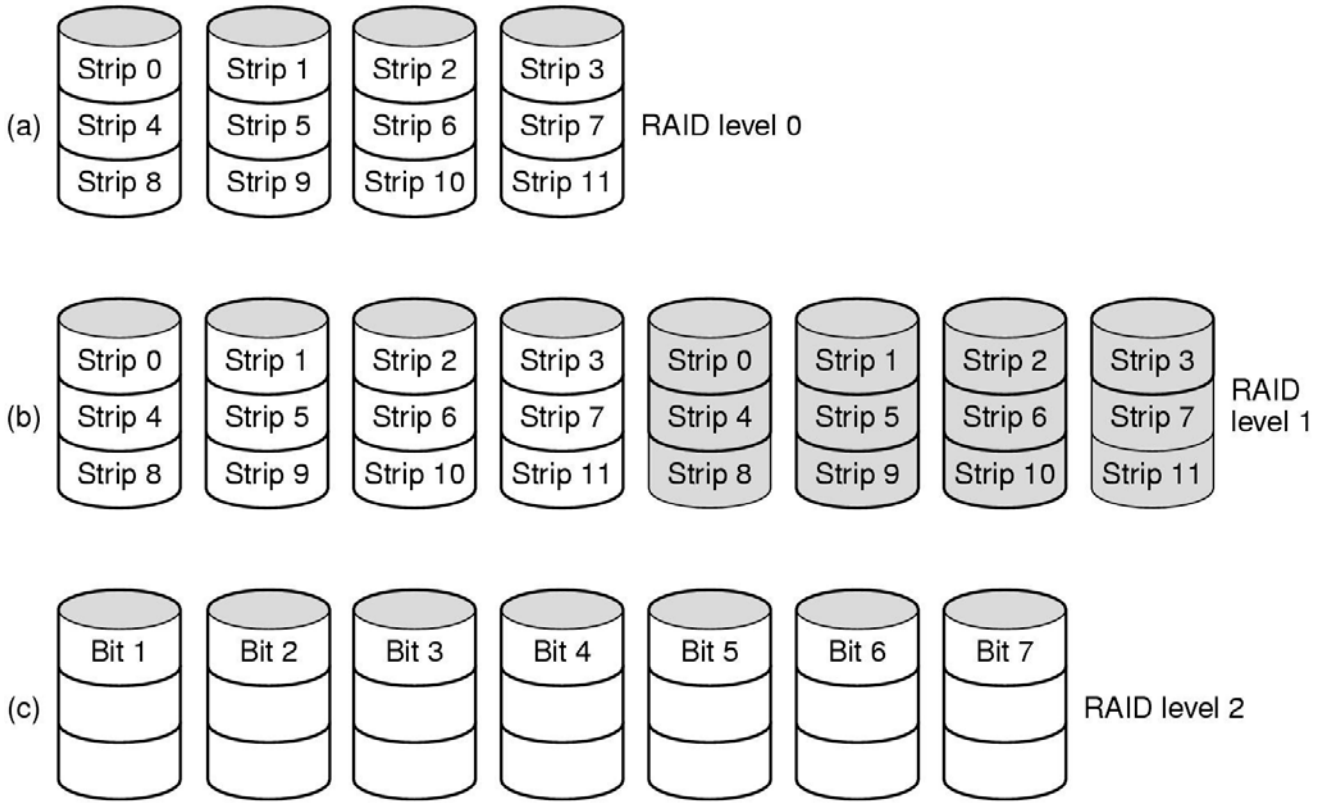
Disk parameters for the original IBM PC floppy disk and a Western Digital WD 18300 hard disk

Disk Hardware (2)



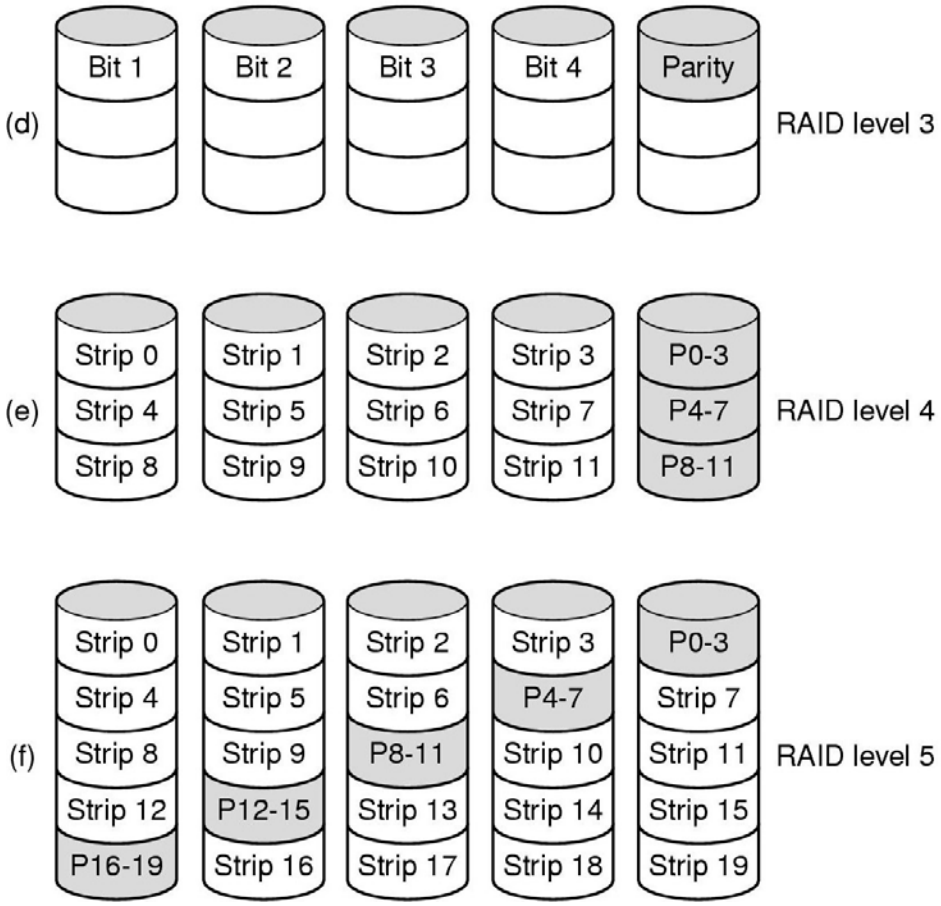
- Physical geometry of a disk with two zones
- A possible virtual geometry for this disk

Disk Hardware (3)



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

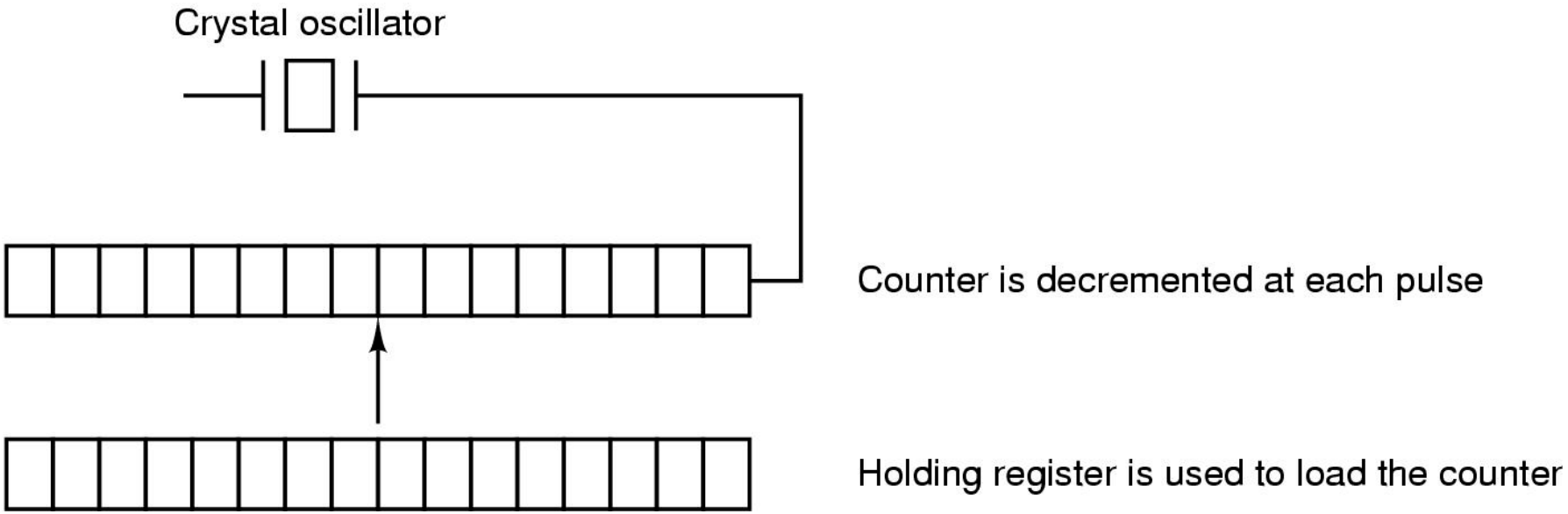
Disk Hardware (4)



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

Clocks

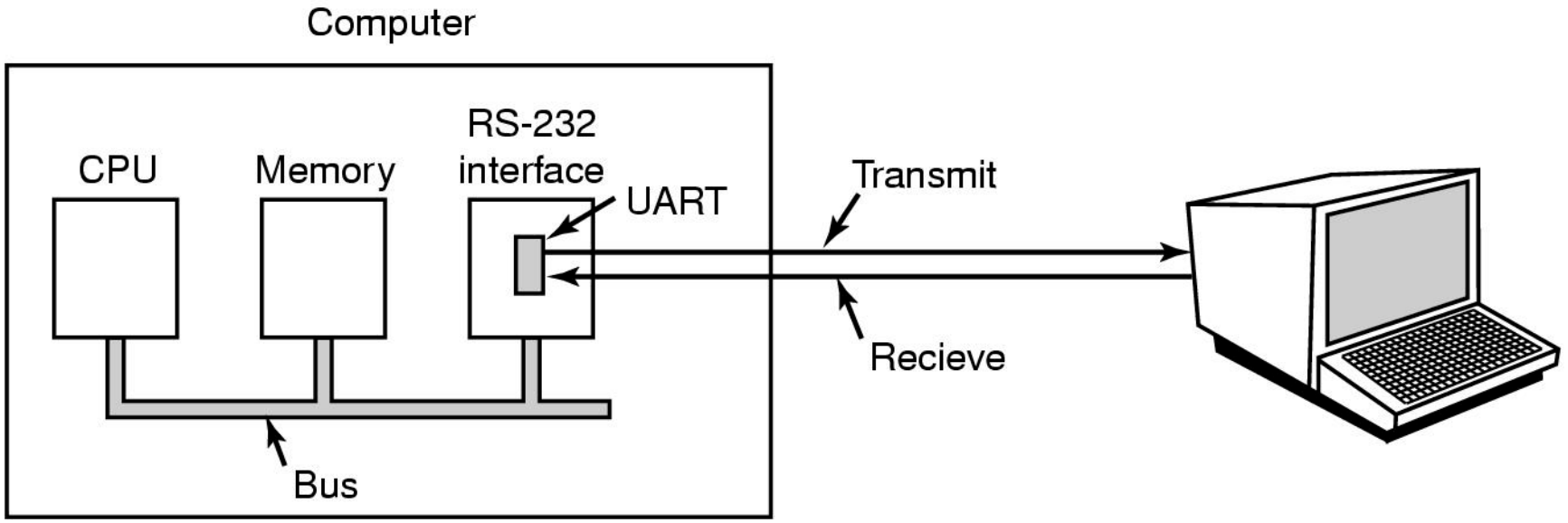
Clock Hardware



A programmable clock

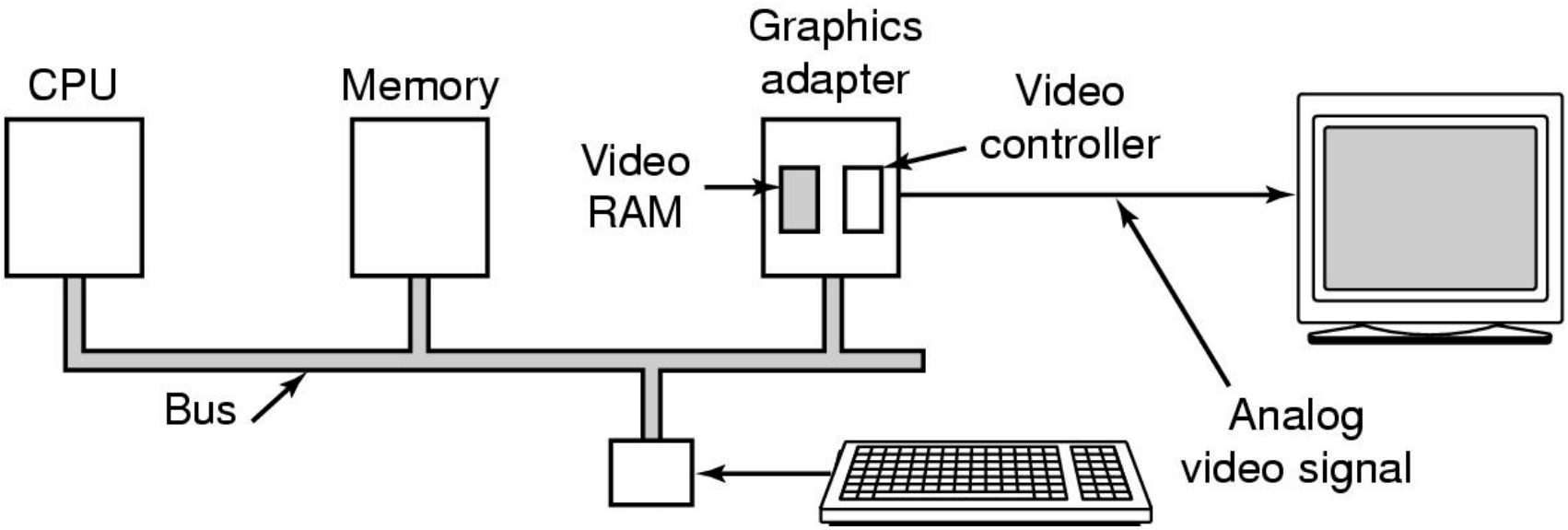
Character Oriented Terminals

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

Display Hardware (1)



Memory-mapped displays

- driver writes directly into display's video RAM