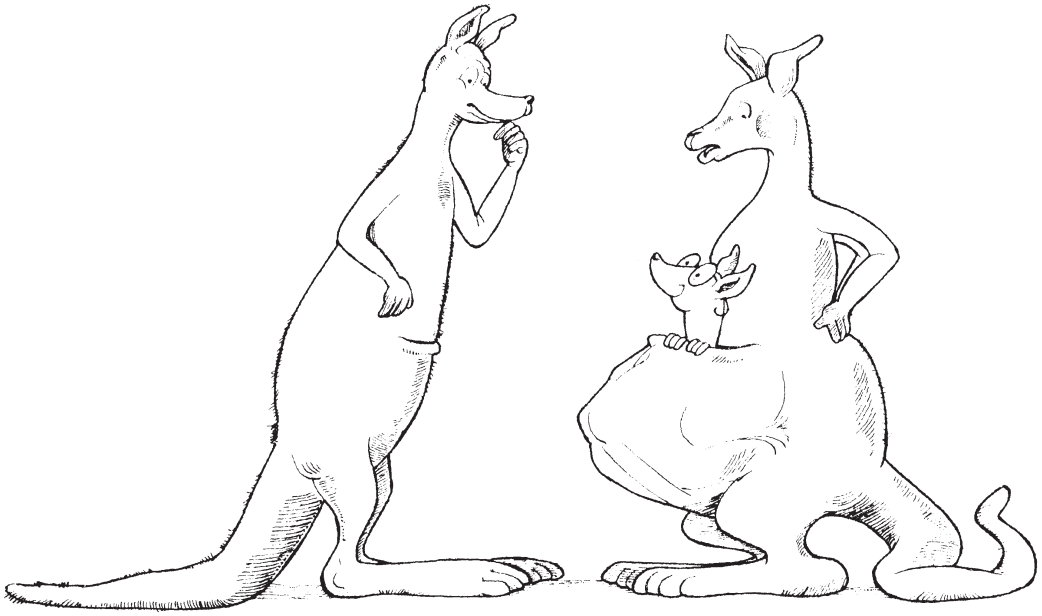
CHAPTER 10

COMPUTER PERIPHERALS



Thomas Sperling. Adapted, courtesy of David Ahl, Creative Computing.

10.0 INTRODUCTION

The typical personal computer system described in an advertisement consists of a CPU, memory, a DVD or CD read-write drive, a hard disk drive, a keyboard, a mouse, wireless and wired network interfaces, USB ports, sound and video system components, usually a modem, perhaps parallel, FireWire, and serial ports, and a monitor. Additional available components include scanners of various types, printers, plotters, TV tuners, floppy disk drives, and tape drives. Internal to the computer there is also a power supply that converts wall plug power into voltages suitable for powering a computer. All the items mentioned, except for the CPU, memory, and power supply, are considered peripheral (that is, external) to the main processing function of the computer itself and are known, therefore, as **peripherals**. Some of the peripherals use the USB, parallel, and serial ports as their interconnection point to the computer. Others have their own interface to internal system buses that interconnect various parts of the computer.

The peripherals in a large server or mainframe computer are similar, except larger, with more capacity. Large numbers of hard disk drives may be grouped into arrays to provide capacities of tens or hundreds of terabytes (TB). One or more high-speed network interfaces will be a major component. The capability for handling large amounts of I/O will likely be a requirement. Means of implementing large-scale, reliable backup will be necessary. Conversely, fancy displays, high-end graphics and audio cards, and other multimedia facilities may be totally unneeded.

Despite different packaging and differences in details, the basic operations of these devices are similar, regardless of the type of computer. In previous chapters we have already looked at the I/O operations that control devices that are external to the CPU. Now we direct our attention to the operation of the devices themselves. In this chapter we study the most important computer peripheral devices. We look at the usage, important characteristics, basic physical layouts, and internal operations of each device. We will also briefly consider the interface characteristics for these devices.

Peripheral devices are classified as input devices, output devices, or storage devices. As you would expect, input data is data *from* the outside world *into* the CPU, and output data is data moving *from* the CPU *out to* the outside world. Storage devices are, of course, both input and output devices, though not at the same time. If you recall the concept of input process-output from Chapter 1, programs require input, process it, and then produce output. Using a storage device, data output is stored, to be used as input at a future time. In a transaction processing system, for example, the database files are stored on line. When a transaction occurs, the transaction processing program will use input from the new transaction together with data from the database to update the appropriate database records as output. The updated database remains in storage for the next transaction.

Because of the importance of storage, we will begin with a discussion of storage devices. Following that, we will consider various input and output devices.

It should be noted that the technologies used for many peripheral components are very sophisticated; some would even say that these devices operate by magic! You may agree when you see the descriptions of some components. It is not uncommon to have more sophisticated control and technology in a peripheral component than in the computer itself. Perhaps you have wondered how these devices work. Here's your opportunity to find out!

We have not attempted to provide a detailed explanation of every possible peripheral device in a computer system. Instead, we have selected several interesting devices that are representative of a number of technologies.

At the completion of this chapter, you will have been exposed to every important hardware component of the computer system, with the exception of the pieces that tie the components of the computer systems together, and furthermore extend the systems themselves together into networks. You will have seen the role and the inner workings of each component that we have discussed, and you will have seen how the different components fit together to form a complete computer system. You will have a better understanding of how to select particular components to satisfy specific system requirements and of how to determine device capacities and capabilities.

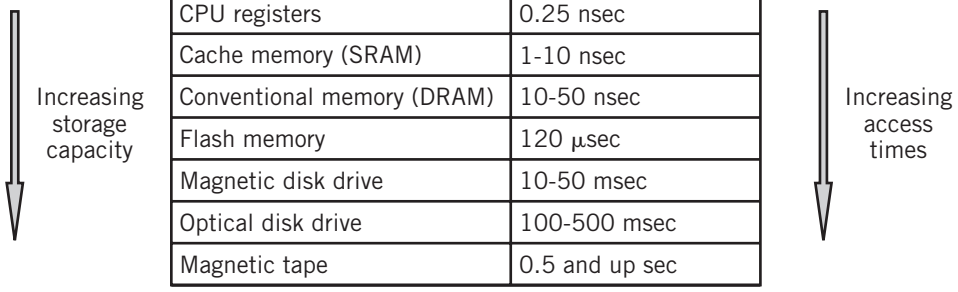
10.1 THE HIERARCHY OF STORAGE

Computer storage is often conceptualized hierarchically, based upon the speed with which data can be accessed. The table in Figure 10.1 shows this hierarchy, together with some typical access times.

At the top of the hierarchy are the CPU registers used to hold data for the short term while processing is taking place. Access to registers is essentially instantaneous, since the registers are actually a part of the CPU. Cache memory, if present, is the fastest memory outside the CPU. You recall from Chapter 8 that cache memory is a small fast memory that is used to hold current data and instructions. The CPU will always attempt to access current instructions and data in cache memory before it looks at conventional memory. There may be as many as three different levels of cache. The CPU accesses the data or instruction in conventional memory if cache memory is not present. Next in the hierarchy is conventional memory. Both conventional and cache memory are referred to as **primary**

FIGURE 10.1

The Storage Hierarchy



Device	*Typical access times*
CPU registers	0.25 nsec
Cache memory (SRAM)	1-10 nsec
Conventional memory (DRAM)	10-50 nsec
Flash memory	120 μsec
Magnetic disk drive	10-50 msec
Optical disk drive	100-500 msec
Magnetic tape	0.5 and up sec

memory. Both provide immediate access to program instructions and data by the CPU and can be used for the execution of programs. The data throughput rate of memory is determined primarily by the capability of the bus and interfaces that connect memory to the CPU. Rates well in excess of 1 GB/sec are common in modern computers.

Below the level of conventional memory, storage in the hierarchy is not immediately available to the CPU, is referred to as **secondary storage**, and is treated as I/O. Data and programs in secondary storage must be copied to primary memory for CPU access.[1] Except for flash memory, access to secondary storage is significantly slower than primary storage. Disks and other secondary storage devices are mechanical in nature, and mechanical devices are of necessity slower than devices that are purely electronic. The location of the desired data is usually not immediately accessible, and the medium must be physically moved to provide access to the correct location. This requires a *seek time*, the time needed to find the desired location. Once the correct data is located, it must be moved into primary memory for use. The throughput rate in Figure 10.1 indicates the speed with which the transfer of data between memory and the I/O device can take place. Most of the access time specified for secondary storage devices consists of seek time. As a result of this access time, even the fastest disks are only about one-millionth as fast as the slowest memory. It should be apparent that a *lot* of CPU instructions can be performed while waiting for a disk transfer to take place.

One important advantage of secondary storage, of course, is its permanence, or lack of volatility. As noted in Chapter 7, RAM data is lost when the power is shut off. Flash memory uses a special type of transistor that can hold data indefinitely without power. The magnetic media used for disk and tape and the optical media used for DVD and CD disks also retain data indefinitely. Secondary storage has the additional advantage that it may be used to store massive amounts of data. Even though RAM is relatively inexpensive, disk and tape storage is much cheaper yet. Large amounts of **online secondary storage** may be provided at low cost. Current hard disks store data at a density of nearly 40 Gbits per square centimeter!

Tape, most flash memory devices, optical disks, and many magnetic disks are designed for easy removal from the computer system, which makes them well suited for backup and for **off-line storage** of data that can be loaded when the data is needed. This provides the additional advantage that secondary storage may be used for offline archiving, for moving data easily from machine to machine, and for offline backup storage. For example, a flash memory card may be used to store digital camera photographs until they are moved to a computer for long term storage; similarly, a removable hard disk can be used to move large amounts of data between computers.

As an increasingly common alternative, data and programs may be stored on a secondary storage device connected to a different computer and accessed through a network connection between the computers. In this context, the computer with secondary storage is sometimes known as a **server** or a **file server**. In fact, the primary purpose of the server may be to act as a storage provider for all the computers on the network. Web services are a common application of this type. Optical disks require little space and can

[1] In the earliest days of computing, secondary storage devices, particularly rotating drums (forerunner of the disk), were actually used as memory with direct access to the CPU. To run efficiently, programs had to be designed to minimize the number of rotations of the drum, which meant that the programmer would always attempt to have the next required location be just ahead of where the drum head was at that instant. Those were interesting days for programmers!

store large amounts of data for archiving and installation purposes, with rapid mounting for retrieval when required. A few high-capacity optical disks could store all the medical records and history for a large insurance company, for example. Most modern programs are supplied on DVD or CD-ROM.

Of the various secondary storage components, flash memory and disk devices are the fastest, since data can be accessed randomly. In fact, IBM refers to disks as **direct access storage devices (DASDs)**. With tape, it may be necessary to search sequentially through a portion of the tape to find the desired data. Also, the disk rotates continuously, while the tape will have to start and stop, and possibly even reverse direction and rewind to find the desired data. These factors mean that tape is inherently slower unless the data is to be read sequentially. This makes tape suitable only for large-scale offsite backup storage where the entire contents of a disk are transferred to tape to protect the data from a potential catastrophe or to meet legal long term data retention requirements. Although magnetic tape storage had large inherent cost and storage capacity advantages in the past, that is no longer the case, and the use of tape is decreasing as businesses replace their equipment with newer technology.

10.2 SOLID STATE MEMORY

Flash memory is nonvolatile electronic integrated circuit memory, similar conceptually to the read-only memory discussed in Chapter 7, but different in technology. The difference makes flash memory suitable for use in situations where traditional ROM would be impractical. Whereas traditional ROM must be read, erased, and written in large blocks of addresses, it is possible to read individual bytes or small blocks of flash memory when necessary. This makes flash memory useful for applications that require random access, particularly those applications where most accesses are reads.

Although read accesses and certain simple overwrite accesses are relatively fast, flash memory must be erased in blocks, so that most write accesses require an additional step that rewrites the unchanged data back to the block. Furthermore, the erase-and-rewrite operation is very slow compared to the read access. Although there is research into other types of nonvolatile memory that might solve this problem, flash memory is generally considered to be impractical as a replacement for conventional RAM, at least for now.

Because of its small size, flash memory is frequently the secondary storage of choice for the memory cards that plug into portable devices such as cell phones, portable music players, and digital cameras. It is also well suited for small, portable ''thumb drives'' that plug directly into a USB port. These drives are useful for moving files and data from one machine to another and also serve as an inexpensive and convenient backup medium.

Flash memory is more expensive than disk storage at this writing. However, its capacity is rapidly increasing and its price falling. As a result, large capacity flash memory units called ''solid-state drives'' have appeared on the market and are starting to supplant disk drives as the long-term storage device of choice in computers where less weight, low power consumption, and small size are important. ''Solid-state drives'' have the additional advantages of being relatively immune to failure due to physical shock and vibration (since they have no moving parts), and generate little heat and no noise. Solid-state drives have not yet reached the huge storage capacities of large disk drives, but their capacity is continually expanding, and is already adequate for many applications.

10.3 MAGNETIC DISKS

A magnetic disk consists of one or more flat, circular platters made of glass, metal, or plastic, and coated with a magnetic substance. Particles within a small area of the magnetic substance can be polarized magnetically in one of two directions with an electromagnet; an electromagnet can also detect the direction of polarization previously recorded. Thus, magnetic polarization can be used to distinguish 1s and 0s. Electromagnetic read/write heads are used for this purpose.

A drive motor rotates the disk platter(s) about its central axis. On most drives, the motor rotates the disk at a fixed speed. An arm has the read/write head mounted at the end. The arm makes it possible for the head to move radially in and out across the surface of the disk. A head motor controls precisely the position of the arm on the disk.

Most **hard disk drives** contain several platters, all mounted on the same axis, with heads on each surface of each platter. The heads move in tandem, so they are positioned over the same point on each surface. Except for the top and bottom, each arm contains two read/write heads, which service the surfaces of two adjoining platters.

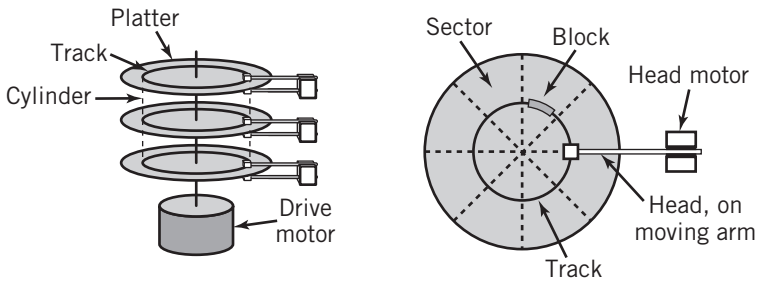
With the head in a particular position, it traces out a circle on the disk surface as the disk rotates; this circle is known as a **track**. Since the heads on each surface all line up, the set of tracks for all the surfaces form a **cylinder**. Each track contains one or more blocks of data. On most disks the surface of the disk platter is divided into equally sized pie shape segments, known as **sectors**, although the disks on some large computers divide up the track differently. Each sector on a single track contains one **block** of data, typically 512 bytes, which represents the smallest unit that can be independently read or written. Figure 10.2 shows the layout of a hard disk.

If you assume that the number of bytes in a sector is the same anywhere on the disk, then you can see from the layout that the bits on the disk are more closely packed on the inner tracks than they are on the outer tracks. Regardless of the track, the same angle is swept out when a sector is accessed; thus, the transfer time is kept constant with the motor rotating at a fixed speed. This technique is called **CAV**, for **constant angular velocity**. CAV has the advantage of simplicity and fast access.

It is possible to increase the capacity of the disk by utilizing the space at the outer tracks to pack more bits onto the disk. But this would result in a different number of bytes per sector or a different number of sectors per track depending on which track is being

FIGURE 10.2

A Hard Disk Layout

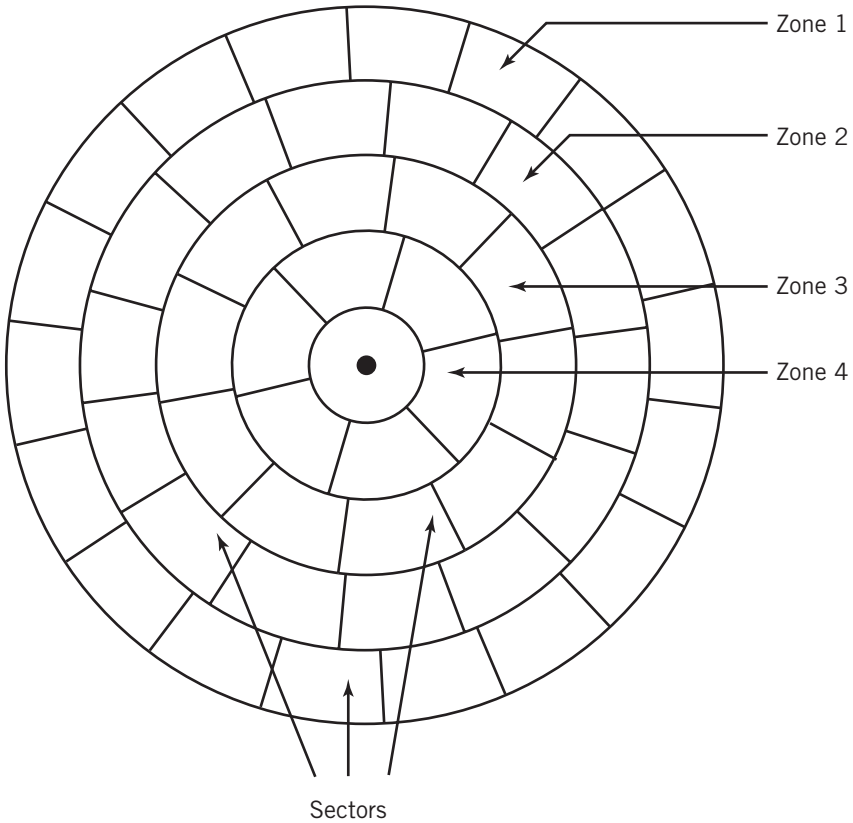


accessed. This would make it more difficult to locate the required sector. Notice, too, that with a constant speed motor, the time to move the head over a pie-shaped sector at the edge is the same as that near the center. If there were more bits packed into the outer tracks, the data would be transferred faster at the edge than at the center. Since the disk controller is designed to expect data at a constant speed, it would seem to be necessary to design the motor so that it would slow down when the head was accessing the outer tracks to keep the data transfer speed constant. In this case, the motor speed would be adjusted such that the speed *along the track* would be constant regardless of the position of the head. This approach is called **CLV**, for **constant linear velocity**. The capacity of a CLV disk with the same diameter and bit density is approximately double that of an equivalent CAV disk. Although CLV technology is commonly used with CDs and DVDs, the design makes it more difficult to access individual blocks of data rapidly, so it is rarely used for hard disks.

As a compromise, modern disk drives divide the disk into a number of zones, typically sixteen. This approach is shown in Figure 10.3. The cylinders in different zones have a different number of sectors but the number of sectors within a particular zone is constant.

FIGURE 10.3

Multiple-Zone Disk Configuration



Obviously, the largest number of sectors will be in the zone containing the outermost cylinders, Instead of adjusting the motor speed, the disk controller buffers the data rate so that the data rate to the I/O interface is constant, despite the variable data rate between the controller and the disk. Different vendors call this technique **multiple zone recording**, **zone bit recording** (**ZBR**), or **zone-CAV recording** (**Z-CAV**).

The platter on a hard disk drive is made of a rigid material and is precisely mounted. The heads on a hard disk do not touch the surface; rather, they ride on a bed of air a few millionths of an inch above the surface. The location of the heads radially is tightly controlled. This precision allows the disk to rotate at high speed and also allows the designers to locate the tracks very close together. The result is a disk that can store large amounts of data and that retrieves data quickly. A typical hard disk rotates at 5400 revolutions per minute (rpm), 7200 rpm, or even 10,800 rpm.

A photograph of a hard disk assembly showing a disk platter, arm, and read/write head is shown in Figure 10.4. This particular hard disk drive contains three platters and six heads. Only the topmost platter and head are fully visible. The entire assembly is sealed to prevent dirt particles from wedging between the heads and the disk platter, since this situation could easily destroy the drive. Even a particle of cigarette smoke is much larger than the space between the head and the disk. When the disk is stationary, the head rests in a **parked** position on the edge of the drive. The head has an aerodynamic design, which causes it to rise on a cushion of air when the disk platter spins.

Figure 10.5 shows the operation required to locate an individual block of data. First, the arm moves the head from its present track until it is over the desired track. The time that is required to move from one track to another is known as the **seek time**. Since the distance between the two tracks is obviously a factor, the **average seek time** is used as a specification for the disk. Once the head is located over the desired track, the read/write

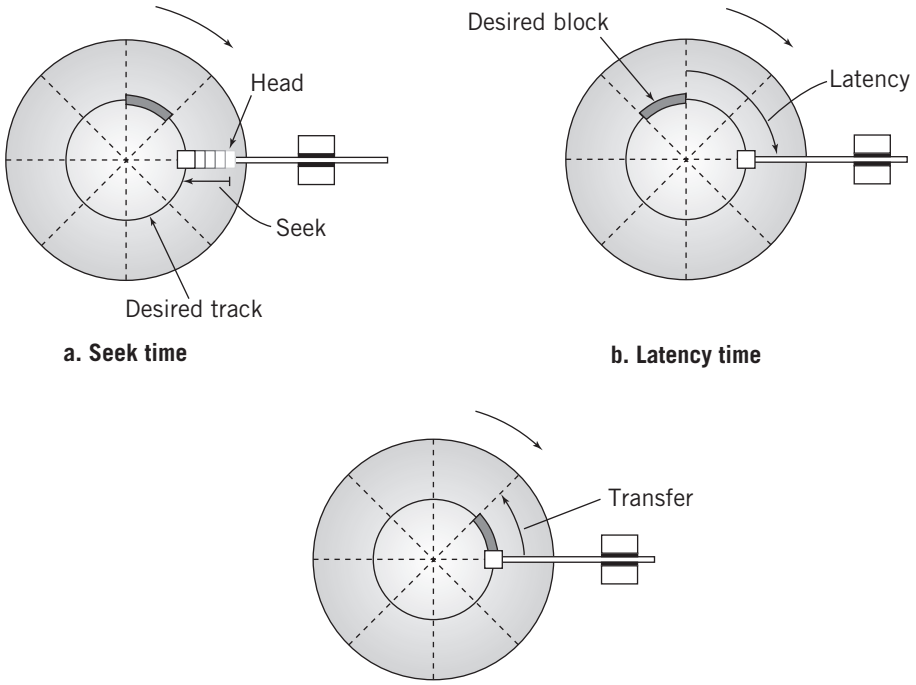
FIGURE 10.4

A Hard Disk Mechanism

Courtesy Western Digital Corporation.

FIGURE 10.5

Locating a Block of Data: (a) Seek Time, (b) Latency Time, (c) Transfer Time



a. Seek time

b. Latency time

c. Transfer time

operation must wait for the disk to rotate to the beginning of the correct sector. The time for this to occur is known as the **rotational latency time**, or sometimes as **rotational delay** or simply **latency time**. The latency time is obviously variable, depending on the position of the disk. As a best case, the head is just about to enter the sector, and the rotational latency time is 0.

At the opposite extreme, the head has just passed the beginning of the sector, and a full rotation is required to reach the beginning of the sector. This time can be calculated from the rotational speed of the disk. Both situations are equally probable. On average, the disk will have to rotate half way to reach the desired block. Thus, the average latency time can be calculated from the rotational speed of the disk as

$$\texttt{average latency} = \frac{1}{2} \times \frac{1}{\texttt{rotational speed}}$$

For a typical hard disk rotating at 3600 revolutions per minute, or 60 revolutions per second, the average latency is

$$\texttt{average latency} = \frac{1}{2} \times \frac{1}{60} = 8.33\ \textit{milliseconds}$$

Once the sector is reached, the transfer of data can begin. Since the disk is rotating at a fixed speed, the time required to transfer the block, known as **transfer time**, is defined by

the number of sectors on a track, since this establishes the percentage of the track that is used by a single data block. The transfer time is defined by

$$\text{transfer time} = \frac{1}{\text{number of sectors} \times \text{rotational speed}}$$

If the hard drive in the example contains 30 sectors per track, the transfer time for a single block would be

$$\text{transfer time} = \frac{1}{30 \times 60} = 0.55\ \textit{milliseconds}$$

Figure 10.6 shows a table of typical disks of different types, comparing various characteristics of the disks.

Since the total time required to access a disk block is approximately the sum of these three numbers, a typical disk access might require 20 to 25 msec. To put these speeds in perspective, consider that the typical modern computer can execute an instruction in less than 1 *nanosecond*. Thus, the CPU is capable of executing *millions* of instructions in the time required for a single disk access. This should make it very clear to you that disk I/O is a major bottleneck in processing and also that it is desirable to find other work that the CPU can be doing while a program is waiting for disk I/O to take place.

An expansion of part of a track to show a single data block is shown in Figure 10.7. The block consists of a header, 512 bytes of data, and a footer. An **interblock gap** separates the block from neighboring blocks. Figure 10.8 shows the layout of the header for a Windows-based disk. The track positions, blocks, and headers must be established before the disk can be used. The process to do this is known as **formatting** the disk. Since the header identifier must be a unique pattern of 1s and 0s, the data being stored must be checked by the disk controller to assure that the data pattern does not accidentally match the header identifier. If it does, the pattern stored on the disk is modified in a known way.

The entire track is laid down as a serial stream of bits. During write and read operations, the bytes must be deconstructed into bits and reconstructed.

Because the transfer speed of the disk is not the same as that required to transfer the block to memory, buffering is provided in the disk controller. The buffer is a first-in, first-out buffer, which receives data at one speed and releases it as required at the other speed. Buffer memory also makes it possible to read a group of blocks in advance so that requests for subsequent blocks can be transferred immediately, without waiting for the disk. Most modern disks provide substantial buffers for this purpose.

It is important to realize that the layout of the disk as discussed here does not take into account the structure of the files stored there, nor does it naturally provide a filing system. There is no direct relationship between the physical size of the block and the logical size of the data block or file that it contains, other than that the data must fit into the physical block or provisions made to extend the data to another block. It is also possible to store multiple logical blocks in a single physical block, if they fit.

File organization issues and the allocation of physical blocks for storage are within the domain of the operating system software, not the disk controller. File storage and allocation issues are discussed extensively in Chapter 17.

Before leaving the subject of disks, it will be useful to review briefly some of the material from Chapter 9 to give you an overview of the typical disk I/O operation. You will recall that the CPU initiates a request to the disk controller and that the disk controller does most of the work from that point on. As you now know from this chapter, the disk controller identifies

FIGURE 10.6

Characteristics of Typical Disks

Disk type	*Platters/ heads*	*Cylinders*	*Sectors per track*	*Block size*	*Capacity*	*Rotational speed*	*Avg. seek time read/write*	*Latency*	*Sustained transfer rate*
Professional SCSI	4/8	74,340	avg. 985	512 Bytes	300 GB	15,000 RPM	3.5–4 msec	2 msec	var. 75–120 MB/sec
Desktop	3/6	est. 102,500	variable	512 Bytes	1 TB	7200 RPM	8–9 msec	4.2 msec	115 MB/sec
DVD-ROM	1/1	spiral	variable	2352 Bytes	4.7–9.4 GB	variable, 570–1600 RPM (1x)	100–600 ms	variable	2.5 MB/sec (1x)
Blu-ray DVD	1/1	spiral	variable	2352 Bytes	24–47 GB	variable, 820–2300 RPM (1x)	variable	variable	4.5 MB/sec (1x)

Notes: (1) Hard disk data courtesy of Seagate Technology
(2) (1×) represents standard DVD speed, higher speeds and data rates are possible

FIGURE 10.7

A Single Data Block

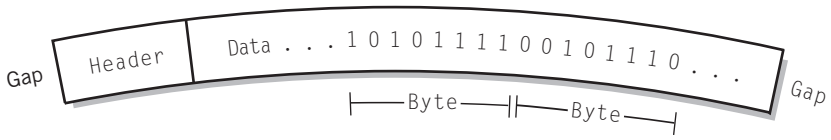
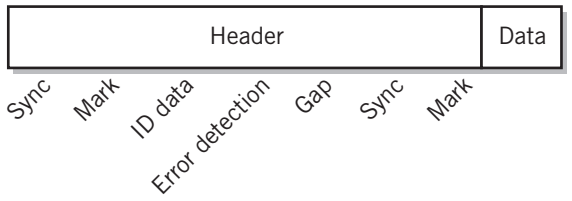


FIGURE 10.8

Header for Windows Disk



the disk block to be located, moves the head to the correct track, then reads the track data until it encounters the header for the correct block. Assuming that it is performing a read, it then transfers the data from the disk to a buffer. From the buffer, the data is transferred to conventional memory using DMA. Once the DMA transfer is complete, the disk controller notifies the CPU with a completion interrupt.

Disk Arrays

In larger computer environments, with mainframe computers or large PCs that provide program and data storage facilities for a network, it is common to group multiple disks together. Such a grouping of two or more disk drives is called a **disk array** or a **drive array**. A disk array can be used to reduce overall data access time by sharing the data among multiple disks and also to increase system reliability by providing storage redundancy. The assumption made is that the number of blocks to be manipulated at a given time is large enough and important enough to justify the additional effort and additional space requirements. One useful type of disk array is known as **RAID**, which stands for **Redundant Array of Inexpensive Disks**. (Some people say "Redundant Array of Independent Disks").

There are two standard methods of implementing a disk array. One is known as a **mirrored array**, and the other as a **striped array**.

A mirrored array consists of two or more disk drives. In a mirrored array, each disk stores exactly the same data. During reads, alternate blocks of the data are read from different drives, then combined to reassemble the original data. Thus, the access time for a multiblock read is reduced approximately by a factor equal to the number of disk drives in the array. If a read failure occurs in one of the drives, the data can be read from another drive and the bad block marked to prevent future use of that block, increasing system reliability. In critical applications, the data can be read from two, or even three, drives and compared to increase reliability still further. When three drives are used, errors that are not detected by normal read failures can be found using a method known as **majority logic**. This technique is particularly suitable for highly reliable computer systems known as **fault-tolerant computers**. If the data from all three disks is identical, then it is safe to assume that the integrity of the data is acceptable. If the data from one disk differs from the other two, then the majority data is used, and the third disk is flagged as an error.

The striped array uses a slightly different approach. In a striped array, a file segment to be stored is divided into blocks. Different blocks are then written simultaneously to different disks. This effectively multiplies the throughput rate by the number of data disks in the array. A striped array requires a minimum of three disk drives; in the simplest configuration, one disk drive is reserved for error checking. As the write operation is taking place, the system creates a block of parity words from each group of data blocks and stores that on the reserved disk. During read operations, the parity data is used to check the original data.

There are five well-defined RAID standards, labeled RAID 1 through RAID 5, and a number of additional proprietary and nonstandard varieties, including one labeled RAID 0. The most common of these are RAID 0, RAID 1, and RAID 5.

RAID 1 is a mirrored array as described above. RAID 1 provides protection by storing everything at least twice, but offers a substantial performance gain, particularly under heavy data read usage. RAIDs 2, 3, and 4 are arrays that are striped in different ways. Each uses a separate disk for error checking. Since data on every disk must be checked, this can create a roadblock on the single disk that is used for error checking. RAID 5 eases the roadblock by spreading the error-checking blocks over all of the disks.

RAID 0 is not a true RAID, because it provides no redundancy and no inherent error checking. Data is striped across all of the disks, primarily for fast access. However, the lack of redundancy means that a failure of *any* single disk block in the array corrupts all of the data in the system. However, this shortcoming can be overcome with proper backup and with certain types of journaling file systems, which we will discuss in Chapter 17. It is also possible to ''nest'' RAIDs. For example, we can use a pair of RAID 0 groups inside RAID 1 to achieve mirrored redundancy. The combination is known as RAID 0+1. With or without the additional protection, RAID 0 is sometimes attractive as a low-cost method of achieving high data transfer rates when they are required.

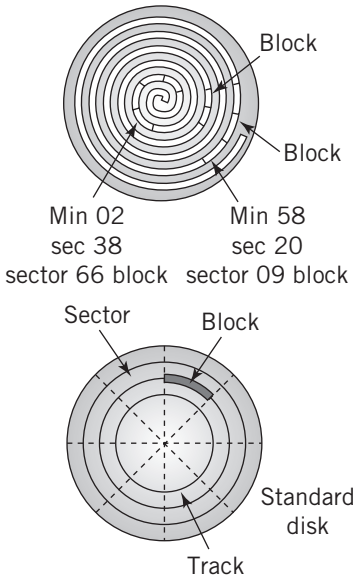
A number of vendors provide RAID controller hardware. particularly for large RAID 5 systems. With RAID controller hardware, RAID processing takes place within the array controller. The array appears as a single large disk drive to the computer. It is also possible to create a RAID using conventional, off-the-shelf disk controllers and operating system software. Although this uses CPU processing time, modern computers have enough spare power to make this a practical solution in many instances. It also reduces the possibility that a single RAID controller can cause the entire array to fail.

10.4 OPTICAL DISK STORAGE

An alternative to magnetic disk storage is optical storage. Optical storage technologies include various types of CDs and DVDs, in read-only, write-once, and read/write forms. Optical disks are portable and are capable of packing a relatively large amount of data into a convenient package. For example, an inexpensive CD-ROM, 12 centimeters in diameter, stores approximately 650 MB, while a Blu-Ray DVD of the same physical size can hold more than 50 GB of data. (There is also a standard for a new optical disk, called HVD, for Holographic Disk, that, when fully developed, is expected to hold more than 1.6 TB, but presently the cost is too high for most uses.) Optical storage serves a different purpose from magnetic disk storage. While magnetic disk storage serves primarily to store, read, and write data for current use, optical storage is intended more for offsite archiving, as well

FIGURE 10.9

Layout of a CD-ROM versus a Standard Disk



as program and file distribution, although the latter use has declined somewhat due to the growth of the World Wide Web.

CDs and DVDs used for data storage use the same basic disk format as their audio and video equivalents. Within certain file structure limitations, personal computer CD and DVD drives can read and write audio and video CDs and DVDs that will play on home media equipment and vice versa.

Conceptually, **CD-ROM** data storage is similar to magnetic disk: data is stored in blocks on the disk. The blocks can be arranged in files, with a directory structure similar to that of magnetic disks. The technical details are very different, however. Figure 10.9 compares the layout of a CD-ROM to that of a sectored magnetic disk. Rather than concentric tracks, data on a CD-ROM is stored on a single track, approximately three miles long, which spirals from the inside of the disk to the outside. Instead of sectors, the data is stored in linear blocks along the track. It should be remembered that the CD design was originally intended primarily for audio applications, where most data access is sequential, from the start of a musical selection to its finish; thus, a single spiral track was a reasonable decision.

Since the CD format was designed for maximum capacity, the decision was made to pack the bits on the disk as tightly as possible by making each block the same length along the spiral track, regardless of location on the disk. Thus, the disk is read at a constant linear velocity (i.e., CLV), using a variable speed motor to keep the transfer rate constant. Since the angle of a block is smaller on the outer tracks, the disk moves more slowly when outside tracks are being read. This is easily observable if you have access to a portable CD player that allows you to observe the disk as it rotates.

A CD-ROM typically stores 270,000 blocks of data. Each block is 2352 bytes long and holds 2048 bytes of data. In addition, there is a 16-byte header, which provides 12 bytes to locate the start of a block and 4 bytes for block identification. Due to the difficulty of the manufacturing process, errors can occur, so the CD-ROM provides extensive means for correcting the errors. Therefore, each block also provides 288 bytes of an advanced form of parity known as cross-interleaved Reed-Solomon error correcting code. This code repairs not only isolated errors but also groups of errors that might result from a scratch or imperfection on the disk. The resulting total data capacity of a single CD-ROM is approximately 550 MB. The error correction is occasionally omitted for applications where errors can be tolerated, such as audio, which increases the capacity of a CD-ROM to about 630 MB.

Blocks on a CD-ROM are identified by a 4-byte identification code that was inherited from the audio origins of the medium. Three bytes, stored in binary-coded decimal (BCD) format, identify the block by minute, second, and sector. There are 75 sectors per second and 60 seconds per minute. Normally, there are 60 minutes, although this number can be increased to 70 minutes if necessary. This increases the disk capacity to about 315,000 blocks. The fourth byte identifies a mode of operation. Mode 1, the normal data mode, provides the data as we've described, with error correction. Mode 2 increases the capacity by eliminating the error correction. Other modes are provided for special audio and video

features. It is possible to mix data, audio, and video on the same disk. Data blocks on CD-ROMs are sometimes called *large frames*.

Data is stored on the disk in the form of pits and lands. These are burned into the surface of the master disk with a high-powered laser. The disk is reproduced mechanically, using a stamping process that is less expensive than the bit-by-bit transfer process required of magnetic media. The disk is protected with a clear coating. Figure 10.10 shows a basic diagram of the read process. A laser beam is reflected off the pitted surface of the disk as a motor rotates the disk. The reflection is used to distinguish between the pits and lands, and these are translated into bits.

On the disk itself, each 2352-byte data block, or large frame, is broken up into 98 24-byte small frames. Bytes are stored using a special 17-bit code for each byte, and each small frame also provides additional error correcting facilities. Translation of the small frames into more recognizable data blocks is performed within the CD-ROM hardware and is invisible to the computer system. The bit-encoding method and additional error correction built into the small frames increases the reliability of the disk still further.

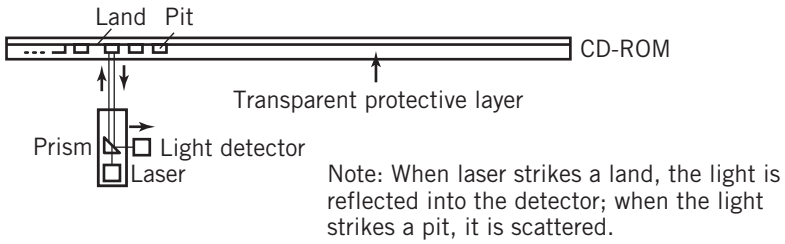
DVD technology is essentially similar to CD-ROM technology. The disk is the same size, and is formatted similarly. However, the use of a laser with a shorter light wavelength (visible red, instead of infrared) allows tighter packing of the disk In addition, the laser can be focused in such a way that two layers of data can be placed on the same side of the disk, one underneath the other. Finally, a different manufacturing technique allows the use of both sides of a DVD. Each layer on a DVD can hold approximately 4.7 GB. If both layers on both sides are used, the DVD capacity is approximately 17 GB. The use of a blue laser extends this capability even further, to approximately 50 GB.

WORM, or **write-once-read-many-times**, disks were originally designed to provide an inexpensive way for archiving data. WORM disks provide high-capacity storage with the convenience of compact size, reasonable cost, and removability. As the name indicates, WORM disks can be written, but, once written, a data block cannot be rewritten. The inability to tamper with the data on a WORM disk has taken on importance in business, where the permanence of many business data archives is required for legal purposes. When a file is updated, it is simply written again to a new block and a new directory entry is provided. Thus, a complete audit trail exists automatically. When the disk is filled, it is simply stored away and a new disk used.

WORM disks work similarly to a CD or DVD. The major difference is that the disk is made of a material that can be blistered by a medium-power laser. Initially, the entire disk is smooth. When data is to be written, the medium-power laser creates tiny blisters in the

FIGURE 10.10

CD-ROM Read Process



appropriate locations. These correspond to the pits in a normal CD-ROM. The WORM disk is read with a separate low-power laser in the same way as a CD-ROM.

This blister technology is used in various CD and DVD formats, called CD-R, DVD-R, and DVD+R. Additionally, there are rewriteable versions of this technology. These are known as CD-RW, DVD-RW, DVD+RW, DVD-RAM, and DVD+RAMBD-RE. There are file compatibility issues between the different WORM and rewriteable CD and DVD formats. Some drives will read every format; others will only read some of the formats.

10.5 MAGNETIC TAPE

Magnetic tape is used by many companies for backups and archives in large computer systems. Like other magnetic media, tape is nonvolatile, and the data can be stored indefinitely. Note that tape is a sequential medium, which makes it impractical for random access tasks. Generally, full system backups are made to tape and moved to offsite locations for long term storage.

There are several basic tape layouts, but all current formats are cartridge-based mechanisms. Regardless of type, the tape cartridge is removable from the tape drive for offline storage. When the tape is in the tape drive, ready for operation, it is said to be **mounted**. Tape cartridges have the major advantage of convenience. They are easy to mount and dismount, and small and easy to store. Current tape cartridges can store as much as 1.6 TB of compressed data or 800 GB of uncompressed data. Cartridges with uncompressed capacities as large as 4 TB are currently in development.

There are two main categories of data cartridge formats in use. The LTO (*linear tape open)* formats are representative of **linear recording cartridges**. An LTO format data cartridge is shown in Figure 10.11. The LTO format typically holds up to 820 meters of

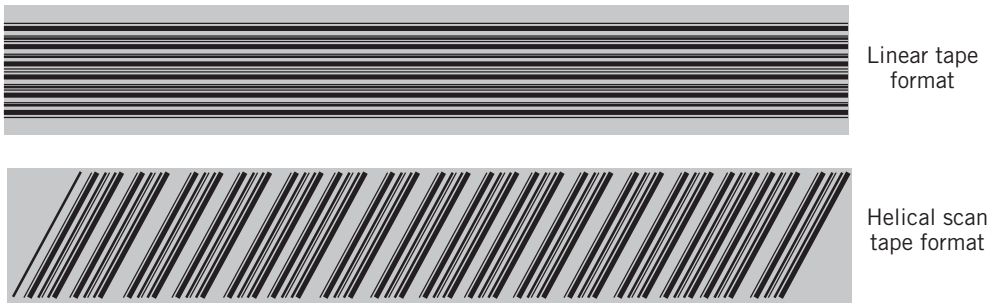
FIGURE 10.11

Tape Cartridge with Top Removed

Image from Wikipedia, http://en.wikipedia.org/wiki/File:LTO2-cart-wo-top-shell.jpg

FIGURE 10.12

Data Cartridge Formats



one-half inch wide tape in a 102 mm × 105 mm × 21.5 mm cartridge. The technique used for storage and retrieval is called **data streaming**. The cartridge tape is divided longitudinally into many tracks, currently as many as 886. The tape mechanism writes and reads the bits longitudinally, along the length of one group of tracks. At each end, the tape reverses, and the next group of tracks are written or read. Data is usually stored on the tape starting with the centermost track and moving outward toward the edge of the tape. Error correction is built into the system, and WORM archiving is also available as an option.

An alternative data cartridge format is based on the technology that was originally developed for videotape. These are called **helical scan cartridges**. The data on helical scan cartridges is very tightly packed, using a read/write head that rotates at a high speed to pack the tape more tightly with data. This results in a track that is made up of diagonal lines across the width of the tape. There are two different helical scan cartridges in common use. The smaller AIT (*advanced intelligent* format) uses 8-mm wide tape in tape lengths of up to 246 meters, with a current maximum uncompressed capacity of 400 GB in a cartridge 95 mm × 62.5 × 15 mm. The larger SAIT *(super-AIT*) cartridge contains up to 640 meters of one-half inch wide tape, with a current maximum uncompressed capacity of 800 GB. The SAIT cartridge is the same size as the LTO cartridge, but the two types of cartridge are not interchangeable.

Figure 10.12 shows the track layouts for both types of cartridges.

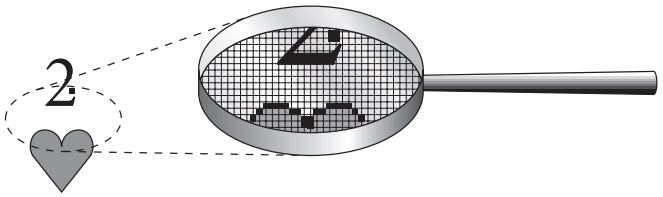
10.6 DISPLAYS

As viewed by the user, a display is an image made up of thousands of individual **pixels**, or picture elements, arranged to make up a large rectangular screen. Each pixel is a tiny square on the display. The layout for a display is shown in Figure 10.13. Older display screens have a horizontal to vertical ratio of 4:3. More recent displays are typically 16:9, described as ''widescreen''. A typical 4:3 screen is made up of 768 rows of 1024 pixels each, known as a 1024 × 768 pixel screen. Screens with resolutions of 1280 × 1024 pixels, or higher have also become common, especially on physically larger screens. Typical 16:9 screens are 1280 × 720 or 1920 × 1080.

Displays are specified by their screen sizes are measured diagonally. Figure 10.14 shows the relationship between the horizontal, vertical, and diagonal dimensions. The **resolution** of the screen is specified either as the size of an individual pixel or as the number of pixels per inch. The pixel size for a typical 15.4-inch wide laptop screen with 1280 × 720 pixel

FIGURE 10.13

Layout for a Display

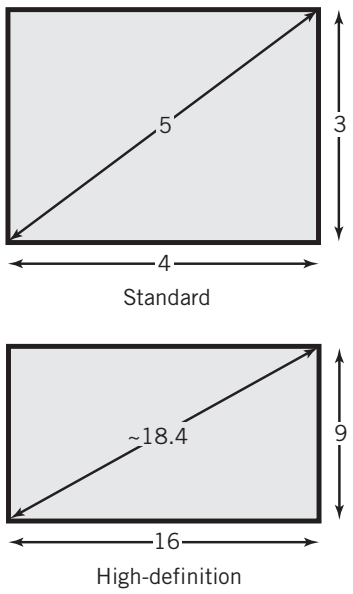


resolution is approximately 0.01 inch square, or about 100 pixels per inch resolution. Since the resolution essentially specifies the minimum identifiable object size capability of the monitor, the larger the number of pixels per inch, the better.

As we noted in Chapter 4, each individual pixel represents a shade of gray (on a monochrome screen) or a color. A color pixel is actually made up of a mixture of different intensities of red, green, and blue (RGB). We could represent a black-and-white image with 1 bit per pixel (for example, on for white, off for black), but, more typically, a color display would present at least 256 colors, and normally many more. It takes 2 bytes per pixel to represent a 65,536-color image, considered the minimum acceptable for Web use. A more sophisticated system would use 8 bits per color, or 24 bits in all. Such a system can present $256 \times 256 \times 256$, or more than 16 million, different colors on the screen and is sometimes described as a **true color** system. There are even a few 30-bit and 36-bit systems.

FIGURE 10.14

Display Screen Ratios



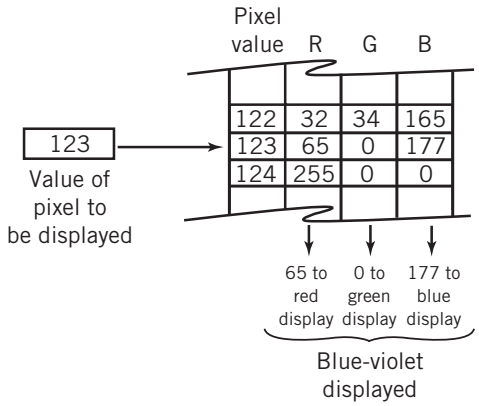
Even 16 bits per pixel requires a substantial amount of video memory. To store a single 1024-pixel by 768-pixel graphic image requires 1.55 MB of memory. A 24-bit-per-pixel image of the same size would require over 2.3 MB.

With 8 bits, there is no way to divide the bits to represent reds, blues, and greens equally. Instead, 256 arbitrary combinations of red, blue, and green are chosen from a larger palette of colors. The 256 colors might be chosen by the artist who created the image. More commonly, a default color scheme is used. Originally designed by Netscape for its Web browser, the default color scheme presents a reasonably uniform selection of colors ranging from black to white. Each selected color is represented by a red value, a green value, and a blue value that together will present the selected color on the screen. Most commonly, the system will use 1 byte for each color, providing an overall palette of sixteen million colors to choose from.

Each pixel value is represented by a value between 0 and 255, representing the color for that pixel. A color transformation table, also known as a palette table, holds the RGB values for each of the 256 possible colors. A few rows of a color transformation table are shown in Figure 10.15. To display a pixel on the screen, the system transforms the pixel color to a screen color by reading the RGB values that correspond to the particular pixel value from the table. The RGB colors are

FIGURE 10.15

Use of a Color Transformation Table



then sent to the screen for display. Although this transformation requires an extra step, the task is performed in special circuitry on the video card and is not difficult to implement.

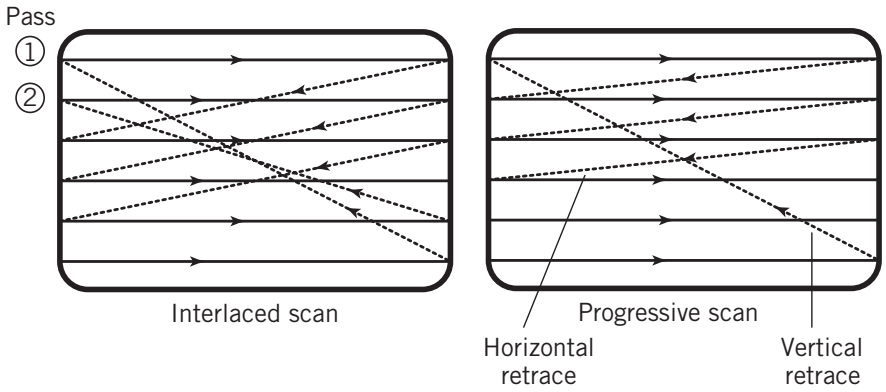
Transformation is also required for a display of sixty-four thousand colors, which uses 16 bits per pixel, however, a 24-bit color can be divided equally into three bytes, one for each color, so no transformation table is required.

In a modern system nearly all output, including text data, is presented graphically. For graphical output, values for each pixel on the screen are produced by a program, then stored in a display memory. Usually, the display memory is separately associated directly with a graphics display controller. On inexpensive personal computers, video memory is sometimes allocated as part of the regular memory.

The actual display is produced by scanning and displaying each pixel, one row at a time, from left to right, top to bottom. This method of displaying all the pixels is known as a **raster scan**. It is essentially identical to the way that television pictures are generated. When one row has been displayed, the scanner returns to the left edge and scans the succeeding row. This is done for each row, from top to bottom. This process is repeated more than thirty times a second. Most display monitors scan each row in turn, row 1, row 2, row 3, and so on. Some monitors **interlace** the display, by displaying the odd rows, row 1, row 3, row 5, and so on, and then coming back and displaying the even rows. Interlacing the rows is less demanding on the monitor, since each row is only displayed half as often, but results in flickering that is annoying to some users. Figure 10.16 shows the difference between interlaced and noninterlaced displays. Noninterlaced displays are also

FIGURE 10.16

Interlaced versus Progressive Scan Raster Screen



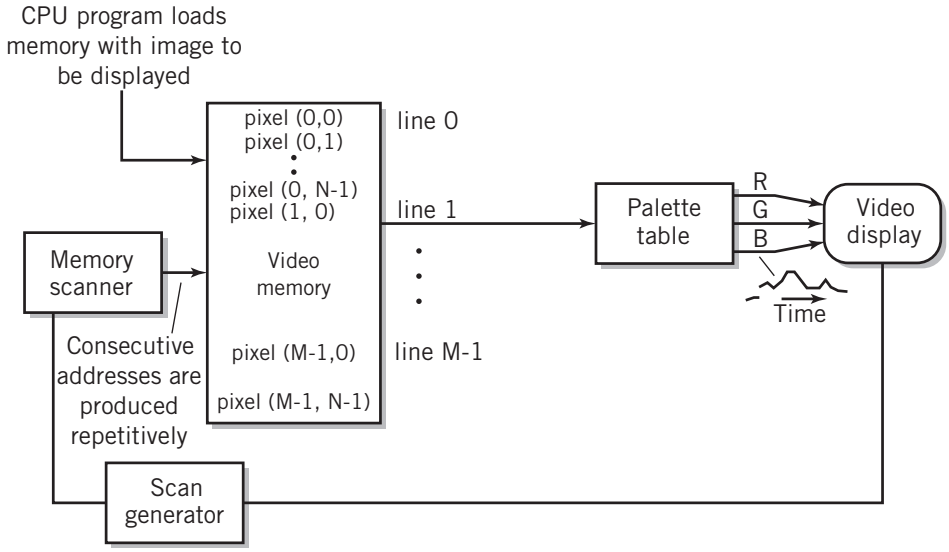
sometimes called **progressive scan displays**. Although some TV displays are interlaced, most computer display monitors are noninterlaced,

An alternative to raster scan is **vector scan**, in which pixels are displayed in whatever order is necessary to trace out a particular image. Vector scan could trace a character, for example, by following the outline of the character. Vector scan is obviously not suitable for bit map graphics, but can be used with object graphics images, such as those used for CAD/CAM applications. Generating vector scan images on a display screen is electronically much more difficult and expensive than producing raster scans, consequently, raster scans are used almost universally today for display. Vector scans are sometimes used when printing object graphics-based drawings to a plotter.

Figure 10.17 is a simplified diagram of the process that is necessary to produce a raster scan image. Each value to be displayed is read from the appropriate location in video memory in synchronization with its appearance on the screen. Although a palette table is shown in the figure, a 3-byte value would be read directly from video memory to the RGB display inputs when 24-bit color is used. A scan generator controls both the memory scanner and the video scanner that locates the pixel on the display screen. For normal images displayed graphically on a noninterlaced monitor, the values are stored consecutively, row by row, so that each traverse through memory corresponds to a single complete scan of the image. Video memory is designed so that changes in the image can be made concurrently by the CPU while the display process is taking place. The display process is illustrated with a simple example.

FIGURE 10.17

Diagram of Raster Screen Generation Process



EXAMPLE

Suppose our system has a 7-pixel by 5-pixel display monitor. On that monitor we wish to display an ''X''. The desired output is shown in Figure 10.18a. The different pixels on the ''X'' are to be colored as shown in the figure.

To support the display, our system provides 35 bytes of video memory. Each byte corresponds to one location. Since each location holds 1 byte, the system supports up to 256 different colors. The display memory is shown in Figure 10.18b. The memory is the usual type of linear memory, but we have redrawn it so that you can see more easily the relationship between the memory and the display. If you look carefully, you can see the ''X'' in this figure. Initially, the video memory was set to all zeros, where zero represents the background color. Presumably, a program in the CPU has since entered the data that represents the figure ''X'' to be currently displayed.

The table in Figure 10.18c represents the color translation palette for this example. The table has a red, blue, and green column for each entry. In our system, each RGB entry in the table holds a 6-bit number. This means that this system can produce $64 \times 64 \times 64 = 256K$ different colors. The RGB value (0, 0, 0) would produce black, the value (63, 0, 0) would produce pure red (i.e., maximum red, no green, no blue), and (63, 63, 63) would produce white. In this case, you can see from the table that the background color for the screen is white.

The display controller reads each memory location in turn, looks up the three values in the palette table, and displays the corresponding pixel on the screen. This process repeats indefinitely. The red, blue, and green signals that go to the video system as a result of the display operation are shown in Figure 10.18d. This pattern will be repeated over and over again, at least thirty times a second, until the display is changed. Notice that the red and blue signals are identical, since red and blue always appear together in maroon and are both totally absent from green.

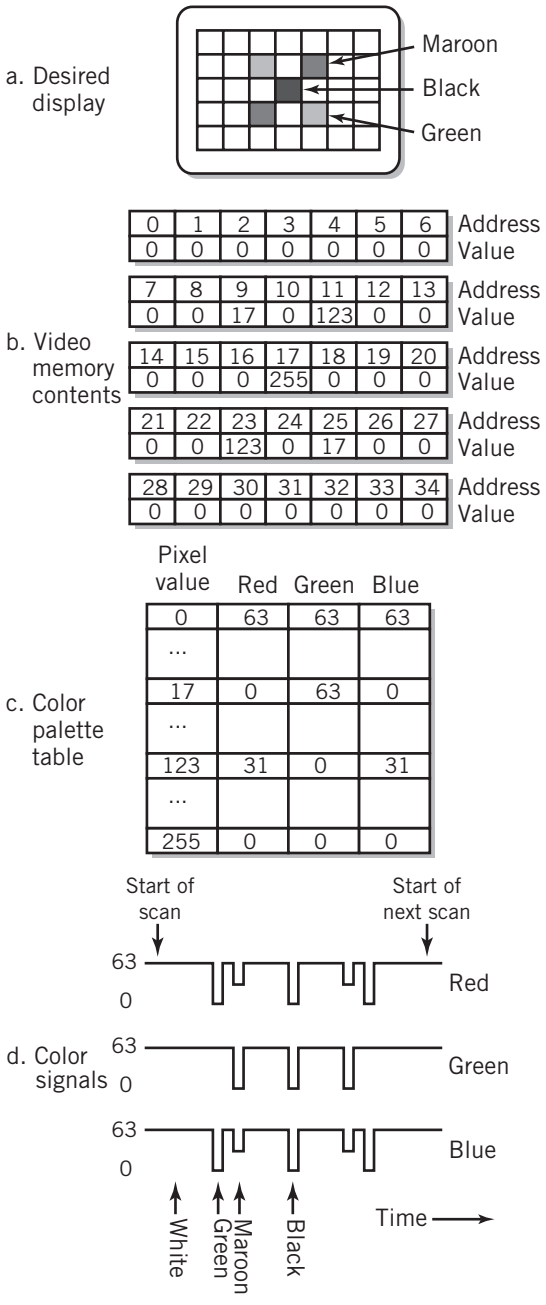
As noted, the method just described is used for graphical images. Since characters are also represented by displaying pixels, most modern computers also treat character output graphically; the popularity of what-you-see-is-what-you-get (WYSIWYG) output requires the ready availability of different fonts and flexibility in layout on the screen. Both these requirements are easily met with the graphical techniques already described.

Some systems, particularly older systems, provide an additional method for dedicated character output. In this method, usually called **text mode**, the pixels of the display screen are divided into blocks, often twenty-five rows of eighty, although other values are often also provided. Each block will display a single ASCII character. Instead of storing individual pixels, the video memory is used to store the ASCII values of the characters to be displayed. Many PCs start up in text mode.

Pixels are displayed on the screen in the usual way. To convert the characters to a raster scan line, the display controller provides a set of character-to-pixel tables, stored in ROM. As each character is read from memory, the appropriate pixels are pulled from the table and output to the screen. Most controllers limit the display output to the fonts that are provided in ROM. Some controllers also provide video memory that can be used to download additional character conversion tables. Most systems also include an ASCII extension set that provides simple graphical shapes for drawing lines and boxes, as well as facilities for creating underlines, blinking characters, and color changes of the character or the block. Note that in text mode it is not possible to alter individual pixels. All addressing of the screen must be performed by block.

FIGURE 10.18

Display Example: (a) Desired Display, (b) Video Memory Contents, (c) Color Palette Table, (d) Color Signals



Every pixel in a graphics display must be stored and manipulated individually; therefore, the requirements for a graphic display are much more stringent than those for a character display. Also, text mode display has the advantage that it requires significantly less memory than does graphics mode. As the price of memory has declined rapidly, this has become less of an issue. Text mode has one important additional advantage, however. Text data can be transmitted to a terminal located remotely from the computer much more compactly and efficiently in text mode than in graphics mode. It is obviously easier to transmit a single character than the dozens of pixels that make up the image of that character. Because of this, some terminals are still character based, particularly in business environments where most of the data is alphanumeric.

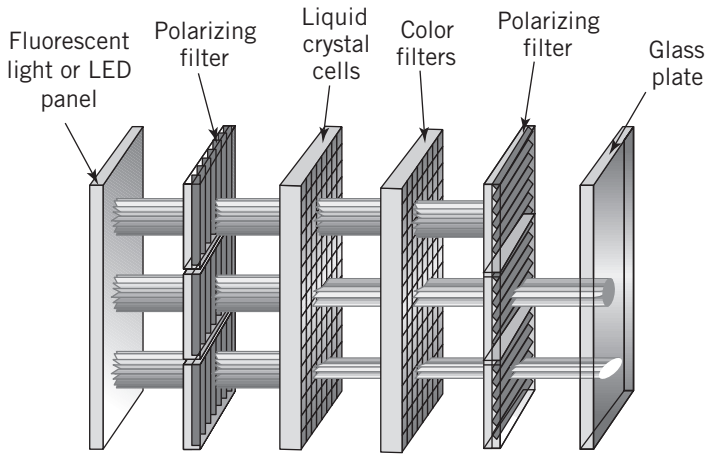
A compromise between the simplicity of text mode and the elegance of graphics mode is to transmit the data using an object-based description language such as PostScript. Fonts described in this way are known as **outline fonts**. By contrast, those fonts that are described by laying out the detailed pixel diagram for the characters are known as **bitmapped fonts**. Outline fonts and graphics described by page description languages have the additional advantage that they may be scaled easily to different sizes and rotated to different angles. The graphic image is then reconstructed at the terminal by translation software that is built into the display controller. This method is particularly amenable to printers and to Postscript displays used for precision graphical and layout work. The methods of managing graphical images are explored more fully in Chapter 16.

Liquid Crystal Display Technology

Although CRT display technology is still in use, liquid crystal display technology has become the prevalent means of displaying images. A diagram of a **liquid crystal display** (**LCD**) is shown in Figure 10.19. A fluorescent light or LED panel, located behind the display, produces white light. A polarizing filter in front of the light panel polarizes the

FIGURE 10.19

Liquid Crystal Display



light so that most of it is polarized in one direction. The polarized light then passes through a matrix of liquid crystal cells. In a color display, there are three cells positioned properly for each pixel. When an electrical current is applied to one of these cells, the molecules in the cell spiral. The strongest charge will cause the molecules to spiral 90 degrees. Since the light is passed through the crystal, its polarization will change, the amount depending on the strength of the electrical current applied.

Therefore, the light coming out of the crystal is now polarized in different directions, depending on the strength of the current that was applied to the crystal. The light is now passed through a red, blue, or green color filter and through a second polarizing filter. Because a polarizing filter blocks all light that is polarized perpendicular to its preferred direction, the second filter will only pass through the light that is polarized in the correct direction. Therefore, the brightness of the light is proportional to the amount of polarization twist that was applied by the liquid crystal's spiral.

There are several different ways of applying the electric current to the crystal. In an **active matrix** display, the display panel contains one transistor for each cell in the matrix. This guarantees that each cell will receive a strong charge, but is also expensive and difficult to manufacture. (Remember that even one imperfect cell will be apparent to the viewer!) A less expensive way provides a single transistor for each row and column of the matrix and activates each cell, one at a time, repetitively, using a scan pattern. This type of panel is known as a **passive matrix** display. The charge is applied for less time and is therefore lower. The result is a dimmer picture. Most modern LCD displays use the active matrix approach.

LCD panels have the advantage of bright images, no flicker, low power consumption, and thinness, so they are ideal for laptop computers. They are also used in most desktop displays. Because they are essentially flat, they can be placed anywhere. The same technology is also used for large-screen computer projectors.

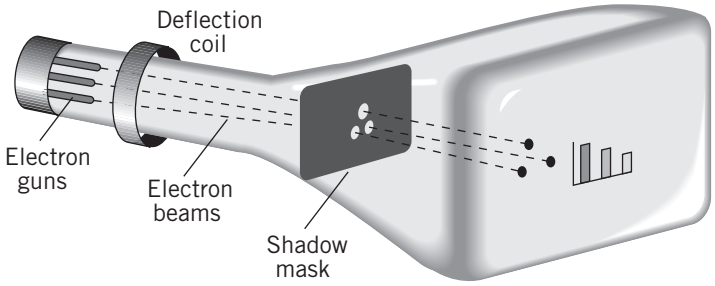
CRT Display Technology

With CRT technology, the image is produced on the face of a **cathode ray tube** (**CRT**), using a methodology similar to that used for older television receivers. A diagram of a color cathode ray tube is shown in Figure 10.20. Three **electron guns** within the tube shoot beams of electrons from the back of the tube. There is a gun for each of the primary colors, red, blue, and green. A high voltage applied to the inside of the face of the tube attracts the beams to the face. The face of the tube is painted with tiny dots or thin stripes of **phosphors**, which glow when struck by electrons. There are phosphors that glow red, blue, and green. A **shadow mask** in the tube is designed such that electrons from each gun can strike only phosphors of the matching color. The strength of the beams varies depending on the color and brightness of the point being displayed. The stronger the beam for a particular color, the brighter that color appears on the screen.

The three beams of electrons are *deflected* both horizontally and vertically by a pair of electromagnetic coils, so that the beam scans across the screen and top to bottom, to form the scan pattern that you already saw in Figure 10.16. Monochrome video monitors work identically, except that only a single gun is required, the phosphor is white, yellow, or green, and no shadow mask is required.

FIGURE 10.20

Diagram of a CRT



OLED Display Technology

OLED (**Organic Light-Emitting Diode**) technology is a new screen technology that is poised to supplement or replace LCD technology in display monitors. OLED technology offers an image that is brighter, with colors that are more vivid and with vastly improved contrast. Despite the improved image, the OLED panel consumes less power, with a package that is even thinner than current flat screen monitors. LCD technology is *passive* in the sense that light is generated by a backlight; the light is selectively blocked by the LCD cells in the panel. Leakage in the cells limits the level of darkness that can be achieved and the maximum brightness is limited by the brightness of the backlight.

In contrast, OLED technology is *active*. OLED technology consists of a thin display panel that contains red, green, and blue LEDs for each pixel with transistors for each LED that generate electrical current to light the LED. The light output is produced directly by these LEDs. The brightness of a pixel is determined by the amount of current supplied by the transistor, which in turn is determined by an input signal indicating the desired level of brightness. The simplicity of the panel, combined with the lack of need for a backlight, result in the thinness of the panel. Sony and Samsung have both demonstrated OLED panels less than 3 mm thick.

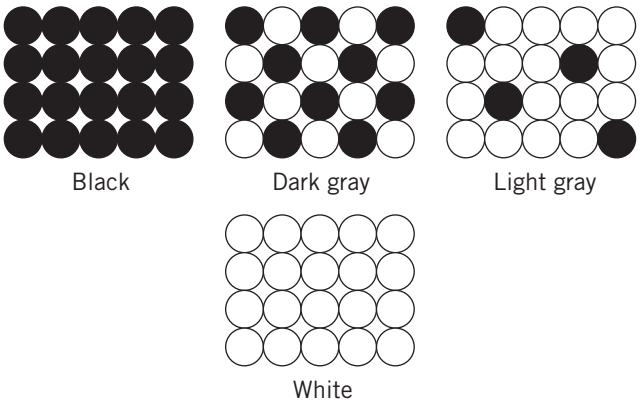
10.7 PRINTERS

Earlier printers were derived from typewriters. They used formed characters that were mounted at the ends of arms, on wheels shaped like a daisy, on chains, or on spheres. Printing resulted from the hammer-like physical impact of the character through an inked ribbon onto paper. These printers were difficult to maintain and were incapable of generating any character or graphical image that was not provided in the set of formed characters. Later **impact printers** used pins that were selectively employed to generate dot matrix representations of the characters on the page. These printers were known as *dot matrix* printers; in addition to the standard characters, they were also capable of printing simple geometric shapes. Impact printers have mostly disappeared from use.

Except for some commercial printing of items such as books, magazines, and newspapers, nearly all modern printing is done using nonimpact technologies. This is true

FIGURE 10.21

Creating a Gray Scale



regardless of the size of the system, the quantity of printing, or the capacity of the printer.[2] Single-color (usually black and white) printers normally use **laser**, or **inkjet** printing technology. Low-cost color printing also uses inkjet technology. More expensive color printing uses **thermal wax transfer** or **dye sublimation**, inkjet, or laser technology.

The impression on the paper is sprayed at the paper or laid down on the paper. Like displays, printer output can be character based or graphics based. Most printers have built-in character printing capability and can also download fonts. Nonetheless, much of the output from modern computers is graphics based, even when text is being printed, since graphics output produces more flexibility. The output to many printers takes the form of graphical bitmaps that represent the required pixels directly. Some printers have built-in computing capability and can accept data in the form of a page description language, predominantly Adobe **PostScript** or *PCL*, an industry standard print command language originally developed by HP. The controller in the printer can then convert from the page description language to the bitmap within the printer itself. Memory is provided within the printer to hold the bitmapped image while it is being printed.

Nearly all modern computer printers produce their output as a combination of dots, similar in style to the pixels used in displays. There are two major differences between the dots used in printers and the pixels used in displays. First, the number of dots per inch printed is generally much higher than the number of pixels per inch displayed. The number of pixels displayed usually ranges between about 70 and 150 per inch. Typical printers specify 600, 1200, or even 2400 dots per inch.

This difference in resolution is partially compensated for by the second major difference: the dots produced by most printers are either off or on. A few printers can vary the size of the dots somewhat, but, in general, the intensity, or brightness, of the dots is fixed, unlike the pixels in a display, which can take on an infinite range of brightnesses. Thus, to create a gray scale or color scale, it is necessary to congregate groups of dots into a single equivalent point and print different numbers of them to approximate different color intensities. An example of this is shown in Figure 10.21.

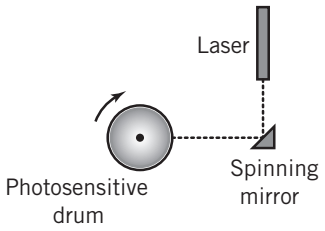
Laser Printers

Today, the prevalent form of printing for most applications is laser printing. Laser printing is derived from xerography. The major difference is that the image is produced electronically

[2]Even most modern commercial printing uses a nonimpact technique called *offset printing* that is based on contact between a rubber mat containing a print image and the paper, a method similar in many respects to laser printing. The impact printing press technology that you see in old movies is called *letterpress* printing.

FIGURE 10.22

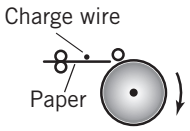
Operation of a Laser Printer



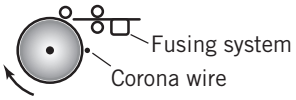
1. A laser is fired in correspondence to the dots that are to be printed. A spinning mirror causes the dots to be fanned out across the drum. The drum rotates to create the next line, usually 1/300th or 1/600th of an inch.

   The drum is photosensitive. As a result of the laser light, the drum will become electrically charged wherever a dot is to be printed.

2. As the drum continues to rotate, the charged part of the drum passes through a tank of black powder called toner. Toner sticks to the drum wherever the charge is present. Thus, it looks like the image.



3. A sheet of paper is fed toward the drum. A charge wire coats the paper with electrical charges. When it contacts the drum, it picks up the toner from the drum.



4. As the paper rolls from the drum, it passes over a heat and pressure area known as the fusing system. The fusing system melts the toner to the paper. The printed page then exits the printer.

   At the same time, the surface of the drum passes over another wire, called a corona wire. This wire resets the charge on the drum, to ready it for the next page.

from the computer using a laser or light-emitting diodes, rather than scanning a real image with a bright light, as in a copy machine. A description of the steps in the operation of a laser printer is shown in Figure 10.22. Color images are produced by printing the sheet four times with different colored toners.

Inkjet Printers

Inkjet printers operate on a simple mechanism that also has the advantages of small size and economy. Despite their simplicity, inkjet printers with high-quality inks are capable of photographic quality color output. Mechanically, the inkjet printer consists of a print

cartridge that moves across the page to print a number of rows of dots, and mechanical rollers that move the page downward to print successive rows.

The inkjet print cartridge contains a reservoir of ink and a column of tiny nozzles, so that several rows can be printed at once. Each nozzle is smaller than the width of a human hair. A dot is produced by heating the ink behind a nozzle. When the ink is boiled it sprays a tiny droplet of ink toward the paper. The volume of each droplet is about one-millionth the volume of the drop from an eyedropper of water! Some printers use a vibrating piezo-crystal instead of heat to produce the ink droplets. Multiple reservoirs of ink make it possible to print multiple colors.

Thermal Wax Transfer and Dye Sublimation Printers

For the highest quality color images, more specialized methods are required. The preferred methods are thermal wax transfer and dye sublimation. The mechanisms for both types are similar. The paper is fed into the printer and clamped against a drum. A print head provides a row of dot-sized heating elements. Between the paper and the print head, the printer feeds a roll of film that is impregnated with either colored wax or dye. The film is made up of page-sized sections of magenta, cyan, and yellow colors; sometimes an additional section of black is also included. Each rotation of the drum exposes the paper to a different color. The heat from the print head melts the wax or dye onto the paper.

Thermal wax can be applied to ordinary paper. To improve quality, some printers precoat the paper with clear wax. This compensates for slight imperfections in the paper so that the wax may be applied more uniformly. Different colors are produced in the same way that black-and-white printers produce gray scales.

The dye sublimation technique differs slightly, in that transparent dyes diffuse in the paper, so that the dots of color actually blend. Furthermore, it is possible to control the amount of dye by adjusting the temperature of individual print head elements. Thus, dye sublimation can print continuous color tones. Unfortunately, the dye sublimation technique also requires higher temperatures, therefore, special paper must be used.

10.8 USER INPUT DEVICES

Keyboards and Pointing Devices

Users use a variety of devices to interact with the computer, but most commonly, the modern user interface is based upon a keyboard and a pointing device. Keyboards consist of a number of switches and a keyboard controller. The keyboard controller is built into the keyboard itself. There are several different types of switches in use, including capacitive, magnetic, and mechanical. In most environments, the type of switch used is not important. Different types of switches feel differently when used. Some switches are more suitable than others for environments where dust or electrical sparks or the need for ultra-high reliability are a problem. When a key is pushed, a signal called a scan code is sent to the controller. A different scan code is sent when the key is released. This is true for every key on the keyboard, including special keys such as *Control, Alt*, and *Shift* keys. The use of two scan codes allows keys to be used in combination, since the controller is able to tell whether a key is being held down while another key is struck. The controller can also determine when a key is to cause a repeated action.

If the keyboard is part of a terminal, the scan codes are converted to ASCII, Unicode, or EBCDIC (see Chapter 4 if you need a reminder) and sent to the computer, usually via a serial port. Keyboards local to a computer such as a PC interrupt the computer directly. The scan codes are converted to ASCII or Unicode by software in the computer. This latter method allows more flexibility in remapping the keyboard for different languages and keyboard layouts.

Modern graphical user interfaces also require the use of a pointer device as input to locate and move a cursor on the display screen. The best known pointer device is a mouse, but there are other pointer devices in use, including light pens, touch screens, and graphics tablets, as well as the special pointer devices used for interfacing with computer games.

The simplest device is the mechanical mouse. As the mouse is moved across a surface, the roller ball protruding from bottom of the mouse also moves. Two wheels, mounted at a 90-degree angle from each other, touch the roller ball, and move with it. These wheels are called **encoders**. As the encoders move, they generate a series of pulses. The number of pulses corresponds to the distance that the mouse was moved. One encoder records movement forward and backward; the other records sideway motion. The pulses are sent to a program in the computer to interpret the current location of a cursor. Some encoders use a tiny light and sensor to create the pulses, others use a tiny mechanical switch, but the method used is not important. Desktop game pointing devices and trackballs work similarly. Space-based game controllers, such as the Nintendo Wii remote, use accelerometers to detect movement in all three dimensions; software in the game console then uses that information to perform the appropriate action upon the object of interest.

Light pens are used differently and work differently. A light pen is pointed at the screen to identify a position on the screen. By moving the pen around the screen, a cursor can be made to follow the pen. The light pen can be used to point to a target, such as a control button on the screen, and can also be used as a drawing tool. The light pen is not actually capable of telling the system its position. Instead, the software program that is used with the light pen rapidly generates pixels of light on the display screen at known locations in the area where the light pen is believed to be pointing. The light pen has a photodetector that can respond to the point of light on the screen, so when the point on the screen that corresponds to the light pen is lit, the light pen is activated, which notifies the program that the current location is correct.

Graphics tablets use a variety of techniques, including pressure-sensitive sensors, optical sensors, magnetic sensors, and capacitive sensors to determine the location of a pen on the pad. Some techniques require the use of a special pen, which is attached to the tablet, while others allow the use of any pointed object, such as a wooden pencil, with or without lead, or even a finger. The resolution and accuracy of graphics tablets depends on the technique employed. Graphics tablets can be used as mouse replacements, but are particularly suited for drawing. A similar mechanism is used for the touch pads commonly found on laptop computers.

Touch screens provide a capability similar to that of graphics tablets, but with the sensing mechanism attached directly to the display screen, allowing the user to point directly to an object on the screen. Touch screens are particularly popular on devices such as PDAs, cell phones, portable game consoles, and portable music and video players. They are also available on many commercial devices that require user interaction with the public, such as store self-checkout machines and information kiosks, as well as some personal computers. A number of different technologies can be used to detect the point of touch.

These technologies differ in cost, accuracy, and durability. Common technologies include resistive, capacitive, and surface acoustic wave. Some touch screens are capable of detecting multiple touch points.

Scanners

Scanners are the primary means used to input paper images. Although video frame grabbers and television cameras can also be used for this purpose, scanners are generally less expensive and more convenient.

There are three primary types of scanners, flatbed scanners, sheet-fed scanners, and handheld scanners, but all three work similarly and differ only in the way the scan element is moved with respect to the paper. In a flatbed scanner, the paper is placed on a glass window, while the scan element moves down the page, much like a copy machine. In a sheet-fed scanner, a single page of paper is propelled through the mechanism with rollers; the scan element is stationary. Handheld scanners are propelled by the user over the page.

Regardless of which means is used, the basic operation is the same. The scanning mechanism consists of a light source and a row of light sensors. As the light is reflected from individual points on the page, it is received by the light sensors and translated to digital signals that correspond to the brightness of each point. Color filters can be used to produce color images, either by providing multiple sensors or by scanning the image three times with a separate color filter for each pass. The resolution of scanners is similar to that of printers, approximately 600–2400 points per inch.

Multimedia Devices

Despite its importance in modern systems, not much needs to be said about this topic. Most modern personal and workstation computers provide input ports with an audio analog-to-digital converter for microphones and other audio input equipment, as well as an output converter and speakers and headphone jacks for audio output. USB ports can also be used for this equipment and for computer-compatible video cameras, TV tuners, and other multimedia devices.

10.9 NETWORK COMMUNICATION DEVICES

It is impossible to overemphasize the fact that, from the perspective of a computer, a network is simply another I/O device, a device that, like a disk, offers input to applications on the computer and receives output from applications on the computer. Like other I/O devices, there is a controller, in this case a **network interface unit** (**NIU**) **controller** or **network interface card** (**NIC**) that handles the physical characteristics of the connection and one or more I/O drivers that manage and steer input data, output data, and interrupts.

There are a number of different types of network interfaces, with different network interface controllers for each. On large mainframe systems, there may be network interface controllers for a variety of different network connections, including various flavors of Ethernet, FDDI fiber, token-ring, and other types. On most systems, the standard connection is to an Ethernet network. Nearly every current computer system is supplied with one or more Ethernet network interface cards as a basic part of the system. Wireless Ethernet and Bluetooth network interface cards are also commonplace.

The interface between a computer and a network is more complicated than that for most other I/O peripherals. Data must be formatted in specific ways to communicate successfully with a wide range of application and system software located on other computers. The computer also must be able to address a large number of devices individually, specifically, every other computer connected to the network, whether connected directly to the local network, or indirectly connected through the Internet. Unlike many device controllers, NICs must be capable of accepting requests and data from the network, independent of the computer, and must be able to provide interrupt notification to the computer. Security of communication is an important concern, whereas local devices normally require only minimal security considerations. Many of these concerns are handled with protocol software in the operating system. The NIC is responsible only for the electrical signals that connect the computer to the network, either directly or through a communication channel, and for the protocols, implemented in hardware, that define the specific rules of communication for the network. These protocols are called **medium access control** protocols, or **MACs**. We note in passing that every NIC and network device throughout the world has a unique address called a MAC address that can be used to identify the specific device and its characteristics. The MAC address is sometimes used by cable and DSL vendors to restrict network access to a specific device.

The hardware aspects of the network interface are considered more fully in Chapter 14. A deeper look at the fundamentals of networking infrastructure, including types of networks, the nature of communication channels, media, the movement of data across a network, protocols, and the operation of the Internet, is described in Chapters 12 and 13.

SUMMARY AND REVIEW

This chapter provides an overview of the workings of the most common computer peripheral devices. Peripheral devices are classified as input devices, output devices, and storage devices. We began by demonstrating that storage can be thought of hierarchically, with registers the most immediately available form of storage, followed by memory, and then the various peripheral devices. We discussed the trade-offs that make each form desirable for some purposes.

Following this general introduction, we introduced flash memory, and discussed its applications, strengths, and weaknesses.

Next, we showed the layout and explained the operation of various forms of disk, including hard magnetic and optical. We showed how the performance factors, capacity and various speed measures, are obtained. For each device we showed how a block is identified and located. We noted the difference between the concentric tracks used on magnetic disks and the spiral tracks used on many optical disks. We explained the difference between CAV and CLV operation. The discussion of disks is followed by a similar discussion for magnetic tape.

The display is the most important output device. We explained the process used to produce a display, from the bytes in memory that represent individual pixels or characters to the actual output on a screen. We showed that there are two different forms of output, character and graphic. We showed how colors are determined for the display. We also showed the basic technology for the two methods that are used to produce an image, video on a CRT, and liquid crystal display.

There are a number of different technologies used in printers. We introduced laser printers, inkjet printers, and thermal transfer printers as representative of the most important current technologies.

The chapter continues with a brief discussion of keyboards, various pointer devices, scanners, and multimedia devices that are used for input. We conclude with a brief consideration of the network as an input/output device, a discussion to be expanded greatly in later chapters.

FOR FURTHER READING

Much of the discussion in this chapter reviews material that you have seen before, probably in an introduction to computers course. Any good introductory textbook will also serve as a further reference for this chapter. In addition, there are several good books that describe I/O devices. White [WHIT05] provides straight-forward explanations of many I/O devices. Mueller [MUEL08] contains comprehensive treatments of memory, disk storage, optical disks, video hardware, and more.

KEY CONCEPTS AND TERMS

active matrix (LCD)
average seek time
bitmapped fonts
block (of data)
cathode ray tube (CRT)
CD-ROM
constant angular velocity (CAV)
constant linear velocity (CLV)
cylinder
data streaming
direct access storage devices (DASDs)
disk array
DVD
drive array
dye sublimation
electron guns
encoders
fault-tolerant computers
file server
flash memory
formatting
graphics tablet
hard disk drive
helical scan cartridge
impact printer
inkjet printer
interblock gap
interlace
laser printer
latency time
light pen
linear recording cartridge
liquid crystal display (LCD)
majority logic
medium access control (MAC)
mirrored array
mounted
multiple zone recording
network interface card (NIC)
network interface unit (NIU) controller
off-line storage
OLED (organic light-emitting diode) display
online secondary storage
outline fonts
parked (position)
passive matrix (LCD)
peripherals
phosphors
pixels
PostScript
primary memory
progressive scan display
raster scan
redundant array of inexpensive disks (RAID)
resolution
rotational delay
rotational latency time
secondary storage
sectors
seek time
server
shadow mask
striped array
text mode
thermal wax transfer
touch screen
track

transfer time
true color
vector scan
WORM (write-once-read-many-times) disks
zone bit recording (ZBR)
zone-CAV recording (Z-CAV)

READING REVIEW QUESTIONS

10.1 Peripheral devices can be categorized into three classes. What are the three classes? Give an example of each.

10.2 State at least three reasons why storage in a computer is organized hierarchically.

10.3 What is the advantage of flash memory over RAM? What is the advantage of RAM over flash memory? What is the advantage of flash memory over magnetic hard disk?

10.4 Draw a circle representing one platter surface of a hard disk. On your drawing show an example of a track, of a sector, and of a block.

10.5 Draw a sector representing one platter surface of a hard disk with sixteen sectors. On your drawing show a track, the sectors, and a single block. Place a magnetic head somewhere on your drawing. Show on your drawing the seek time, latency time, and read time for the block that you drew on the disk.

10.6 Suppose a disk is rotating at 7200 rpm. What is the minimum latency time for this disk? What is the maximum latency time for this disk?

10.7 What is a *disk array*? What advantages does a disk array offer over those of a single disk?

10.8 How does the layout of a typical optical disk differ from that of a magnetic disk? How many tracks does a standard single-layer CD-ROM contain?

10.9 What does *WORM* stand for when it is used to describe an optical disk?

10.10 What are the advantages and disadvantages of magnetic tape as compared to other peripheral storage devices?

10.11 What do the numbers 1920×1080 represent when describing a display?

10.12 How many pixels are there in a 1024×768 display? What is the picture ratio of this display?

10.13 What is true of the red, blue, and green pixel values if the color of the pixel is white? What if it's black?

10.14 What is the difference between interlaced scan and progressive (or noninterlaced) scan?

10.15 Explain how a raster scan works.

10.16 What are the advantages of LCD technology over CRT technology?

10.17 What does OLED stand for? How does OLED technology differ from LCD technology?

10.18 What are the two types of printers in primary use today?

10.19 What is the measure used to indicate the resolution of a printer?

10.20 Name at least three user input devices in common use.

10.21 What does NIC stand for?

EXERCISES

10.1 Explain why it is easy to perform read and write in place on a disk but not on a tape.

10.2 What are the advantages of flash memory over hard disk storage? What are the advantages of hard disk over flash memory storage? What are the advantages of both hard disk and flash memory storage over RAM? What is the major advantage of RAM over other types of storage?

10.3 A multiplattered hard disk is divided into 1100 sectors and 40,000 cylinders. There are six platter surfaces. Each block holds 512 bytes.

The disk is rotating at a rate of 4800 rpm. The disk has an average seek time of 12 msec.

a. What is the total capacity of this disk?

b. What is the disk transfer rate in bytes per second?

c. What are the minimum and maximum latency times for this disk? What is the average latency time for this disk?

10.4 The average latency on a disk with 2200 sectors is found experimentally to be 110 msec.

a. What is the rotating speed of the disk?

b. What is the transfer time for one sector?

10.5 Old fashioned twelve-inch laser video disks were produced in two different formats, known as CAV and CLV. The playing time of a CLV disk is approximately twice that of a CAV disk, although the number of tracks, track width of the tracks on the disk, and amount of data per video frame is the same. Explain why this is so.

10.6 An optical disk consists of two thousand concentric tracks. The disk is 5.2 inches in diameter. The innermost track is located at a radius of 1/2 inch from the center. The outermost track is located 2 1/2 inches from the center. The density of the disk is specified as 1630 bytes per inch along the track. The transfer rate is specified as 256,000 bytes per second. The disk is CLV. All blocks are of equal size.

a. The innermost track consists of ten blocks. How many bytes are contained in a block?

b. How many blocks would the outermost track contain?

c. The capacity of the disk is approximately equal to the capacity in bytes of the middle track times the number of tracks. What is the approximate capacity of the disk?

d. What is the motor rotation speed when reading the innermost track? the outermost track?

10.7 Why is the average seek time for a hard disk much shorter than for a CD-ROM or DVD-ROM?

10.8 There is a current proposal to cut the size of an individual bit in a DVD-ROM in half so as to increase the capacity of the disk. This would cut both the width

of the track and the track length required per bit in half. If the current capacity of a DVD-ROM is approximately 4.7 GB, what would be the capacity of the new "high-density" DVD-ROM?

10.9 A typical published page consists of approximately forty lines at seventy-five characters per line. How many published pages of 16-bit Unicode text would fit on a typical 600 MB CD-ROM? How many published pages of text would fit on a netbook computer with an 80 GB flash memory?

10.10 A high-quality photographic image requires 3 bytes per pixel to produce sixteen million shades of color.

- **a.** How large a video memory is required to store a 640 × 480 image during display? A 1600 × 900 image? A 1440 × 1080 image?
- **b.** How many 1024 × 768 non-compressed color images will fit on 4.7 GB DVD-ROM?

10.11 A 1024 × 768 image is displayed, noninterlaced, at a rate of thirty frames per second.

- **a.** If the image is stored with 64K-color resolution, which uses 2 bytes per pixel, how much memory is required to store the picture?
- **b.** How much video memory is required to store the picture as a "true color" image, at 3 bytes per pixel?
- **c.** What is the transfer rate, in bytes per second, required to move the pixels from video memory to the screen for the "true color" image?

10.12 For a motion picture image it may be necessary to change every pixel in the image as many as thirty times per second, although usually the amount of change is somewhat smaller. This means that without data compression or other tricks that a large number of pixel values must be moved from main memory to video memory each second to produce moving video images. Assume a video image on the screen of 1 1/2″ × 2″, with a pixel resolution of seventy-two dots per inch and a frame rate of thirty per second. Calculate the required data transfer rate necessary to produce the movie on the screen. Do the same for an image of 3″ × 4″.

10.13 The cost of a monitor increases rapidly with increasing bandwidth. The bandwidth of a monitor is measured roughly as the number of pixels displayed on the screen per second.

- **a.** Calculate the bandwidth of a 640-pixel by 480-pixel display operating in an interlace mode. One-half of the image is generated every 1/60th of a second.
- **b.** Do the same for a 1920-pixel by 1080-pixel display operating in noninterlace mode. One entire image is generated every 1/60th of a second.

10.14 A 1600-pixel by 900-pixel display is generated on a 14-inch (diagonal) monitor.

- **a.** How many dots per inch are displayed on this monitor?
- **b.** What is the size of an individual pixel? Would a .26 mm pixel resolution monitor be sufficient for this display?
- **c.** Repeat (a) and (b) for a 1280 × 720 display.

10.15 A text display displays 24 rows of 80 characters on a 640-pixel by 480-pixel 15-inch monitor. Assuming four spaces for horizontal space between each row of characters, how big are the characters in inches? In pixels? How big would a character of the same pixel size be if the display is increased to 800 × 600? How many rows of characters could be displayed in this case?

10.16 What is the actual resolution of a gray scale picture printed on a 600-dot-per-inch laser printer if the gray scale is created with a 3 × 3 matrix?

10.17 In printer jargon, ''replaceables'' are the items that are used up as part of the printing process. What are the replaceables in a laser printer? In an inkjet printer? In a dot-matrix impact printer?

10.18 Explain the difference in the method used to generate characters between graphics mode and character mode display.

10.19 Explain the difference between pixel graphics and object graphics, and discuss the advantages and disadvantages of each.

10.20 What are the limitations of typewriter-type (formed character) printers that caused them to fade from popularity?