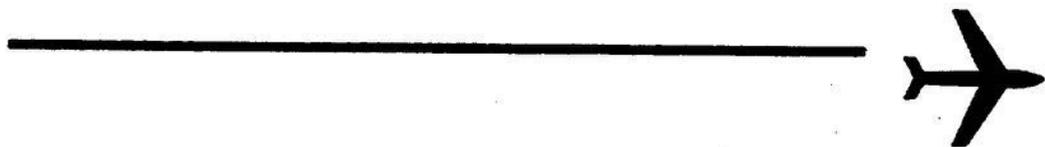


**CDC 3C251A**

**Communications-Computer  
Systems Control  
Journeyman**

**Volume 4. Computer Fundamentals  
and Digital Devices**



**Extension Course Institute  
Air Education and Training Command**

**Author:** MSgt Timothy L. Bearden  
335th Training Squadron  
Keesler Training Wing (AETC)  
Keesler Air Force Base, Mississippi, 39534-2235  
DSN 597-5161  
E-mail bearden@kee3b201.kee.aetc.af.mil

**Instructional Systems**

**Specialist:** James L. Coleman

**Editor:** Senta Washington

Extension Course Institute  
Air University (AETC)  
Maxwell Air Force Base, Gunter Annex, AL 36118-5643

**Material in this volume is reviewed annually for technical accuracy, adequacy, and currency. For SKT purposes the examinee should check the *Weighted Airman Promotion System Catalog* to determine the correct references to study.**

THIS FOURTH volume of CDC 3C251A, *Computer Fundamentals and Digital Devices*, deals with the basic equipment and terms associated with computers and digital communications devices used in systems control facilities around the world. In it, we continue to expand and build on the material presented in previous volumes. This information also familiarizes you with fundamentals that are building blocks for various systems and networks discussed in greater detail in Part B of this CDC.

Unit 1 covers characteristics of digital signals. We discuss new terms as well as some with which you should already be familiar. It is important to have a basic knowledge of these terms in order to understand the operation of modern computer networks. Also covered are the industrial and military electrical standards that apply to digital signals. Next, we introduce you to some of the digital conditioning devices used to prepare these digital signals for system transmission and reception.

In Unit 2, we discuss computers. We begin with a brief history of early versions of these powerful communication devices, and then delve into some of their secrets—the inner components and programs that make them work. A computer has to have some means of gathering input and delivering output to be a successful communications device. This assistance comes as attached peripheral devices (or somewhat detached in the case of resource sharing) and thereby, warrants coverage in our discussion.

Unit 3 brings to light that like us, all computers do not speak the same language and, therefore, find it difficult to communicate effectively. Thankfully, there are certain rules, called protocols, and overall standards (architectures) that govern how computers and communication systems must interact with one another to help curtail this problem. We cover these areas of interest, and then end our final volume by discussing some of the physical and logical devices used to connect computers together and control the tremendous amount of traffic we process.

This is the final volume of the first half (Part A) of your 5-level CDC. By the time you complete it, you should be prepared to move right into the second half (Part B) with no problems.

The glossary supplement to the CDC includes terms used in this volume. This allows you to simultaneously refer to terms used throughout the text and reduce the time you might spend looking for definitions of previously covered items.

Code numbers appearing on figures are for preparing agency identification only.

The inclusion of names of any specific commercial product, commodity, or service in this publication is for information purposes only and does not imply endorsement by the Air Force.

To get an *immediate response* to your questions concerning subject matter in this course, call the author at DSN 597-5161

E-mail [bearden@kee3b201.kee.aetc.af.mil](mailto:bearden@kee3b201.kee.aetc.af.mil) between 0700 and 1600 (CT), Monday through Friday. Otherwise, write the author at 335 TRS/TTMQEA, ATTN: MSgt Timothy L. Bearden, 600 Hangar Road, to point out technical errors you find in the

text, Unit Review Exercises, or Course Examination. Sending subject matter questions to ECI slows the response time.

**NOTE: Do not use the Suggestion Program to submit corrections for printing or typographical errors.**

Consult your education officer, training officer, or NCOIC if you have questions on course enrollment or administration, *Your Key to a Successful Course*, and irregularities (possible scoring errors, printing errors, etc.) on the Unit Review Exercises and Course Examination. Send questions these people can't answer to ECI, 50 South Turner Blvd, Maxwell AFB, Gunter Annex AL 36118-5643, on ECI Form 17, Student Request for Assistance.

This volume is valued at 24 hours (8 points).

**NOTE:**

In this volume, the subject matter is divided into self-contained units. A unit menu begins each unit, identifying the lesson headings and numbers. After reading the unit menu page and unit introduction, study the section, answer the self-test questions, and compare your answers with those given at the end of the unit. Then do the Unit Review Exercises (UREs).

**Acknowledgement**

PREPARATION of this volume was aided through the cooperation and courtesy of Siemens Transmission Systems, Inc., publisher; Electronic Industry Association, publisher; Datapro Research Corporation, publisher; Microsoft Corporation, publisher; and Petrocelli Books, Inc., publisher. These companies have furnished illustrations and technical information used throughout this volume.

Specifically, figures and technical data appearing in units 1 and 3 have been reproduced from *GTE Lenkurt Demodulator*, Jose C. De Leon, Editor ©1986; technical data and figures appearing in unit 1 have been reproduced from *EIA Standard RS-232-C Bulletin No. 9*, *EIA Standard RS-422-A*, *EIA Standard RS-423-A*, and *EIA Standard RS-449*, Electronic Industry Association; figures and technical data appearing in unit 1 have been reproduced from *Datapro Reports on Data Communications*, Volume 1, Datapro Research Corporation ©1987; technical data used in unit 2 has been reproduced from *Microsoft® MS-DOS Operating System version 5.0 User's Guide and Reference*, Microsoft Corporation ©1991; and figures appearing in unit 3 have been reproduced from *The Handbook of Data Communications and Computer Networks* by Dimitris N. Chorafas ©1985. Permission to use the information from these publications is gratefully acknowledged.

In accordance with the copyright agreement, distribution of this volume is limited to DOD personnel. The material covered by this permission *may not* be placed on sale by the government.

	<i>Page</i>
<b>Unit 1. Digital Data Signals.....</b>	<b>1-1</b>
1-1. Data Signal Characteristics .....	1-2
1-2. Electrical Interface Standards .....	1-15
1-3. Digital Signal Conditioning Devices .....	1-31
<b>Unit 2. Computer Systems and Equipment.....</b>	<b>2-1</b>
2-1. Computer Operation Principles .....	2-2
2-2. Types of storage and devices .....	2-16
2-3. Peripheral Devices .....	2-35
2-4. Operating Systems and Computer Memory.....	2-53
<b>Unit 3. Protocols, Standard Systems Architectures, and Digital Interface Devices .....</b>	<b>3-1</b>
3-1. Protocols .....	3-2
3-2. Standard Systems Architectures .....	3-28
3-3. Digital Interface Devices .....	3-43

**Please read the unit menu for Unit 1 and begin →**



## Unit 1. Digital Data Signals

	<i>Page</i>
<b>1-1. Data Signal Characteristics .....</b>	<b>1-2</b>
600. Characteristics and terms associated with digital data signals .....	1-2
601. Synchronization methods used in specific digital signals ....	1-11
<b>1-2. Electrical Interface Standards .....</b>	<b>1-15</b>
602. Industrial interface standards .....	1-16
603. Types of international interface recommendations .....	1-23
604. Military interfaces standards .....	1-25
<b>1-3. Digital Signal Conditioning Devices .....</b>	<b>1-31</b>
605. Purpose and operation of a converter .....	1-31
606. How encryption/masking devices are used .....	1-33
607. The purpose of coders/decoders (CODECs) .....	1-34

**D**ATA communication is a dynamic and rapidly expanding field, stimulated by the increasing need to link computers and other electronic machines across great distances. The resultant union of the data processor and the communicator has provided a vital service to the everyday operation of business, industry, and government.

Data systems and digital communicating techniques are the future of communications. The time is drawing near when virtually all information will be transmitted over long and short distances as high speed digital data signals. High-speed data transmission does not present a problem unless cost or media-type imposes restrictions.

Of course, we know cost and transmission restrictions do exist in military communications. On the other hand, the systems chosen for military use give good service when maintained and operated at peak performance levels. Knowing how to do this is fundamental to doing our job well. That is why we devote the content of this unit to digital data signals, methods, standards, and conditioning equipment.

Portions of this unit were developed using material from *GTE Lenkurt Demodulator* by permission of Siemens Transmission Systems, Inc. Permission to utilize this material is gratefully acknowledged.

## 1-1. Data Signal Characteristics

There are basic elements common to all data systems. In this section, we review the elements of data signals and synchronization techniques. We also examine electrical interface standards and digital data conditioning equipment.

### 600. Characteristics and terms associated with digital data signals

In the field of electrical communications, data is in two forms: analog data and digital data.

**Analog.** Analog data is processed information that is represented in a *continuous* form. One example of this form is the continuously variable electrical current produced by an electronic thermometer. Another, and perhaps better, example is the electrical signal generated by a telephone transmitter. The currents produced by the mouthpiece can have any value between defined limits and may be continuously changing in direct response to changing acoustical (sound) pressures. One reason the term "analog" applies is the fact that the currents vary in a way that resemble acoustical pressures. The currents are *analogous* in that they correspond to sound waves; otherwise, they are dissimilar. One way in which they do correspond is in the number of events per unit of time (frequency).

**Digital.** In contrast, digital data is processed information in *discrete* form. That is to say, the information consists of unconnected distinct parts. Because of this characteristic, the parts can be counted by a number system. This is why we use the term "digital."

Most information used by humans is in a digital form. The written language is digital, and human speech sounds are digital (discrete) in nature as well. Most of the instruments we have devised give us a digital readout, even though they may be measuring a continuously variable entity. An automobile speedometer, for example, is an analog device with a digital readout. A slide rule is an analog calculator, but its readout is digital. The current flowing through a meter movement is analog; again, the readout is digital.

Although speech is made up of individual sounds and pauses between sounds and groups of sounds, it is practical to classify it as a *nondiscrete* source of information, rather than a discrete source. Complexity is the reason. A speech wave is drawn from a continuous series of possibilities in which no part can be distinguished from other parts except by arbitrary division. As we speak, one sound flows into another. One word flows into the next. When pauses do come, they are often in the middle of a single word instead of between words. Moreover, information is carried in the inflections of your voice. Because of these things, the information value of a speech sound is hard to evaluate mathematically.

In telecommunications terminology, it is common practice to use the terms "digital" and "data" interchangeably, thus restricting data to mean digital information. We think of data as being information that can be processed or

produced by a computer, usually a digital computer. When we describe types of communications, a data system is any system of telecommunications that uses binary signals to provide information or to control a process.

In data systems the numbers, letters, and other symbols used to transfer information are represented by the sequential transmission of a specific number of electrical pulses. Each pulse is binary in nature; that is, each pulse can exist in either of *two* states. The most common terms for expressing the two states are "mark" or "space" and "1" or "0." To be sure, other terms *are* used when other characteristics of the electrical pulses are considered or when the pulses are changed into other forms.

**Digital signal formats.** In the Defense Communications System (DCS), the digital signal formats we're most likely to see are two-level nonreturn-to-zero (NRZ), three-level bipolar NRZ (NRZ AMI), and three-level bipolar return-to-zero, or RZ (RZ AMI). A digital signal varies only in predetermined, discrete steps, as opposed to an analog signal, which may vary continuously over its operating range. A digital signal moves from one discrete level to another. There may be two, three, or more discrete levels, but the signal moves instantaneously from one to another. For example, a two-level signal can assume either of two values, and a three-level digital signal can assume any one of three values. This is all the term "digital" means. It says nothing about speed, format, or level.

**Binary signal.** A binary digital signal is one that has two levels: +1 V and -1 V, +3 V and -3 V, +6 V and -6 V, etc. The term "binary" refers to two conditions. In other applications, binary could mean two frequencies, or just on and off; but with respect to digital signals, it means two levels. In this connection, we often refer to logic ones and logic zeros. In a binary numbering system, on which binary logic is based, there are only two numbers, one and zero. Hence, we have logic ones and logic zeros. In transmitting the signals in a binary system, we sometimes perform operations that cause the transmitted signal to have more than two levels, even though the information being transmitted is binary. A good example of this is bipolar RZ AMI signals. These signals, as transmitted, are three-level digital signals more correctly called ternary signals, even though the information transmitted is binary.

**Nonreturn-to-Zero (NRZ).** As applied to a digital signal, this means that when a signal is keyed to one level, it stays at that level until it's keyed to another level; i.e., it doesn't return to the zero level during the bit time. Certain modifying letters are sometimes used with this term.

a. NRZ-L (level). The L indicates that the information is contained in the level. One level is transmitted for a mark (or logic one), and another is transmitted for a space (or a logic zero). This is the most frequently used format in the DCS. When the abbreviation NRZ is used without a modifying letter, it is taken to mean NRZ-L.

b. NRZ-M (mark). The information is contained in the transition. There's a transition from one state to the other only when a mark bit is sent.

c. NRZ-S (space). The NRZ-S is just the reverse of NRZ-M.

**Return-to-Zero (RZ-S).** When a signal transition occurs, the signal level moves to one of the discrete signal levels, but after a predetermined time (normally 50 percent of a bit), it returns to the zero level.

**Alternate Mark Inversion (AMI).** In AMI, alternate marks (or logic ones) are inverted in polarity. If a logic one is transmitted, it is represented by a positive voltage if the previous logic one was negative. Conversely, it is represented by a negative voltage if the previous one was positive. Logic zeros are represented by zero voltage. This results in a three-level digital signal.

**Bipolar.** Bipolar is generally taken to mean a type of coding such as that described under AMI; that is, a ternary (three-level) signal in which logic zeros are represented by zero voltage, and logic ones by alternate positive and negative voltages. A bipolar signal may be either RZ or NRZ.

However, the term "bipolar" is used in connection with other types of signals, also. When this term or any other of these terms appear in a manufacturer's literature, it's a good idea to find out what the manufacturer really means.

A typical binary polar signal is most often an NRZ-L signal. Each mark (positive voltage) represents a one, and each space (negative voltage) represents a zero. In a bipolar NRZ AMI signal, the mark signal does not return to zero voltage without a space signal. The space signal stays at zero voltage until a mark bit occurs. In a bipolar RZ signal with 50 percent duty cycle, each mark bit only lasts half the time of a bipolar NRZ mark bit, then it Returns to Zero.

**Teletypewriter signaling.** The equipment used in the transfer of narrative information in digital data systems must use some keying or signaling method. These methods must be uniform between various types of equipment. In the Defense Communications System (DCS), we use two basic signal forms: neutral and polar operation.

**Neutral operation.** In neutral operation, you transmit coded signals by sending current over the line for marking pulses and by interrupting the current flow for spacing pulses. Neutral circuits use 60 mA or 20 mA for mark current and 0 mA for space current (fig. 1-1,a). A bias circuit must be used in neutral circuits to draw the armature from the mark contact to the space contact. The bias is normally set at half the current value of the mark current to overcome residual magnetism of the relay. Refer to figure 1-2.

The neutral system has the immediate advantage of simplicity. It requires very little terminal equipment; it is easy to operate; and, in general, it requires little attention. The chief disadvantages are from the transmission standpoint. Since the terminal impedance is different for the open and closed conditions, the transmitted wave for marks and spaces is not symmetrical. This usually results in distortion and limits the use of the neutral system. Therefore, you most likely see neutral keying used where the distance between the terminals is rather short, such as on in-station circuits, tributary station links, and local lines going to allocated

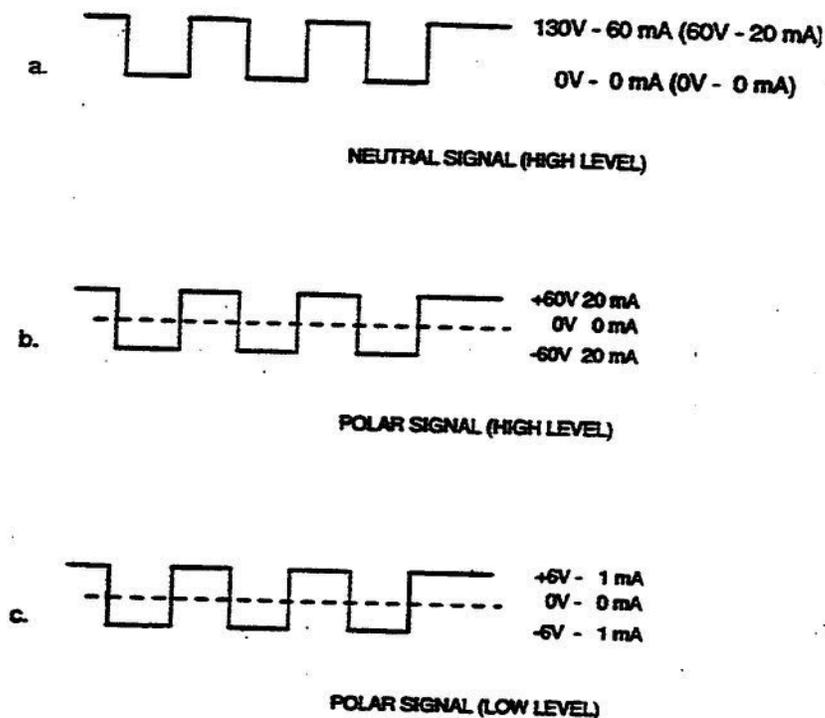


Figure 1-1. Signal formats.

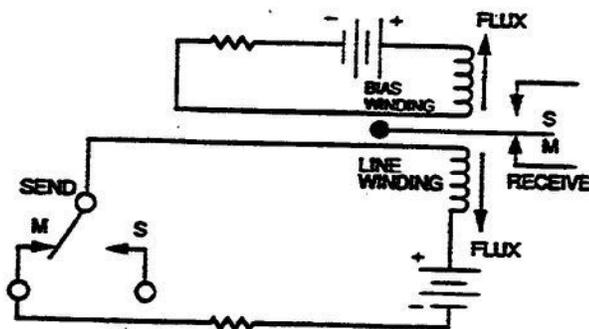


Figure 1-2. Simple neutral circuit.

channel customers. We should mention that neutral systems are being replaced by low-level polar systems, which are the military standard for teletypewriter operation.

**Polar operation.** Polar transmission is accomplished by changing the direction of current flow for mark and space pulses. In this system, the marking and spacing terms are retained, but they refer to the direction of current flow rather than to the current/no-current conditions found in neutral operation. In other words, the mark is produced by causing the current to flow in one direction, and the space is produced by causing the current to flow in the opposite direction.

The voltage for both is equal in magnitude, but of opposite polarity. See figure 1-1, b, and 1-1, c, for examples of polar signals. Low-level polar signals use  $\pm 6V$  1 mA; high-level polar signals use  $\pm 60V$  20 mA. The DCS objective is to eliminate all other types of signals, thus resulting in the total use of low-level polar signals. Figure 1-2 is an electrical circuit diagram of polar operation.

In figure 1-1, b and c, notice there are two different polar signals. One is high-level polar and the other is low-level polar. This difference in signal level (high and low) is brought about by engineering changes in equipment. This was done to lower the level of emanation (radiation) of signals being transmitted by high-level means. High-level signals are a threat to security, since they can be intercepted by sensitive receivers and converted back to readable text.

The *polar* principle of operation has an advantage over the *neutral* principle, because it is affected very little by normal variation in the DC voltage or by variations in line characteristics, whether the variations are caused by capacitance, inductance, or leakage. This is the principle that makes polar keying more desirable than neutral—especially where the line circuit is rather long. For example, a line that has capacitance in it causes the mark pulses of a neutral signal to become longer than normal. Since there is no current flowing, the space pulses (no-current condition) are unaffected. The result, then, is marking bias on the circuit.

Now, let's put polar keying on the same line. The capacitance still affects the circuit, but it has equal effect on both the marks and spaces, so the result is a canceling of the effect on the circuit as a whole.

Another advantage of polar signaling over neutral is in handling the problem of foreign battery on a line. In the neutral circuit, foreign battery appears on the space pulses, which should normally show an open. If the voltage of the foreign battery is high enough to overcome the bias voltage in the neutral line relay, false keying results and you have garbled traffic. A low threshold of foreign battery causes bias on the circuit—either marking or spacing—depending on its polarity. Again, polar signaling is unaffected by this circuit condition. Since the polar signal is keyed with opposite polarity for marks and spaces, the presence of foreign battery merely changes the reference point for marks and spaces. In theory, the presence of a foreign battery charge up to the line battery voltage can be tolerated on a polar circuit without noticeable signal degradation. Also, polar keying reduces the effects of reflection due to mismatch in the lines. Neutral keying, since it operates on the principle of current on and current off, imposes infinite line impedance during space pulses and nominal line impedance for marks.

To the system controller, the greatest advantage is that polar keying gives much less distortion than neutral keying on the same line. Figure 1-3 illustrates a typical polar DC teletype circuit. Notice that the send side of this circuit has a neutral relay; that is, mark pulses cause current to flow through the primary of the relay, pulling the armature to the mark contact. This closes the circuit in one direction through the polar relay, causing current to flow and draw the armature of the

receive relay to the mark contact. A space pulse on the send relay allows the bias winding to draw the send armature to the space contact. This closes the circuit through the receive relay in the opposite direction, thereby causing the receive armature to be drawn to the space contact.

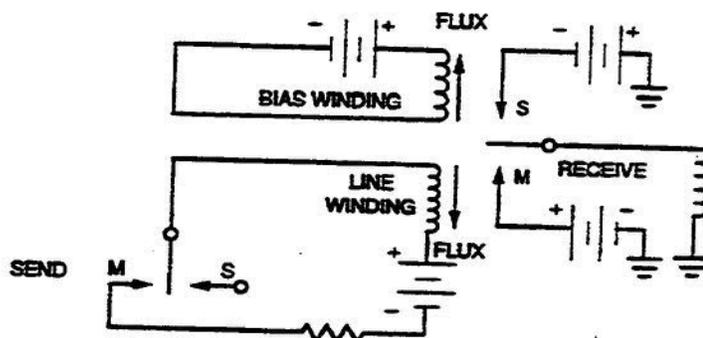


Figure 1-3. Simple polar circuit.

An additional advantage of the polar method of keying over the neutral method is that it gives the system's controller more clues as to what might be wrong. In the case of a neutral system, the controller, seeing an open, does not know if the equipment at the distant end of the line has failed. The polar method always transmits some sort of battery that the controller can use to help isolate the probable trouble in the line or equipment. A steady space battery indicates that there is a stuck relay or that something is wrong with the sending device, whereas the same thing over a neutral circuit could indicate a bad line, bad equipment, or power failure. In other words, the whole book of possible troubles is open to the controller.

**Digital data terminology.** To make sure you understand the terms used in this text, let us firmly establish their meanings before proceeding further. As you know, a digital signal has two states. These are 0 or 1, mark or space, current or no current, holes or no holes (as in paper tape), and many more combinations that indicate either of two contradictory conditions.

**Bit.** In digital signals, each condition, state, or pulse is called a binary digit. This term has been shortened to "bit." A bit is a unit of information based on two symbols, states, or conditions. Remember, in technical school, bit was stated as being the smallest amount of useful information. This is true: 1 bit of information is the amount of information needed for a receiver (person or machine) to make a correct choice or decision regarding two possible messages (fig. 1-4). The more bits that are added to a digital signal, the greater the number of choices to be made. If these bits are arranged in a preset sequence that is understood by the sender and receiver, a code set is established. With a code set, it is possible to transmit complex messages that are limited only by the number of bits used. In figure 1-4, you can see that 1 bit can indicate only two possible messages. When a second bit is added, the number of possible messages increases to four. A further increase to 3 bits increases the possibilities to eight different choices. Each time a

bit is added to the code set, the number of choices is increased by adding 1 to the exponent of the base 2. For example, a code with 7 bits ( $2^7$ ) can represent 128 possible messages or symbols.

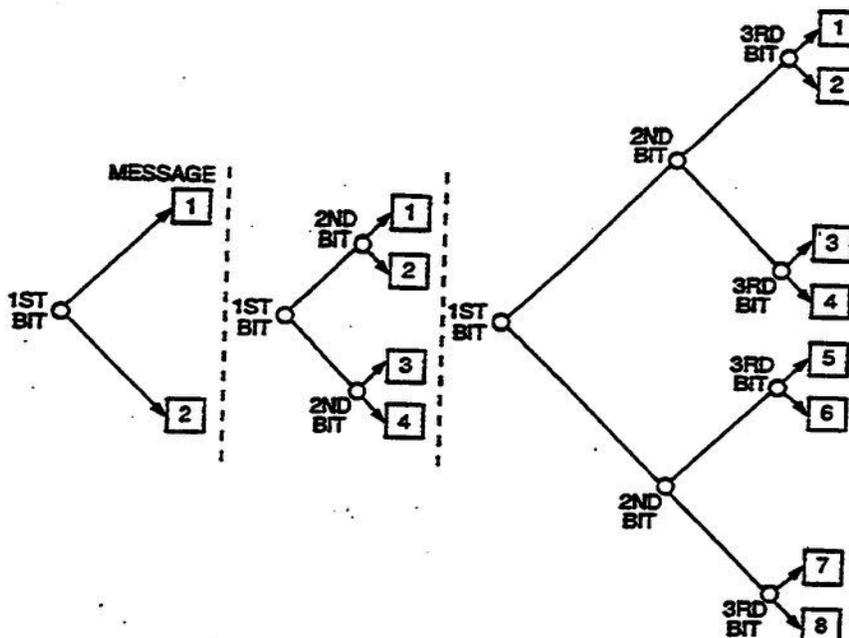


Figure 1-4. Amount of information required to specify a message.

When we express the term "bit" with regard to time, we have bits per second (b/s). This simply means that we can move a certain number of symbols, states, or conditions in 1 second. This is the method commonly used in data systems to express signaling speed for equipment.

**Signaling rate.** There are several methods used to refer to modulation rate or signaling speeds. As stated in the preceding paragraphs, signaling speed is expressed by the amount of information that can be transferred in a given time. For example, if we say 1,200 bits per second, we are stating that we can transfer 1,200 bits in 1 second. What we are not stipulating is whether all the bits are equal in length.

Another method of referring to signaling speed or modulation rate is through the use of the term "baud." Let's first examine the definition of baud and then make a comparison of bits per second and baud as expressions of transfer rate or speed.

**Baud.** The word "baud," by definition, is the unit of modulation rate. Modulation rate expressed in baud is the reciprocal of time of the shortest unit interval in seconds. The unit interval is the bit in a digital signal that takes the shortest amount of time. Baud, therefore, expresses the maximum bits transferred if all bits are equal in length and a 1-to-1 (bit-to-modulation) encoding technique is used.

The example in figure 1-5, A, shows that all bits are of the same length. In this case, the unit interval is 2 milliseconds (ms). As stated above, baud is the reciprocal of time, or written in mathematical form:

$$\begin{aligned} \text{bauds} &= \frac{1 \text{ second}}{\text{shortest unit interval}} \\ &= \frac{1 \text{ second}}{0.002 \text{ second}} \\ &= 500 \text{ bauds} \end{aligned}$$

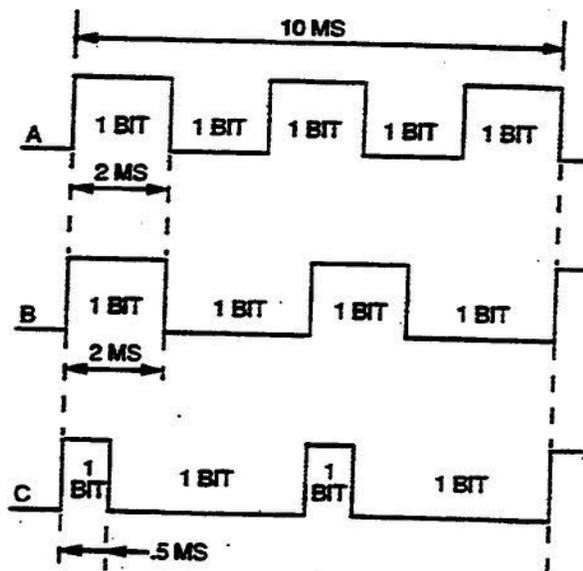


Figure 1-5. Bits versus baud.

**Baud versus bit.** We have talked about baud and bit. The terms “unit interval,” “baud,” “bit,” and “modulation rate” are vital tools to the communicator. The systems controller must reach a thorough understanding of these terms to ensure continuous quality communications to customers.

Trouble arises when the terms “baud rate” and “bit rate” are used interchangeably. Baud is an expression of time of the shortest unit duration. Bit, on the other hand, carries no suggestion of time; it expresses the number of information pulses in 1 second, but it tells nothing about their duration. Modulation rate is a measure of speed, the speed with which the available bandwidth can be keyed. The modulation rate is expressed in baud because it indicates the duration of the unit interval and gives us an accurate indication of transmission rate without regard to any grouping of the bits.

As a result of the influence of the computer and data processing upon our language, modulation rate is sometimes expressed as “bits per second” (b/s). The term “bit rate” carries no implication of a “unit bit” with which to assemble characters and position bits of any desired length. Refer to figure 1-5. You can see that signal “A” has 5 bits, each identical in length. Each bit is 2 milliseconds in length, making the unit interval the same. The total time involved for all three

signals is 10 milliseconds. If we figure the modulation rate for figure 1-5, A, we see that it is 500 bauds.

$$\frac{1 \text{ second}}{0.002 \text{ second}} = 500 \text{ bauds}$$

At the same time, we can figure the maximum number of bits transferred in a period of 1 second.

Since there are 1,000 milliseconds in a second and 5 bits are transferred each 10 milliseconds, the signaling speed is 500 bits per second. Here the numerical identification is the same. However, if we figure the modulation rate and signaling speed for the signal shown in figure 1-5, B, we find them to be 500 bauds and 400 bits per second. In figure 1-5, C, we find them to be 2,000 bauds and 400 bits per second. From this example, you should see it is technically inaccurate to use the two terms interchangeably.

The modulation rate of a channel is the speed with which it transmits pulses, assuming all pulse intervals contain pulses. It is not a direct measure of the information capacity of the channel, except for binary signals. Since each bit in a binary signal occupies one signaling element, the use of the terms "baud" and "bits per second" is only interchangeable when bits per second equals baud for a binary signal.

Other ways you may hear signaling speed expressed are "words per minute" and "operations per minute." Actually, these two terms are expressions of symbol transfer rate used with teletypewriter operation. Neither term has any technical accuracy. They are used in communications with non-technical persons who do not understand the terms "bits per second" or "baud." Words per minute are the actual number of five-character words that are transferred between sending and receiving devices in a 1-minute period. Operations per minute describes the number of machine functions a send or receive device can handle in a 1-minute period.

**Bit count integrity.** Bit count integrity is nothing more than counting the number of bits received on a circuit or system. Naturally, if the transmitting station is sending 9600 b/s, it is not unreasonable to expect to receive 9600 b/s. However, there are times when the number of bits received varies from what is transmitted. Let's say, for example, we receive 9605 b/s. We would need a buffer to store the extra bits. This is called an *overflow* and, depending on the bit count specifications for the circuit or system, it could cause an alarm to be activated. Less than the expected bit per second rate is called an *underflow* and could also cause an alarm. Either way, you have a problem.

Bit count integrity problems are primarily caused by system delay or timing faults. Remember, the bit count integrity is only concerned with the number of bits received in a specific time and not with errored bits.

## 601. Synchronization methods used in specific digital signals

Digital devices fall into three categories: synchronous, asynchronous, and isochronous. These terms characterize the methods used to synchronize the send and receive devices.

**Synchronous.** Synchronous operation requires no synchronizing pulses in the signal stream, and all bits are the same length in time. In synchronous operation, the receiving device is adjusted automatically to the speed of the transmitting device by comparing the speed of the incoming signal with the time base of the receiving device. The time base can be an externally supplied timing or clock signal. The equipment using this method is very stable and requires a very small amount of correction once a proper phase relationship between transmit and receive devices is achieved.

In maintaining synchronization, the correction portion of the receive device acts upon the received information to correct the local timing. In one method of doing this, the receiving device senses all mark-to-space transitions and determines the average position in time of its occurrence. In a perfect signal, the sampling point is at the center of a signal bit (there is a similar method of operation that uses the space-to-mark transitions as the reference point). The second method requires the receiving device to sense both space-to-mark and mark-to-space transitions and determine the average position in time of their occurrence. The sampling interval is positioned at a point midway between the average occurrence of transmission or in the center of the signal bit. The receiving device uses this sample of the incoming signal for translation into a useful form of intelligence; that is, reproduction of what was transmitted. Figure 1-6 shows examples of incoming signals and sample points with no distortion of the incoming signal. Notice that all bits in the synchronous data signal are of equal duration, unlike that of the start-stop signal, where the stop element was longer in time than the other elements.

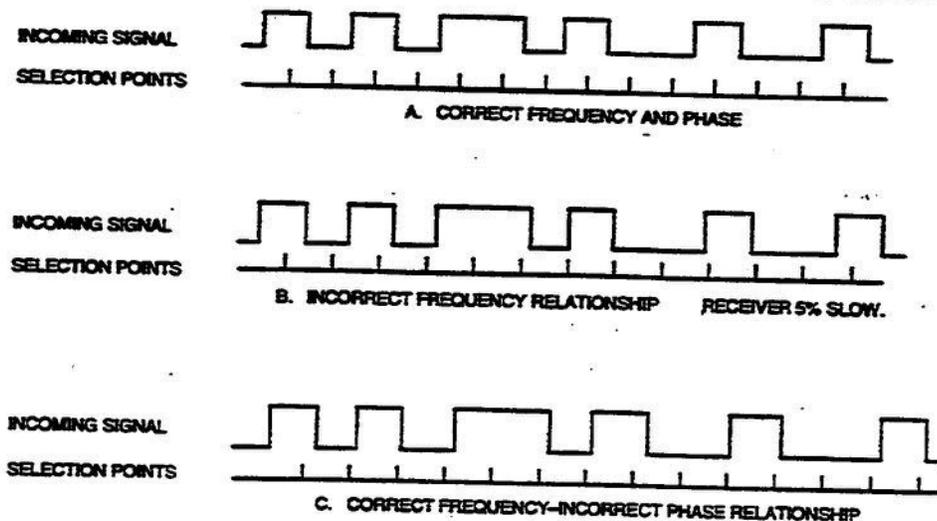


Figure 1-6. Relationship between incoming signals and receiver timing.

**Asynchronous.** An asynchronous signal is any signal that contains synchronizing bits within its signal stream. This method of synchronization was at one time the most common form. It is identified by a start pulse or bit at the beginning and a stop pulse or bit at the end of each character. The start pulse is recognized by always being a space and equal in unit interval to the information bits. The stop pulse, on the other hand, is always a mark and may have the same or longer unit interval than the information bits (fig. 1-7).

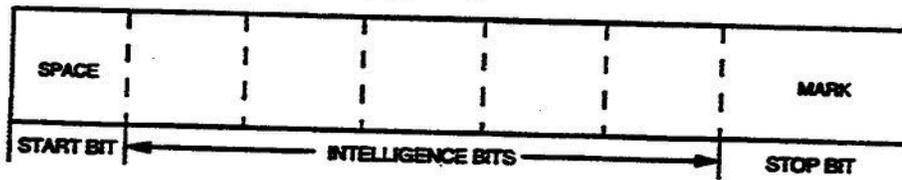


Figure 1-7. Asynchronous bit stream.

These pulses are used to synchronize the receiving device to the transmitting device. Synchronization is needed due to the motor speed differences between different pieces of equipment. The slightest amount of speed variation causes errors in the received character. This start-stop operation is used exclusively in teletypewriter operation. In this mode, the receiving machine is allowed to run only one character and is then stopped to await the reception of the next start signal, indicating that the next character is about to start. In this way, the speed difference between transmitting and receiving machines can accumulate only during the length of one character.

There is a disadvantage to this method. The length of each character must be increased to include 1 bit to start the receive device and a longer bit to stop it; also, the receive device must be slightly faster than the transmitter so that it completes its cycle of receiving and decoding the received character and is ready for the next character before the transmitter sends it. As you can see, speed of transmission and operating margins are sacrificed for synchronization.

**Isochronous.** An isochronous signal is defined as a signal that has all bits of equal duration. The term is basically used in channel-packing systems to refer to a method of synchronization that uses its own internal timing. For simplicity, isochronous signals are defined as signals that have equal unit intervals. Therefore, if we have an asynchronous signal (start-stop) that uses a stop element equal in duration to the intelligence bits, it is also an isochronous signal. All synchronous signals are isochronous, since they fit the preceding description.

## Self-Test Questions

After you complete these questions, you may check your answers at the end of the unit.

### 600. Characteristics and terms associated with digital data signals

1. Which form of data, analog or digital, is processed in a continuous form?
2. Which form of data is processed in a discrete form?
3. What does the term *binary* refer to when it is associated with digital signals?
4. What is a ternary digital signal?
5. Describe a nonreturn-to-zero (NRZ) signal.
6. Which part of a NRZ-L signal contains information?
7. Where is the information contained in a NRZ-M signal?
8. If the last "mark" transmitted in an alternate-mark-inversion (AMI) signal was negative, what will be the polarity of the next mark?
9. Describe a bipolar signal.
10. What represents mark and space pulses when you use neutral teletypewriter keying?

11. What represents mark and space pulses when you use polar teletypewriter keying?
12. What principle of polar keying makes it more desirable than neutral keying?
13. To a systems controller, what is the best advantage of using polar keying instead of neutral keying?
14. Define the term, "bit."
15. What unit of measurement is normally used in data systems to express signaling speed for equipment?
16. Define the term, "baud."
17. Briefly explain the difference between baud and bit.
18. What is the difference between "overflow" and "underflow" in relation to bit count integrity?

**601. Synchronization methods used in specific digital signals**

1. What are the three terms used to describe synchronization methods used in digital signaling?
2. How many synchronizing pulses are needed in synchronous operation?
3. Briefly explain the principle of synchronous operation.

4. What constitutes an asynchronous signal?
5. Briefly explain the principles of asynchronous operation.
6. How does asynchronous synchronization affect transmission speed?
7. What is an isochronous signal?
8. Of the synchronization terms synchronous, asynchronous, or isochronous, which is normally used with channel-packing systems to indicate the utilization of internal timing?

## 1-2. Electrical Interface Standards

Since data communications was first thought of as a means for transferring information from one point to another, several methods for interfacing the data terminal equipment (DTE) to the data circuit-terminating equipment (DCE) have been invented. As each new method was developed, a new "standard" to apply to the new data network configuration was established. The standards specify what type of connectors each piece of equipment uses, the number of pins on the connectors, and the electrical characteristics of the pins, as well as the use of each pin. As different companies applied different standards to the equipment they manufactured, it became necessary for the buyers of the new equipment to consider what standard was used in its development. They had to ensure that the new equipment was compatible to their existing system.

As the Worldwide Military Command and Control System (WWMCCS) and Local Area Network (LAN) came into being, the engineers considered which standard they wanted to use. Since the existing equipment on the market was already set to certain standards, they were forced to use the standard that was available. As the world became more concerned with survivability, new standards were developed to satisfy this need. Some of these new standards were adopted into the WWMCCS and LAN world and some were not. As major revisions are instituted within the work centers, equipment within the center is upgraded to the new standards to provide this "survivability" that some of the existing systems

lack. Since you might work with some of this equipment, you should be familiar with these standards.

The Electronic Industry Association (EIA) writes standards for all industries utilizing electronic equipment. This is the standard used by most of our commercial counterparts. In this section, we discuss some of the more common standards in use today.

Permission to use material in the following lesson was granted by Electronic Industry Association.

Some material used in lessons 602-604 was taken from *Datapro Reports on Data Communications*; permission to use this material was granted by Datapro Research Corporation and is gratefully acknowledged.

## 602. Industrial interface standards

The EIA developed four electrical standards used by manufacturers to interface data terminal and data circuit terminating electronic equipment: (1) EIA RS-232-C, (2) EIA RS-422-A, (3) EIA RS-423-A, and (4) EIA RS-449.

**EIA RS-232-C.** The EIA RS-232-C is probably one of the oldest standards and the most widely used DTE-to-DCE interfaces in the WWMCCS network. It applies to all classes of data service including the following:

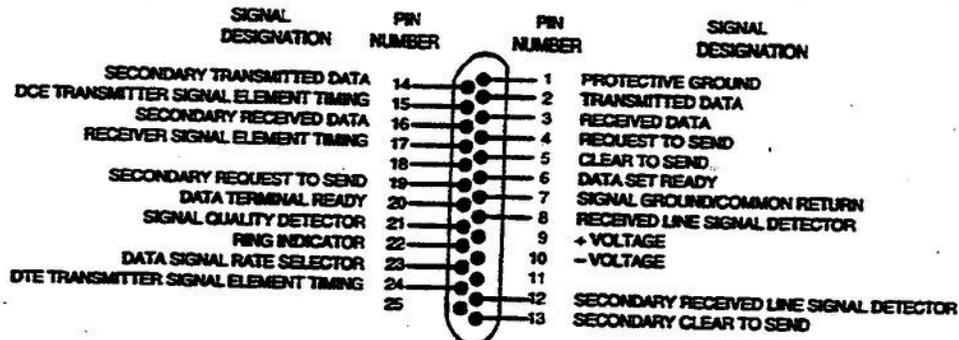
- a. Dedicated leased or private line service, either two-wire or four-wire. Consideration is given to both point-to-point and multi-point operation.
- b. Switched network service, either two-wire or four-wire. Consideration is given to automatic answering of calls; however, this standard does not include all of the interchange circuits required for automatically originating a connection.

This standard applies to both asynchronous and synchronous data transmissions that support speeds up to 20 kb/s in the full- or half-duplex mode. RS-232-C is a single-ended or unbalanced interface, where all of the interchange signals share a common return ground (signal ground) that can be interconnected at the interface point.

The RS-232-C standard meets three distinct specifications: (1) type of connector, (2) purpose and definition of connector pins, and (3) electrical characteristics of those pins.

**Type of connector.** RS-232-C uses a plug-in, 25-pin connector. The connector is keyed with 13 pins on the top row and 12 pins on the bottom row to prevent improper connection. The male connector is used on the cable coming from the DTE, while the female connector is used on the cable coming from the DCE. Normally, cable between the two is no longer than 50 feet or 15 meters, but may be longer if load capacitance requirements are met. Use of longer cables is restricted by the data rate and by environmental conditions.

**Purpose and definition of pins.** The pins on the connector contain "interchange circuits," which are simple paths between the DTE and the DCE. Twenty-three of the circuit leads have a function; the remaining two are unassigned (fig. 1-8). The purpose of the various pins can be easily understood by knowing how they are grouped. The categories are data, clock, control, and common or ground leads.



Pin Number	Function	Circuit	Data Signal from		Control Signal from		Timing Signal from		CCITT Equivalent*
			DCE	DTE	DCE	DTE	DCE	DTE	
1	Protective Ground								
2	Transmitted data	AA		X					101
3	Received data	BA	X						103
4	Request to send	BB				X			104
5	Clear to send	CA					X		105
6	Data set ready	CB			X				106
7	Signal ground/common return	CC			X				107
8	Received line signal detector	AB				X			102
9	Reserved for data set testing	CF							109
10	Reserved for data set testing								
11	Unassigned								
12	Secondary received line signal detector	SCF				X			122
13	Secondary clear to send	SCB			X				121
14	Secondary transmitted data	SBA		X					118
15	Transmission signal element timing (DCE)	DB					X		114
16	Secondary received data	SBB	X					X	119
17	Receiver signal element timing (DCE)	DD					X		115
18	Unassigned								
19	Secondary request to send								
20	Data terminal ready	SCA					X		120
21	Signal quality detector	CD				X			108.2
22	Ring indicator	CG			X				110
23	Data signal rate selector	CE			X				125
24	Transmit signal element timing (DTE)	CH/CI			X	X			111/112
25	Secondary clear to send	DA						X	113
		SCB			X				

Figure 1-8. RS-232-C pin assignments.

**Data leads.** Data leads are those pins that carry the data intelligence from the DTE to the DCE and vice versa. They are divided into two categories: primary channel and secondary channel.

The primary channel is the data channel with the highest signaling rate where two channels share a common interface connector. It is comprised of pin 2 (transmit data) and pin 3 (receive data). The secondary channel has the lower signaling rate of the two channels. It is comprised of pin 14 (secondary transmit data) and pin 16 (secondary receive data).

The primary and secondary channels may be independent of other interchange circuits in terms of direction and speed. Basically, secondary channels function the same as primary channels. When not in use, both primary and secondary channels are held in a binary "1" condition (mark).

*Clock leads.* Clock leads provide "clock" or "timing" for the data transmission synchronization. On most circuits, the modem provides this timing to the DCE on pins 15 (transmission signal element timing) and 17 (receiver signal element timing). Occasionally, on a "direct-connect" without a modem, the DTE provides this timing on pin 24 (transmit signal element timing).

Clock, or timing, is very important when we transmit synchronous data signals. An on-to-off transition of the timing indicates the occurrence of the center of the data element. It is during this transition that the signal is sampled to confirm if it is a mark or a space. An off-to-on transition of the timing indicates the occurrence of a transition in the data signal. This timing/data relationship exists to make sure that a sample is not taken during or near the data transition, where distortion may alter the signal and give a faulty indication of the data status (mark/space transition).

To maintain this relationship on a synchronous system, the timing source must be twice the speed of the data rate. When we use an asynchronous signal, it is not necessary that these timing leads be connected.

*Control leads.* Control signals are used to enable and disable data transmission and reception; they are also used to indicate the operational status and condition of the DTE and DCE. These control circuits include the following:

- a. Request to Send (RTS) – Pin 4. This circuit is used to condition the local data communications equipment for data transmission and, on a half-duplex channel, to control the direction of data transmission of the local data communications equipment.
- b. Clear to Send (CTS) – Pin 5. Signals on this circuit are generated by the data communications equipment to indicate whether or not it is ready to transmit data.
- c. Data Set Ready (DSR) – Pin 6. Signals on this circuit are used to indicate the status of the local DCE.
- d. Data Terminal Ready (DTR) – Pin 20. Signals on this circuit are used to control switching of the data communications equipment to the communications channel.
- e. Received Line Signal Detector (Carrier Detect - CD) – Pin 8. This circuit is used to indicate when the data communications equipment is receiving a signal that meets its suitability criteria for demodulation. These criteria are established by the data communications equipment manufacturer.

*Common or ground leads.* This conductor establishes the common ground reference potential for all interchange circuits, except for the protective ground circuit. There are two common or ground leads used in the equipment.

1. **Protective Ground – Pin 1.** This ground connection keeps you from getting shocked when you touch the equipment. It is electrically bonded (fastened) to the equipment frame and should be connected to an external ground such as the ground wire of an electrical plug. This wire strap can be connected or removed at installation, as required to meet applicable regulations or to minimize the introduction of noise into electronic circuitry.
2. **Common (Signal) Ground – Pin 7.** This ground establishes a common reference for all interchange circuits except pin 1. This should always be passed directly through an interface and terminated only at the ends of the DC circuit. Modems have this capability through the use of an internal ground strap option.

Within the data communications equipment, this circuit is brought to one point. It is possible to connect this point to the protective ground circuit by means of a wire strap inside the equipment.

**Electrical characteristics of pins.** The EIA RS-232-C standard prescribes bipolar-voltage serial data transmission within 50 feet of the DCE connection. Transmit data is represented by the mark for binary (1) condition and by a space for the binary (0) condition. A data signal is in the mark condition when the voltage at the interface point is more negative than minus (-) 3 volts with respect to signal ground. When the signal is more positive than positive (+) 3 volts, it is in the space condition. (This is completely opposite of what is mentioned in MIL-STD-188-C, where +6 would be a mark and -6 would be a space.) The area between -3 and +3 volts is designated as the "transition" region, and the signal state is not defined. RS-232-C allows 3 to 15 volts on the control leads and 3 to 12 volts on the clock leads, but both normally use 6 to 12 volts. Normally, we standardize our systems by using +6 and -6 volts for the data, control, and clock leads.

All interchange circuits utilize a DC square wave for information interchange, and ground potential must always maintain 0 volts potential on both ground pins to ensure signals appearing on the data, control, and clock leads are not distorted.

**EIA RS-422-A.** The RS-422-A is the first in a series of new standards designed to replace the RS-232-C. It specifies a balanced electrical interface that operates at a much higher speed and over longer distances than its predecessor. Since it specifies a balanced interface, we need to have a balanced circuit in which the positive and negative signal lines are isolated from ground. Each signal lead has its own common ground return lead, which makes it much less susceptible to noise. With the reduction of noise in this standard, we have the advantage of higher data rates and longer cable runs.

The RS-422-A is designed for application on a twisted-pair telephone wire at distances up to 4,000 feet and data using rates up to 100 kb/s. The data rate can increase up to 10 Mb/s at distances of 40 feet or less. This makes it possible to interconnect equipment within a facility without the need for modems. The RS-422-A standard is also full-duplex, and as a result of the interface equipment used,

it requires no control leads. This standard was specifically designed for point-to-point configurations, but can be modified for other purposes.

The RS-422-A circuit includes a generator (source device), a receiver (sink device), and an interconnecting cable. The generator produces a balanced voltage source (one source for positive voltage and another source for negative voltage) on two wires in the range of 2 to 6 volts. The receiver is a differential balanced receiver with an input impedance greater than 4,000 ohms and a sensitivity of 200 mV. The standard specifies the conditions under which cable terminations and fail-safe circuits must be used. Although it does not specify the wire size to be used in the cable, distance figures are based upon the use of overall shielded 24 gauge, multiple twisted-pair cables.

Some typical situations where the balanced interface might be used are as follows:

- a. The interconnecting cable is too long for effective unbalanced operation.
- b. The interconnecting cable is exposed to extraneous noise sources that may cause an unwanted voltage in excess of  $\pm 1$  volt measured between the signal conductor lead and its accompanying common return at the end of the cable with a 50-ohm resistor substituted for the generator.
- c. It is necessary to minimize interference with other signals.
- d. Inversion of signals is required. Suppose you need a positive mark instead of a negative mark; you can accommodate this by simply inverting the leads instead of changing your battery source.

While a restriction on maximum cable length is not specified, guidelines are given with respect to conservative operating distances as a function of the data signaling rate. In general, these conservative values may be greatly exceeded where the installation is engineered to ensure that noise and ground potential values are held within specified limits.

**EIA RS-423-A.** This standard specifies the electrical characteristics of the unbalanced digital interface. The complete unbalanced digital interface circuit consists of a generator, an interconnecting cable, and a receiver (or load).

RS-423-A specifies that the generator be a low-impedance (5 ohms) source producing an unbalanced voltage that is applied to the cable in the range of 4 to 6 volts. The receiver is a differential receiver with an input impedance greater than 4,000 ohms and a sensitivity of 200 mV. The cable should be 24 gauge or larger and may be twisted pair or flat cable.

The provisions of the standard may be applied to the circuits employed at the interface between equipment where the information being conveyed is in the form of binary signals at the DC baseband level. The unbalanced digital interface circuit is normally utilized on data, timing, or control circuits where the data signaling rate on these circuits is up to 100 kb/s. Unbalanced digital interface devices meeting the electrical characteristics of this standard need not operate over the entire data-signaling-rate range specified. They may be designed to

operate over narrower ranges to satisfy more economically specific applications, particularly at the lower data signaling rates.

While the unbalanced interface is intended for use at lower data signaling rates than the balanced interface, its general use is not recommended where the following conditions prevail:

- a. The interconnecting cable is too long for effective unbalanced operation.
- b. The interconnecting cable is exposed to extraneous noise sources that may cause an unwanted voltage in excess of  $\pm 1$  volt measured differentially between the signal conductor and circuit common at the load end of the cable with a 50-ohm resistor substituted for the generator.
- c. It is necessary to minimize interference with other signals.

One important thing to remember about this standard is that it is similar to the RS-232-C in many respects and, as such, can be used for interoperation under certain conditions with generators and receivers of the RS-232-C standard.

**EIA RS-449.** The RS-449 standard is intended eventually to replace the familiar RS-232-C standard. RS-449 and its companion standards were developed to permit an orderly transition from existing equipment using RS-232-C to a newer generation of equipment using RS-449 without forcing obsolescence or costly retrofits. However, RS-232-C remains the most common interface because most United States manufacturers of data communications equipment have embraced the older standard.

RS-449 operates in conjunction with either of two standards specifying electrical characteristics: RS-422-A, for balanced circuits; and RS-423-A, for unbalanced circuits. It governs the mechanical and electrical characteristics of the interface between the DTE and the DCE. The RS-449 standard applies to binary, serial, synchronous, or asynchronous communications. Half- and full-duplex modes are accommodated, and transmission can be point-to-point or multi-point over two- or four-wire facilities. Point-to-point arrangements can be either switched or dedicated. Multi-point arrangements are connected by dedicated lines. Normally the RS-449 places the transmission speed and distance limitations between the terminal and the network at 2 Mb/s and 60 meters.

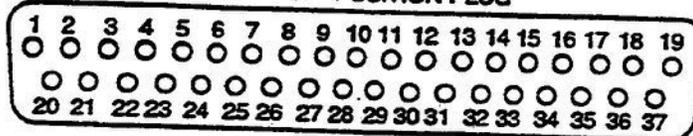
Most of the interchange functional definitions given in RS-232-C have been retained in RS-449; some significant differences are as follows:

- a. Data rates are accommodated up to 2 Mb/s when using the balanced interface specified in RS-422-A.
- b. RS-449 specifies a different interface connector size and latching arrangement. A 37-pin connector is used to accommodate additional interface leads that support newly defined functions and, also, to accommodate balanced operation for 10 interchange circuits when it is used. In addition, a separate 9-pin connector is specified to serve secondary channel interchange circuits, when applicable.

- c. Ten additional circuit functions are defined including: three circuits for control and status of test functions associated with DCE; two circuits for control and status of a transfer function of the DCE to a standby channel; one circuit to provide an out-of-service function under control of the DTE; one circuit to provide a New Signal function; one circuit to provide DCE frequency selection; and two circuits to provide a common reference for each direction of transmission across the interface.
- d. Three interchange circuits defined in RS-232-C have not been included in RS-449. Pins 9 and 10 of RS-232-C are reserved for data set testing; these have been excluded in RS-449 to minimize interface connector size. Protective ground has also been excluded so bonding of equipment frames can be implemented in accordance with national and local electrical codes.
- e. The option in RS-232-C that permits the omission of the Request To Send interchange circuit for certain transmit-only or full-duplex applications is excluded in RS-449.
- f. The definition of the Data Set Ready function in RS-232-C has been changed in RS-449, and a new function, Data Mode, which indicates further DCE status conditions, has been added.
- g. To avoid confusion with RS-232-C, RS-449 has established its own set of circuit names and mnemonics.

For a breakdown of the pin assignments and the mnemonics used with this 37-pin plug, refer to figure 1-9.

## 37 - POSITION PLUG



Pin	Circuit Mnemonic	Circuit Name	Circuit Direction	Circuit Type
19	SG	Signal Ground	—	Common
37	SC	Send Common	To DCE	
20	RC	Receive Common	From DCE	
28	IS	Terminal In Service	To DCE	Control
15	IC	Incoming Call	From DCE	
12.30	TR	Terminal Ready	To DCE	
11.29	DM	Data Mode	From DCE	
4.22	SD	Send Data	To DCE	Data
6.24	RD	Receive Data	From DCE	
17.35	TT	Terminal Timing	To DCE	Timing
5.23	ST	Send Timing	From DCE	
8.26	RT	Receive Timing	From DCE	
7.25	RS	Request to Send	To DCE	Control
9.27	CS	Clear to Send	From DCE	
13.31	RR	Receiver Ready	From DCE	
33	SQ	Signal Quality	From DCE	
34	NS	New Signal	To DCE	
16	SF SR	Select Frequency Signaling Rate Selector	To DCE To DCE	
2	SI	Signaling Rate Indicator	From DCE	Control
10	LL	Local Loopback	To DCE	
14	RL	Remote Loopback	To DCE	
18	TM	Test Mode	From DCE	
32	SS	Select Standby	To DCE	Control
36	SB	Standby Indicator	From DCE	
3.21	—	Spare	—	—
1	—	Shield	—	—

Figure 1-9. RS-449 pin assignments. (Reproduced with permission of the Electronic Industry Association, ©1977.)

### 603. Types of international interface recommendations

Another agency that deals with electronic interface is the International Telegraph and Telephone Consultative Committee (CCITT). This is the official body where telecommunications carriers decide how they work across international borders. It is part of the International Telecommunications Union (ITU) in Geneva that is an organization associated with the United Nations. CCITT publishes recommendations that guide all the communications carriers. However these recommendations are not always used as a worldwide standard because in many

cases, several different recommendations are adopted to suit different countries or groups of countries. Here, we discuss very briefly two recommendations: the X- and V-series.

**X-series.** The X-series recommendations relate to data transmission. The first of these recommendations defines the international classes of service in terms of link speeds and their usage (asynchronous or synchronous). It also defines the facilities to be offered by the various services. The main ones you may see are the X.20, X.21, and X.25. They are listed and defined below.

**X.20.** Interface between data terminal equipment (DTE) and data circuit-terminating equipment (DCE) for start/stop transmission services on public data networks.

**X.21.** General-purpose interface between data terminal equipment (DTE) and data circuit-terminating equipment (DCE) for synchronous operation on public data networks.

**X.25.** Interface between data terminal equipment (DTE) and data circuit-terminating equipment (DCE) for terminals operating in the packet mode on public data networks.

**V-series.** The CCITT V-series is a code designation used to list the recommendations for data transmission using the telephone network. Many of the recommendation designators deal with modems and data speeds. Some of the more common ones you might find are the following:

**V.21.** 200-b/s modem standardized for use in the general switched telephone network.

**V.23.** 600/1.2 kb/s modem standardized for use in the general switched telephone network.

**V.26.** 2.4 k/1.2 kb/s modem standardized for use on four-wire leased circuits.

**V.27.** 4.8 kb/s modem standardized for use on leased circuits.

**V.29.** 9.6 kb/s modem for use on leased circuits.

**V.35.** Data transmission at 48 kb/s using 60-to-108 kilohertz (kHz) group b/s circuits.

**V.36.** Modems for synchronous data transmission using 60-to-108 kHz group b/s circuits.

As you can see from the many recommendations and code designations mentioned, as new equipment is developed, new standards are developed to keep pace and follow their advancements. There are many that are used by our commercial counterparts. Those listed above are just some of the more common ones you might encounter. Knowing the basics of how these operate makes it easier for you to understand other types you may be working with at your facility.

## 604. Military interfaces standards

As new electronic equipment is installed, the military establishes new standards. These standards are used as guides in planning and developing new military facilities. Here we discuss the standards we use, MIL-STD-188-C and MIL-STD-188-114A.

**MIL-STD-188-C.** The standards in this document apply to communications systems used primarily by Air Combat Command (ACC) in support of their military operations. Normally, tactical military personnel install and operate these communications systems. The types of systems range from tactical radio nets to large transportable communications complexes and fixed-station installations. In some cases, the standards may not be applicable to highly specialized systems with unique equipment requirements. Such systems should, however, follow these standards at any interface with other tactical systems.

The theater (tactical) military communications system provides communications service in every essential functional area: command and control, logistics, intelligence, weather, and administration. Traffic in the system can be in the form of voice, graphics, teletypewriter, or data and can be transmitted either as analog signals or as digital signals. The system accepts traffic from and delivers traffic to individual user stations and other systems.

The MIL-STD-188-C is similar to EIA RS-232-C in that it uses a 25-pin connector, and the pin configuration is the same. In addition to using the same connector, they both use unbalanced voltage interface circuits. We can look at the different leads and find some comparisons and differences that are present.

**Data leads.** Both transmit data on Pin 2 and receive data on Pin 3. MIL-STD-188-C has a positive mark and negative space. This makes it incompatible with the RS-232-C, which is positive space and negative mark. It operates on  $\pm 6$  volts.

Transitions between mark to space and vice versa must take a finite time to minimize emissions. It may take 50 to 15 percent of the total bit time (unit interval) to go from one binary state to another. This makes this standard more secure in terms of having less emissions than RS-232-C. This requirement gives the data signal a trapezoidal waveshape as indicated in figure 1-10.



Figure 1-10. MIL-STD-188-C trapezoidal waveshape.

**Clock leads.** Listed are some of the differences in the clock leads.

- 1) MIL-STD-188-C allows anywhere from 3 to 12 volts on clock leads. RS-232-C uses 6 to 12 volts.
- 2) Bit transition is triggered by negative to positive transition of clock. RS-232-C uses the positive to negative transition.

- 3) Bit state is sampled during the positive to negative transition of clock. RS-232-C uses the negative to positive transition.
- 4) Clock rate on both is twice the rate of data, thus giving it a sampling rate of once per bit for digital signals.

**Control leads.** Listed below is valuable information to remember regarding the control leads.

- 1) MIL-STD-188-C control leads serve the same function as the RS-232-C control leads.
- 2) MIL-STD-188-C allows for 3 to 25 volts for control signals. RS-232-C uses  $\pm 6$  to 12 volts, but normally operates at  $\pm 6$  volts.
- 3) The circuit is said to be in the off condition when no voltage or negative voltage is applied to the lead in both cases.
- 4) The circuit is said to be in the on condition when positive voltage is applied to the lead in both cases.

**Ground leads.** The ground leads should maintain 0-volts potential for both the protective ground and the circuit ground. Any potential causes distortion on data, control, and timing leads. The MIL-STD-188-C has a distance limitation of 1,000 feet, whereas the standard limit for the RS-232-C is 500 feet.

There is also a frequent need for DTE complying with RS-232-C standards to be connected to a communications path complying with MIL-STD-188-C. This calls for the introduction of some type of conversion equipment. One such unit is the CV-188-C converter, which converts the negative mark and square waveshape of RS-232-C to a positive mark and a trapezoidal waveshape of MIL-STD-188-C. This type of conversion is often necessary in the world of communications equipment that complies with many different standards. We talk more about converters later.

**MIL-STD-188-114A.** This standard specifies the electrical characteristics of the unbalanced (25-pin connector) voltage digital interface circuit and the balanced (37-pin connector) voltage digital interface circuit normally implemented in integrated circuit technology. It is compatible with the RS-449 standard for voltage and connector requirements. These circuits are employed for the interchange of serial digital binary signals between and among DTE and DCE or in any interconnection of binary signals between physically separated equipment. This is regardless of the type of information, such as digitized voice or data, that is represented by the binary signals.

The document is used in the design, installation, and operation of new communications facilities for both the long-haul and tactical systems. This document is applicable to data, timing or clock, and control circuits employed at the interface between equipment where the information being conveyed is in the form of binary signals at the direct current (DC) baseband level. It is also applicable to alarm and control circuits that are not directly related to data or timing.

This standard is applicable at all signaling rates regardless of the type of transmission medium used; for example, a nominal 4-kHz channel derived by frequency-division multiplexing (FDM), a nominal 48-kHz channel derived by FDM, a channel derived by time-division multiplexing (TDM), or a fiber optic or metallic wire connection. Figure 1-11 depicts a block diagram of a data terminal subsystem. It is an example to illustrate where the digital interface applies. This figure shows the data and timing circuits, but does not show the control circuits.

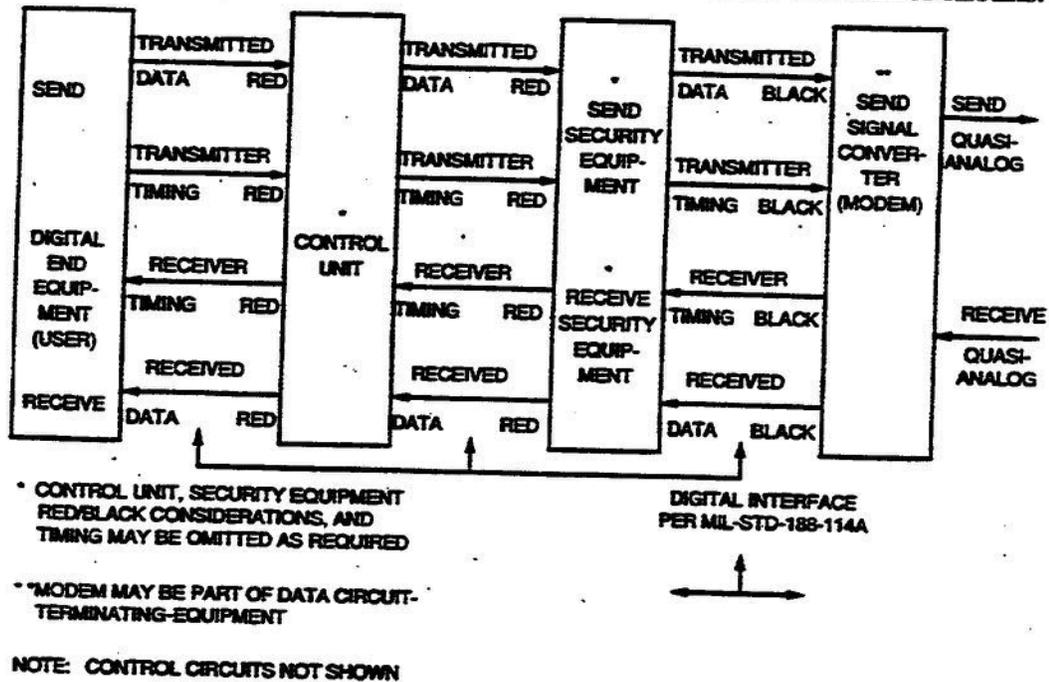


Figure 1-11. Block diagram of a data terminal system.

To comply with MIL-STD-188-114A, the interface circuits consist of a generator connected by an interconnecting wire or cable to a load. The load is comprised of one or more receivers and a termination resistor, where applicable. The electrical characteristics of the digital interface circuits are specified in terms of required voltage, current, and resistance values obtained from direct measurement of the generator and receiver components. The generator characteristics for the unbalanced voltage digital interface circuit are electrically different from the generator characteristics for the balanced voltage digital interface circuits; whereas, the receiver characteristics are electrically identical for both the unbalanced and the balanced voltage digital interface circuits.

The generator for a balanced circuit should have a low impedance (100 ohms or less) and produce a differential voltage of 2 to 6 volts. The generator for an unbalanced circuit should have a low impedance (50 ohms or less) and produce a differential voltage applied to the interconnecting wire or cable in the range of 4 to 6 volts. The differences make it impossible for simultaneous use as balanced and unbalanced interface circuits.

Functional interchange circuits are used to connect the DTE and the DCE for the purpose of exchanging data and timing signals and to control the flow of information. Flow control can be extended from one DTE through one or more DCE to another DTE on an end-to-end basis by appropriate communication protocols.

The unbalanced voltage digital interface circuit can generally be utilized on data, timing or clock, and control circuits where the signaling rate on these circuits is up to 100 kb/s. The balanced voltage digital interface circuit can generally be utilized on data, timing or clock, and control circuits where the signaling rate on these circuits is up to 10 Mb/s. The terminated voltage digital interface circuit can generally be utilized on data, timing or clock, and control circuits where the signaling rate on these circuits exceeds 10 Mb/s.

MIL-STD-188-114A is the first standard to specify the maximum loss that can be experienced over the path between the generator and the load (receiver). The maximum allowed is 6 dB signal voltage loss. A balanced circuit specifies 4,000 feet as the maximum allowable cable length for speeds up to 100 kb/s. As the modulation rate increases towards 10 Mb/s, the cable length is decreased to only 40 feet. An unbalanced circuit specifies 4,000 feet maximum for speeds up to 1 kb/s. As the rate increases to 100 kb/s, the cable length is decreased to 40 feet. Experience has shown that, in most practical cases, the operating distance at lower modulation rates can be extended to several miles.

The choice of either the balanced or the unbalanced voltage digital interface circuit is left to the designer and depends upon the signaling rate, the distance between generator and load, noise and grounding conditions, and other factors. It is not intended that all existing equipment be converted to comply with this standard, but when new systems are acquired or major modifications are completed, they should conform with this standard. This does not interfere with the purchase of new equipment, because new equipment still has the option of either having a positive or negative mark.

---

### Self-Test Questions

After you complete these questions, you may check your answers at the end of the unit.

#### 602. Industrial interface standards

1. What classes of data services are supported by EIA RS-232-C?
2. What data transmission specifications are supported by EIA RS-232-C?

3. What three distinct specifications is EIA RS-232-C established to meet?
4. What type of connector does EIA RS-232-C require?
5. What are the categories of pins used in RS-232-C connectors?
6. Briefly describe the purpose for each group of pins contained in a RS-232-C connector.
7. What advantages does the EIA RS-422-A standard have over the EIA RS-232-C standard?
8. Which EIA standard applies to equipment interface circuits where information is sent in the form of binary signals at a DC baseband level?
9. What EIA standard is similar to the RS-232-C standard and can, under certain conditions, be used for interoperation with RS-232-C specified equipment?
10. Which EIA standard is intended to replace the older RS-232-C standard?
11. What EIA standard operates in conjunction with either RS-422-A or RS-423-A?
12. What type of connector is specified under EIA RS-449?

### 603. Types of international interface recommendations

1. What agency establishes electrical interface recommendations for use across international borders?
2. What two international electrical interface recommendations do we use?
3. Match the descriptions in Column A with their corresponding international electrical interface recommendation in Column B. Column B items are used once.

#### Column A

- \_\_\_ (1) Interface between data terminal equipment (DTE) and data circuit-terminating equipment (DCE) for start/stop transmission services on public data networks.
- \_\_\_ (2) Modems for synchronous data transmission using 60-to-108 kHz group b/s circuits.
- \_\_\_ (3) 9.6 kb/s modem for use on leased circuits.
- \_\_\_ (4) 200-b/s modem standardized for use in the general switched telephone network.
- \_\_\_ (5) General-purpose interface between DTE and DCE for synchronous operation on public data networks.
- \_\_\_ (6) 2.4 kb/s or 1.2 kb/s modem standardized for use on four-wire leased circuits.
- \_\_\_ (7) 4.8 kb/s modem standardized for use on leased circuits.
- \_\_\_ (8) 600/1.2 kb/s modem standardized for use in the general switched telephone network.
- \_\_\_ (9) Data transmission of 48 kb/s using 60-to-108 kHz group b/s circuits.
- \_\_\_ (10) Interface between DTE and DCE for terminals operating in the packet mode on public data networks.

#### Column B

- a. X.20.
- b. X.21.
- c. X.25.
- d. V.21.
- e. V.23.
- f. V.26.
- g. V.27.
- h. V.29.
- i. V.35.
- j. V.36.

### 604. Military interface standards

1. What are the two military standards that are used as a guide in the planning and developing of new military facilities?

2. Which standard primarily deals with theater (tactical) military operations?
3. The MIL-STD-188-C is similar to what EIA standard?
4. Which standard can have either unbalanced or balanced electrical characteristics?
5. How are the electrical characteristics of the digital interface circuits in MIL-STD-114A specified?
6. What is the maximum allowed signal loss you can experience over the path between generator and receiver with MIL-STD-188-114A?

### **1-3. Digital Signal Conditioning Devices**

In the future, you'll find that some circuits coming into your station do not arrive in the proper condition or format. When this occurs, you will be required to make some changes at your station so the circuit can be made operational. The equipment we use on numerous occasions are digital signal conditioning devices.

#### **605. Purpose and operation of a converter**

By definition, a converter is a device used to change information from one form to another without changing the meaning. In computer technology, a converter is used to translate data from one form of expression to a different form. We are converters at times; when we read a story and put it in our own words to tell someone else, we basically relate the same information we read, but put it in words the listener understands. We have many types of converters. In the following text, we discuss a few you may find at your station.

**Serial-to-parallel converter.** A serial-to-parallel converter is a device that accepts a single time sequence of signal elements and distributes them among multiple parallel outputs. It also accepts a single time sequence of signal states representing data and translates these states into a spatial distribution. Most teletype (TTY) multiplexers are examples of serial-to-parallel converters.

**Analog-to-digital converter.** An analog-to-digital converter is a device that converts an analog input signal to a digital output signal carrying equivalent information, and vice versa. The device changes a continuously varying voltage or current into a digital output. The input may be AC or DC, and the output may be serial or parallel, binary or decimal. A modulator/demodulator (MODEM) is an example of this device.

**Hybrid converter.** A hybrid converter takes a two-wire connection and transforms it into a four-wire connection, and vice versa. Also, any device that provides impedance matching between certain circuits may be referred to as a "hybrid converter." The hybrid used, of course, depends upon knowing the impedance of the loop and the frequency range of interest so it can provide a balanced circuit. If the impedance is somewhat variable, the hybrid can provide an imperfect balance at best. The difficulty is compounded by the fact that normal telephone hybrids need only provide balance at voice frequencies, while the digital subscriber loop hybrid must provide a balance over a bandwidth approaching 100 kHz. An example of a hybrid converter is a four-wire termination unit that allows a balanced four-wire circuit to be interconnected with a balanced two-wire circuit.

**Level converter.** A level converter is a device that converts high-level keying to low-level keying, and vice versa. This device can be neutral or polar polarity. Converting  $\pm 60$  volts DC for high-level polar keying to  $\pm 6$  volts DC for low-level polar keying is an example of this; a level converter is more commonly called a line-level interface unit (LLIU).

**Code converter.** A code converter is a device that converts the bit grouping for a character in one code into the corresponding bit grouping in another code. Changing American Standard Code for Information Interchange (ASCII) code to Extended Binary Coded Decimal Interchange Code (EBCDIC) is a good example. The ASCII is the latest effort of the common carriers and computer communications industry in the United States to produce a universal common language code. It has been adopted by the American National Standards Institute (ANSI) and is currently the standard code for computer communications. Although individual computers use different internal codes for word storage and character displays, the ASCII is the most widely used.

EBCDIC is similar to the ASCII, but it is a true 8-bit code. The eighth bit is used as an added bit to extend the code, providing 256 distinct code combinations for assignment. The code is constructed in binary sequence as is the ASCII, but it lacks a parity bit because all 8 bits are used for information. Its sequence of comparing the letters before the numbers is the reverse of that used with the ASCII. The EBCDIC was developed by IBM primarily for use in scientific applications and for transmission between byte-oriented computers.

The code converter device is a decision making-type of digital building block that converts information received from its inputs to another digital code that is

transmitted at its outputs. Sometimes these devices are called encoders and decoders.

**Optical isolator.** An optical isolator is an optical link inserted into a communication system to provide electrical isolation between two or more parts of the system and signal conversion. It is used primarily to electrically isolate digital signals flowing between red (unencrypted) and black (encrypted) equipment rack configurations. It accomplishes isolation as it converts digital signal pulses originating from one piece of equipment to infrared light pulses. The light pulses are then transmitted through fiber optic cables to other optical isolators at the receiving end that convert the light pulses back into digital signal pulses. You should note however, that the isolator *does not* have cipher capabilities. When used for red-to-black isolation purposes, the isolator works in conjunction with some type of cryptographic device.

### 606. How encryption/masking devices are used

Like a vault in a bank, information and communications security has been based on the concept of a secret combination. The sender of the information locks the transmitted information with the secret combination, or key, and the receiver, the only person with knowledge of the key, can unlock the information. In communication links, this is called *encryption*.

Applying the same security principles to communications links provides much the same advantages to the sender, as long as the key is kept secret. The economics of the encoding process greatly favor the authorized users of the communications links. For example, in 1980 the trends in data processing costs to encode was possible at about 20,000,000 bits for 10. The required decoding by an intruder without the secret key would require 20,000,000<sup>2</sup> operations, or a total of  $(4 \times 10)^{14}$  bit operations, for an estimated processing cost of more than \$2 million. If the decoding processor could manipulate 100 Mb/s, it would take the processor more than 46 days, operating 24 hours a day, to break the code of a single message.

The encryption process takes place as a mathematical manipulation of the sender's message, with an inverse process taking place at the receiving end. With binary communications, the mathematical process can be simply binary addition of a randomly chosen sequence with a similarly long portion of the message. If the randomly chosen key is at least as long as the message and used only once, the message is unconditionally secure and cannot be broken regardless of processing power. However, this is not a very practical way to perform the encoding. As we have seen, a finite-length key can be sufficiently long to ensure conditional security; that is, determining the proper key by an exhaustive search is impossible in a practical sense. With a finite-length key and a relatively simple mathematical operation at the sending and receiving ends, a high degree of protection can be achieved. The encryption can be applied on either a link-by-link basis or an end-to-end basis.

**End-to-end encryption.** In end-to-end encryption, two users of a distributed network each apply encryption devices at their terminal locations and agree on the secret key to be used. End-to-end encryption provides the greatest degree of protection since the information is fully encoded all through the network, except at the users' own end terminals. However, end-to-end encryption is the most difficult to implement. In order to ensure proper operation of the decoding process at the receiving end, the decryption mechanism must be synchronized with the encryption mechanism. There are a number of ways to do this, but each is rendered less effective by the presence of variable delays. In addition, if the network uses switching, there must be a method for transmitting the address and signaling information in the clear, so that the overhead information can be understood by the network switches and control facilities.

**Link encryption.** Link encryption is applied on each line segment in the distributed network, so that all information flowing on the communications line is unintelligible to an unauthorized intruder. Even the overhead, address, and control information are protected. This offers traffic flow security as well as information security. (Traffic flow security means that an observer does not even know if and when information is actually being transmitted on the communication line.) The major disadvantage of link encryption is that the information has to be decoded at each nodal element of the network so the information is intelligible to an intruder who might penetrate the switching or processing elements of the network. Link encryption also requires more encryption devices in a typical network since the lines between the switches and nodes as well as the lines between the users and the network have to be individually encrypted. However, in truly distributed networks and networks based on broadcasting techniques, the distinction between link and end-to-end encryption is minimized since, for the most part, the traffic moves directly between the source and destination users.

**Bulk encryption.** We also have bulk encryption, in which the information on several channels is encrypted simultaneously by a single encryption device. A typical arrangement uses a multiplexer to combine several channels into a single channel for encryption.

## 607. The purpose of coders/decoders (CODECs)

Another electronic development provides one final illustration of the indistinction of the line between voice and data communications: CODECs are electronic circuits that code and decode analog voice signals into digital signals. Until recently, CODECs were part of a private branch exchange (PBX), which, in turn, was hooked up to the telephone system. Now, because technological advances are bringing the cost of these semiconductor chips down to a very low price, it is possible to install the CODEC directly into the telephone set; the telephone then is able to convert the human voice directly into a digital signal. As a result, digitized voice signals are transmitted like digital computer signals, and it is possible to transmit both voice and data digitally on the same wires at the same time. In this

environment, most of the practical distinctions between voice and data communications disappear.

We see CODECs used directly in the telephone set first with digital PBXs. When these PBXs use a telephone with a built-in CODEC and are connected to a data terminal or computer, they can transmit both voice and data simultaneously over one or two pairs of wires by using digital transmission techniques.

Eventually, we will see this approach expanded to the local loops from the telephone dial exchange, so that all phones will be "digital," and you will not need a PBX in order to use digital transmission. However, because of the large investment the military has made in analog local loops and central base exchanges, it will be many years before digital telephones will replace today's analog telephones for direct connections to the central dial exchange.

## Self-Test Questions

After you complete these questions, you may check your answers at the end of the unit.

### 605. Purpose and operation of a converter

1. What is the purpose of a converter?
2. Match the functions in Column A with the correct type of converter in Column B. Column B items may be used once or more than once.

#### *Column A*

- \_\_\_(1) Accepts a single time sequence of signal elements and distributes them among multiple parallel outputs.
- \_\_\_(2) Accepts an AC or DC input and provides a serial or parallel, binary or decimal output.
- \_\_\_(3) Optically provides electrical isolation and signal conversion.
- \_\_\_(4) Changes a continuously varying voltage or current into a digital output.
- \_\_\_(5) Changes a two-wire connection to a four-wire connection and vice versa.
- \_\_\_(6) Accepts a single time sequence of signal states representing data and translates the signal states into a spatial distribution.
- \_\_\_(7) Provides impedance matching between certain types of circuits.
- \_\_\_(8) Converts an analog input signal to an equivalent digital signal, and vice versa.
- \_\_\_(9) Converts the bit grouping for characters of one type of data signaling code into the corresponding bit grouping of another data signaling code.
- \_\_\_(10) Changes ASCII to EBCDIC.
- \_\_\_(11) Converts high-level keying to low-level keying, and vice versa.
- \_\_\_(12) Works in conjunction with a cryptographic device to isolate "red" and "black" equipment configurations.
- \_\_\_(13) Converts digital pulses to infrared light pulses and vice versa.
- \_\_\_(14) Encodes and decodes digital signals.

#### *Column B*

- a. Serial-to-parallel.
- b. Analog-to-digital.
- c. Hybrid.
- d. Level.
- e. Code.
- f. Optical isolator.

**606. How encryption/masking devices are used**

1. What is the purpose of an encryption device?
2. What encryption method is considered to provide the greatest degree of protection? Why?
3. What encryption method is the most difficult to implement? Why?
4. What encryption method is used to secure individual line segments of a distributed system?
5. What is a major disadvantage of link encryption?
6. Of the three encryption methods, which requires the most encryption devices within a typical network? Why?
7. What encryption method is used to secure several communication lines simultaneously using a single encryption device?

**607. The purpose of coders/decoders (CODECs)**

1. What is the purpose of CODECs?
2. With technological advances bringing cost of semiconductor chips down, what possibilities are anticipated for transmission with digitized voice signals?
3. What impact does a CODEC device installed directly to a telephone set have?

## Answers to Self-Test Questions

### 600.

1. Analog.
2. Digital.
3. Two levels.
4. A three-level signal.
5. A signal that, when keyed to one level, remains at that level until it is keyed to another level.
6. The level.
7. In the transition.
8. Positive.
9. A ternary signal in which logic zeros are represented by zero voltage and logic ones are represented by alternate positive and negative voltages.
10. Current represents a mark and absence of current represents a space.
11. Marks are represented by a positive voltage and spaces are represented by negative voltages.
12. It is little affected by variations in line characteristics.
13. Polar keying gives less distortion than neutral keying on the same communication line.
14. A unit of information based on two symbols, states, or conditions.
15. Bits per second (b/s).
16. The unit of modulation rate of the shortest unit interval.
17. Baud is an expression of time (duration) while bit carries no suggestion of time.
18. Overflow is when more bits are received than expected. Underflow is when less bits are received than expected.

### 601

1. Synchronous, asynchronous, and isochronous.
2. None.
3. In synchronous operation, the receiving device is adjusted automatically to the speed of the transmitting device by comparing the speed of the incoming signal with the time base of the receiving device.
4. Any signal that contains synchronizing bits within its signal stream.
5. In asynchronous operation, a receiving device is started with a start bit and runs only until it receives one character, then a stop bit causes it to stop and wait on the next start bit.
6. It slows transmission speed.
7. A signal in which all bits are of equal duration.
8. Isochronous.

### 602

1. Dedicated leased or private line service and switched network service, either two-wire or four-wire.
2. Asynchronous and synchronous transmissions up to 20 kb/s.

3. Type of connector, purpose and definitions of connector pins, and electrical characteristics of those pins.
4. A plug-in, 25-pin connector.
5. Data, clock, control, and common, or ground.
6. Data leads carry intelligence between DTE and DCE. Clock leads provide timing for synchronization. Control leads enable and disable transmission and reception and indicate equipment status. Ground leads establish the common reference potential.
7. It specifies a balanced interface that operates at much higher speeds and over longer distances.
8. EIA RS-423-A.
9. EIA RS-423-A.
10. EIA RS-449.
11. EIA RS-449.
12. A 37-pin connector and a 9-pin connector.

**603**

1. International Telegraph and Telephone Consultative Committee (CCITT).
2. X-series and V-series.
3. (1) a, (2) j, (3) h, (4) d, (5) b, (6) f, (7) g, (8) e, (9) i, (10) c.

**604**

1. MIL-STD-188-C and MIL-STD-188-114A.
2. MIL-STD-188-C.
3. EIA RS-232-C.
4. MIL-STD-188-114A.
5. In terms of voltage, current, and resistance values obtained from direct measurement of the generator and receiver components.
6. 6 dB.

**605**

1. To change information from one form to another without changing its meaning.
2. (1) a, (2) a, (3) f, (4) b, (5) c, (6) a, (7) c, (8) b, (9) e, (10) e, (11) d, (12) f, (13) f, (14) e.

**606**

1. To secure communications between a sender and receiver.
2. End-to-end because the information is fully encoded throughout the network.
3. End-to-end because encryption and decryption devices must be synchronized.
4. Link encryption.
5. Information has to be decoded at each nodal element of the network, making it susceptible to intrusion.
6. Link encryption because the lines between switches and nodes and lines between users and networks have to be individually encrypted.
7. Bulk encryption.

607

1. To code and decode analog voice signals into digital signals.
2. A telephone set will be able to convert the human voice directly into a digital signal.
3. A PBX is not needed in order to use the digital transmission media.

**Do the Unit Review Exercises (URE) before going to the next unit.**

## Unit Review Exercises

**Note to Student:** Consider all choices carefully, select the *best* answer to each question, and *circle* the corresponding letter. When you have completed all unit review exercises, transfer your answers to ECI Form 34, Field Scoring Answer Sheet.

**Do not return your answer sheet to ECI.**

1. (600) A binary signal conveys information in the form of
  - a. two levels.
  - b. three levels.
  - c. complex analog signals.
  - d. complex ternary signals.
  
2. (600) Individual pulses within a digital signal are called
  - a. bits.
  - b. bytes.
  - c. codes.
  - d. blocks.
  
3. (600) What unit is *most* commonly used in data systems to express signaling speed?
  - a. Baud.
  - b. Baudot.
  - c. Sense interface.
  - d. Bits per second.
  
4. (600) Bit count integrity problems are primarily caused by
  - a. impedance mismatch.
  - b. system timing faults.
  - c. improper equipment settings.
  - d. changes in atmospheric conditions.
  
5. (601) In synchronous data operation,
  - a. timing pulses are transmitted within the signal stream.
  - b. start and stop pulses control data transmission.
  - c. system timing is not of critical importance.
  - d. all data bits are the same length in time.

6. (601) A digital signal that contains synchronizing bits within the signal stream describes
  - a. synchronous operation.
  - b. asynchronous operation.
  - c. synchronous and/or isochronous operation.
  - d. asynchronous and/or synchronous operation.
  
7. (602) What type connector does the Electronic Industry Association (EIA) standard RS-232-C specify be used?
  - a. 9-pin.
  - b. 13-pin.
  - c. 25-pin.
  - d. 37-pin.
  
8. (602) The Electronic Industry Association (EIA) standard that was designed specifically for point-to-point configurations is the
  - a. EIA RS-232-C.
  - b. EIA RS-422-A.
  - c. EIA RS-423-A.
  - d. EIA RS-449.
  
9. (602) The new EIA RS-449 interface standard operates in conjunction with
  - a. both balanced and unbalanced circuits.
  - b. both balanced and terminated circuits.
  - c. unbalanced circuits only.
  - d. balanced circuits only.
  
10. (603) The X-series in International Interface recommendations relate to
  - a. data networks.
  - b. WWMCCS networks.
  - c. switched networks.
  - d. telephone networks.
  
11. (603) Which International Interface recommendation series deals with telephone networks?
  - a. T-series.
  - b. V-series.
  - c. X-series.
  - d. Y-series.

12. (604) The standards in MIL-STD-188-C apply to communications systems used mainly by
  - a. Air Combat Command (ACC).
  - b. Air Force Material Command (AFMC).
  - c. Air Education and Training Command (AETC).
  - d. Air Force Special Operations Commands (AFSOC).
  
13. (604) The military standard that applies to unbalanced 25-pin and balanced 37-pin connectors is the
  - a. MIL-STD-188-C.
  - b. MIL-STD-188-114A.
  - c. MIL-STD-188-300.
  - d. MIL-STD-188-544C.
  
14. (605) A device that accepts a single time sequence of signal elements and distributes them among multiple parallel outputs is a
  - a. serial-to-parallel converter.
  - b. analog-to-digital converter.
  - c. hybrid converter.
  - d. code converter.
  
15. (605) Which of the following is an *example* of a hybrid converter?
  - a. Modem.
  - b. Repeater.
  - c. Termination unit.
  - d. Line-level interface unit (LLIU).
  
16. (605) A device that converts digital signals into infrared light pulses is
  - a. an optical isolator.
  - b. a cryptographic device.
  - c. a statistical multiplexer.
  - d. a digital-to-fiber converter.
  
17. (606) The encryption method that offers the greatest degree of protection is
  - a. link encryption.
  - b. bulk encryption.
  - c. segment encryption.
  - d. end-to-end encryption.

18. (606) All of the following are methods of encryption *except*
- a. link.
  - b. bulk.
  - c. end-to-end.
  - d. single-line.
19. (607) The use of CODECs built directly into a telephone will enable which of the following to be transmitted simultaneously over one pair of wires?
- a. Two data circuits.
  - b. Two voice circuits.
  - c. One voice and one data circuit.
  - d. One voice and one teletype (TTY) circuit.
20. (607) In terms of convenience, why is the installation of CODEC devices directly into the telephone set so advantageous?
- a. Dialing is easier.
  - b. Less circuitry is required.
  - c. Defense Switched Network (DSN) processing is quicker.
  - d. Private branch exchange (PBX) assistance is no longer needed.

**Please read the unit menu for Unit 2 and continue. →**



## Unit 2. Computer Systems and Equipment

	<i>Page</i>
<b>2-1. Computer Operation Principles .....</b>	<b>2-2</b>
608. Computer classifications and terminology .....	2-2
609. Internal circuit functions .....	2-6
610. Internal control functions .....	2-8
<b>2-2. Types of storage and devices .....</b>	<b>2-15</b>
611. Disk subsystems .....	2-15
612. Hard disk .....	2-16
613. Portable disk systems .....	2-19
614. Magnetic tape systems .....	2-24
615. Paper tape systems .....	2-28
<b>2-3. Peripheral Devices .....</b>	<b>2-33</b>
616. Consoles and terminals .....	2-34
617. Printers .....	2-39
618. Optical read units .....	2-44
619. Monitors .....	2-46
<b>2-4. Operating Systems and Computer Memory .....</b>	<b>2-52</b>
620. Operating system functions .....	2-52
621. Popular operating systems .....	2-54
622. Computer memory .....	2-55
623. Memory management .....	2-62

**T**HE digital computer extends a person's thinking by doing routine and time-consuming computations. The computer solves mathematical problems, sorts and classifies data, and makes logical decisions according to given instructions. In principle, there is little general difference in the present-day electronic digital computers and the earlier machines. The great advances are in circuit design and the use of new components. These advances have resulted in greater capabilities, broader uses, and much greater speeds.

The greater capabilities and broader uses ultimately appear in newly developed equipment; therefore, it is important to increase your understanding of computer

fundamentals. Inevitably, the increased knowledge will improve your ability to maintain electronic equipment.

## 2-1. Computer Operation Principles

One thing that causes trouble for many people is the idea that a computer "thinks." Computers *do not think*. A computer is basically a mechanical or electronic device that manipulates information under automatic control. A computer performs simple math operations and makes logical comparisons. Any "decision" made by a computer is based on a comparison of some type. People provide the data for the comparison. Let's explore some of the various types of computers and how they operate.

### 608. Computer classifications and terminology

One very important development of modern digital computers is the *stored program concept*. Initially a computer's instructions are provided by removable circuit boards with connecting cords that are plugged into various electric terminals (plugboard wiring), or by some external medium such as magnetic tape or magnetic disk.

In the stored program concept, the computer's instructions are stored within the computer's memory. When one instruction is completed, the computer has immediate access to the next instruction. All modern digital computers use the stored program concept.

**Development of the computer.** Some say Stonehenge, a mysterious grouping of prehistoric stones in Great Britain, is a computer of sorts. Archaeologists believe that prehistoric people worked out a calendar from the position of shadows as the sun shined on the stones. We can think of numerous analogies similar to this and argue about practically every tool, including a pencil, but here, we limit our discussion of computers to the modern definition.

The analytical engine was invented by an English mathematician, Charles Babbage, who lived from 1792 to 1871. He designed his machine to do complicated sums and store the results of each stage in the calculations. This analytical engine never worked, but Babbage's ideas are the basis for modern computers.

The development of modern computers can be divided into three main categories or generations. The first generation was the large mainframes that were built with vacuum tubes. The second generation was the smaller, more reliable computers that were built with transistors. The third generation is the computers made with silicon chips.

Following is a list of historic dates of the modern computer.

In 1945, the electronic numerical integrator and calculator (ENIAC) was built. It was the first all-electronic machine and was more like a calculator than a present-day computer. It could not store data or programs.

In 1947, the transistor was invented. It was not used in computers until about 1953.

On 21 June 1948, the Manchester University Mark I, the first true computer (that is, one which could store a program of instruction), ran for 52 minutes.

In 1950, the Ferranti Mark I, based on the Manchester Mark I, was sold commercially in Europe.

In 1958, the first working integrated circuit was developed.

In 1960, the first "chips" (integrated circuits on chips of silicon) were produced.

In 1964, the first computers built with integrated circuits were manufactured for the general market.

In 1976, the first small computer, the Altair, was offered commercially.

In 1980, the first pocket computer, the Japanese Sharp PC1211, was sold.

**Mainframes.** First generation computers were big and expensive and used a lot of power. They were called "mainframes" because the parts were mounted on frames in large metal cabinets. The large, powerful computers of today are still called mainframes, but now we have small machines called "minis" and even smaller, desk-top models called "micros." Over the last 40 years, computers have progressively become smaller, cheaper, and more powerful. Whatever their size, all computers have the same basic parts.

The equipment for a large, modern computer can easily fill several rooms. There are rows of data-storage cabinets containing information for the computer, as well as many different kinds of input and output equipment such as printers, monitors, and keyboards. A modern mainframe can carry out millions of instructions every second and work quick enough to do many different jobs at once. Although slow compared to today's microcomputers, mainframes with their powerful input/output processors have provided processing solutions for decades and will be used for years to come. They will remain useful until their huge databases and other programs are transferred onto other media for use with smaller computer systems.

**Minis.** A minicomputer is smaller than a mainframe and cannot handle as much data as a mainframe or work as fast as the super computers. A modern minicomputer is still many times more powerful than the mainframes of the early days of computers. Minicomputers are designed for one particular type of work, while a mainframe does many different jobs.

**Micros.** When the less expensive microcomputers were developed, many more people could afford a computer. Present-day micros are not as powerful as larger

computers, but most can connect to extra equipment so they can store additional information and operate input and output machines such as plotters and printers. For their size, microcomputers are the most powerful computers in use today. They are easily networked to provide resource sharing that negates the need for large centralized processors.

**Electronics.** Today, all computers are electronic; that is, all their work is done by pulses of electricity. Development work on the first electronic machines began in the 1940s in an effort to crack enemy codes and work out target distances for the artillery in World War II.

One of the first American electronic machines, ENIAC, did thousands of calculations a second, but it was not a true computer because it could not store information or instructions. It was, however, much faster than a mechanical calculator; that is, one which works by moving gears and wheels.

The Manchester University Mark I, an early British electronic computer, was not as fast as ENIAC, which could do 800 calculations per second; however, it could store the instructions for carrying out a series of calculations. Because of its storage ability, it is considered to be the first true computer. The Mark I was built with electronic parts left over from World War II and first ran (for 52 minutes) on 21 June 1948.

Modern computers are powerful machines used to collect, analyze, and process infinite amounts of information rapidly and with a near-perfect degree of accuracy. The physical form of this information is called *data*. A computer processes data by executing a sequence of precise instructions called a *program*. Computer programs are designed and constructed by people (programmers) to perform data processing tasks that solve real world problems. Programming a computer is meticulous work because each computer understands only a limited set of very exact instructions.

**Computer terms.** As the development of the computer progressed, so did computerese, the technical language of the computer profession. It isn't necessary to learn all terms associated with computers to develop an understanding of them, but a basic computerese vocabulary will help.

**Address.** An address is defined as a label, symbol, or other set of characters used to designate a location or register where data is stored. In computers, binary signals are decoded and the result is used to select a specific storage register.

**Bit.** Information stored in a computer's memory is in the form of "bits." Bits are the smallest unit of information recognized by the computer system; a single one or zero, yes or no, or true or false. The term "bit" is a contraction of *Binary digIT*.

**Bus.** A bus is a group of conductors used to transfer signals. Buses are often identified by the type of signals they carry. For example, a group of conductors carrying control signals is often referred to as a "control bus." There are three principal buses in a computer: the address bus, the data bus, and the control bus.

A bus extends from the central processing unit (CPU) as a series of wires to each component in the computer system and allows information to flow in either direction. Like a highway with several lanes, information is traveling towards its intended destination. A bus' capacity is measured in the number of bits transferred between the CPU and other components. The most common types are Industry Standard Architecture (ISA) that has a transfer rate of 8 and 16 b/s, Expanded Industry Standard Architecture (EISA) that transfers 32 b/s and is faster than ISA simply because it transfers at twice the rate, Micro Channel Architecture (MCA) that also transfers at the 32-b/s rate but allows interface with two processors, and finally, the Local Bus which is much faster than conventional bus systems and provides a direct connection to the computer's system board or expansion board to help alleviate traffic on the ISA bus.

**Byte.** The term "byte" generally refers to a unit of 8 bits of data. If there are eight flip-flops in each storage register, the register stores one byte of data. When a byte contains 8 bits, eight separate conductors are required to transfer a byte of data at one time.

**Clock.** Within the computer, there is a master-timing generator called a "clock". The clock can be as simple as a free-running oscillator that produces timing or clock pulses at a fixed frequency. Its function is to provide a signal to synchronize the internal flow of instructions and data through the computer system. The computer's internal states and the states of the input and output lines may be designed to change only when a clock pulse appears. Thus, the clock can control the timing of computer operations. The clock rate is used to assess a processor's capability. The faster the clock rate, the faster the processor functions.

**Data.** The contents of each storage register are referred to as "data." If there are four flip-flops in each storage register, the register stores 4 bits of data. Data, when used as a general term, denotes numbers, letters, symbols, etc., that a computer can produce or process.

**Instruction.** An instruction is a sequence of binary digits that tells the computer what operation to perform. For example, we may want to add a number "A" and a number "B." A and B are contained in different storage locations. The instruction, "Add A and B" causes the computer to add the numbers. Other instructions can make the computer display the results of the addition.

All computers *do not* use the same kinds of instructions. The sequence of digits (instruction) that causes one computer to add may make another computer subtract; however, some computers do have common or compatible instructions.

**Program.** A computer program is a series of instructions for the computer to execute. The program tells the computer what to do and when to do it. If the program sequence is coherent and logical, processing the program produces intelligent and useful results.

**Software and hardware.** Computer programs are often referred to as *software* as opposed to the actual mechanical, magnetic, electrical, and electronic components

that make up the computer and that are referred to as *hardware*. Hardware can also refer to the interconnection of chips to form a computer. The software makes the hardware work.

The hardware in a computer limits what the software can tell it to do. If you told a computer to fly, it couldn't unless it was designed for flight. Without the necessary components (hardware), the computer cannot perform the operation you desire.

**Word.** A word is *generally* made of more than one byte. This is actually a loose definition because the number of bits in a word ultimately depends on the design of the computer. Some processors handle a greater number of bits *as a unit* than others. For example, one type of microprocessor may be manufactured to handle 16-byte words while another is designed to handle 64-byte words. In some cases, a word is the largest group of bits treated *as a unit* throughout the computer's central processor.

### 609. Internal circuit functions

The five essential functions of a digital computer can be diagrammed in block form, as shown in figure 2-1. All digital computers have certain characteristics in common, no matter how much they vary in size, speed, or function. The diagram in figure 2-1 is so general in nature that it can be used to represent the characteristics of data processing machines ranging from the desk calculator to the most intricate and high-speed electronic data processing equipment.

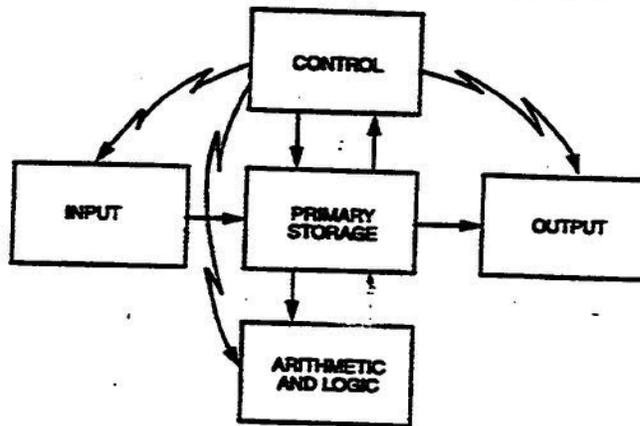


Figure 2-1. Logistical sections of a computer.

It's not easy to move from such a diagram to the hardware itself. The functional block diagram might mislead you to expect that a computer is mechanically or physically divisible into separate blocks. This is not true. Unless you are very familiar with digital computer hardware, it would be difficult to identify all of the components that perform each particular operation outlined in the functional block diagram.

Figure 2-1 shows either a one-way or two-way connection to other blocks. These interconnecting lines do not represent single wires. While there is a general direction of movement from input to output, there are many possible paths a signal can take. The computer designer establishes the interconnecting network inside and between the blocks. Circuit components, logically interconnected, are the building blocks of the computer. Some circuits in the computer are uniquely designed to do one job, while standardized circuits such as counters, adders, or registers can be applied as needed.

The digital computer's basic components fall into two general classes—those which store or retain information (memory elements) and those which make decisions based on the information supplied to them (decision elements). These two classes are applicable to almost any block in the functional block diagram and need further definition. Each block in the functional block diagram performs a specific operation.

**Input.** The information used by the computer has to be “read” into the system, and must be translated into computer language (binary). The input unit does this. The input can be fed in manually from a keyboard or provided from magnetic tape or disk, a modem, a light pen, a mouse, a touch-sensitive video terminal, or even from a speech recognition system.

**Output.** The computer sends results of operations, as dictated by the program, to an output device. The output results must be translated into the desired form. The output can be displayed on a video screen, recorded on a magnetic tape or disk, or sent to a printer or modem. It can also produce the proper response through a recorded message of voice synthesis.

**Input/output.** In all computers, I/O is under the control of the program being executed. The input/output (I/O) section of the computer provides a communications capability between the computer and a peripheral device. A peripheral device is an external device, such as a printer or visual display unit, that is connected to the computer. The I/O section controls data transfers between the computer and peripheral devices. These devices are frequently used to provide input to the computer, or to display output from the computer. Peripheral devices cannot be connected directly to the computer's internal address and data buses. They are connected through interface devices.

**Interfacing.** The electronic circuit that connects the peripheral device to the computer's I/O system is termed an *interface*. The process of interfacing can be defined as “the connecting of peripherals, digital or analog data handling equipment, and communications hardware to a computer so that they can work together in a consistent manner”. Interfacing requires special circuits to implement the connections and software to make it function. Data may be sent from the computer to a peripheral in either serial or parallel format. The peripheral device determines the format used. In serial format, data is transferred one bit at a time; in parallel format, it is transferred many bits at a time.

**Primary storage.** Not all of the data fed into the computer can be used at one time, nor can all the answers be processed at one time. A system for storing results is required. *Memory* is that section of a computer that is used for storing information. Information is stored in locations which have *addresses* to identify where in memory it is located. Each address contains a word, the size of which is determined by the machine. Each location can store one binary word of a fixed number of bits. Various types of memory devices are available. We discuss memory circuits later.

**Arithmetic and logic.** As stated earlier, a computer performs simple math operations and makes logical comparisons. The section of the computer which deals with arithmetic and logic functions is commonly referred to as the *arithmetic logic unit (ALU)*. The ALU can only add and perform logic operations. By manipulation of the input data, the ALU can simulate subtraction, multiplication, and division. The ALU can solve problems in geometry, trigonometry, algebra, calculus and logical reasoning when it follows the proper program.

**Control.** To solve problems or process data, a number of operations must be carried out in sequence. The program's instructions control these operations. The control unit decodes the instructions and then produces a series of control signals used by the other units to complete the action called for by the instruction. Not every digital computer refers to the five functional blocks the same way, but the basic functions do not change. The ALU and control sections together are called the central processing unit, or CPU.

## 610. Internal control functions

The physical machinery of a computer is referred to as "hardware." Each hardware device consists of electronic circuits and wires assembled to create certain data processing capabilities. Basically, a computer has four functional units: a central processing unit (CPU), an input unit, an output unit, and a memory.

**Central processing unit (CPU).** The CPU, also called a central processor, is the brains of the computer system. It is responsible for processing information, which may be an instruction or data. Instructions can cause data to move from one location to another or have mathematical and/or logical operations performed on it. Busses are used within the CPU chip to transfer data from one register to another. The CPU must be able to call instructions from memory, decode their binary contents, and execute them.

**Interrupt requests (IRQ).** The CPU must interface with the computer's input/output section (I/O). One method used in programs is for the CPU to periodically "check" with the I/O to see if it needs service. Often I/O operations are initiated by the peripheral device through the use of *interrupt requests*. When the CPU receives the interrupt request, it halts action on the main program,

services the I/O, then resumes the main program. Interrupts are important because they let the processor execute the system program without requiring the program to monitor the status of each peripheral device. The software that handles the operation of each peripheral is executed only when required.

The functional units that let a CPU perform its functions are the registers, the arithmetic logic unit and the control circuitry.

**Registers.** Registers are temporary storage units within the CPU. Some registers, such as the program counter and instruction register, have dedicated uses. The *program counter* identifies the location, in memory, of the next instruction to be performed. The instruction is then placed in the *instruction register*, decoded by the *instruction decoder*, and executed. Other registers are for more general purpose uses.

**Arithmetic logic unit (ALU).** The ALU typically has two inputs and two outputs. A temporary register and an acting accumulator feed the ALU. The main processes of a computer are its arithmetic operations. The ALU can perform all the major mathematical and logical operations on the two inputs. Mathematical problems are solved by making certain decisions in a certain sequence, based on stored numerical data and operation instructions. Accumulator registers receive all results of arithmetic operations.

The flag register, also called the status register, indicates specific conditions after an arithmetic or logic operation. For example, if two binary numbers are subtracted, and the result is zero, the zero flag is set. Other types of flags include a carry flag, an overflow flag, a negative (sign bit) flag, an interrupt mask flag, and a half carry flag. The flag register is very important because the CPU uses these flags to make decisions.

**Control circuitry.** The control unit directs and coordinates the step-by-step operations of a computer which includes directing and sequencing arithmetic and logic operations, accessing memory, and controlling input/output (I/O) devices through I/O controllers. The control unit is managed by stored programs (sets of instructions) that tell the CPU the exact steps to follow in the processing of data.

The control circuitry is the primary functional unit in a CPU. The control circuitry may have a control sequencer, which may have some other name—but its function is the same. The control sequencer receives its input from the decoder, then generates control signals in the proper order to accomplish the instruction. It is responsible for managing the control bus. There are wires running from the control sequencer to every unit in the computer. Using clock inputs, the control circuitry maintains the proper sequence of events required for any processing task. Since the activities of the CPU are cyclical, the clock furnishes the reference for all processor actions.

**Microprocessor.** A microprocessor is defined as a central processing unit (CPU), fabricated as *one* integrated circuit (single chip), that executes program instructions. Some forms of microprocessors use more than one chip. Remember

from the definition—a microprocessor is a CPU; but, a CPU does not have to be a microprocessor. There are a wide variety of microprocessors available. Each one has its own distinguishing characteristics. Some rather general differences which distinguish the various microprocessors are the following:

- a. Clock speeds.
- b. Instruction set.
- c. Data bus size.
- d. Hardware requirements.
- e. Symbology.

**Clock speeds.** The discussion of the CPU's control circuitry mentioned a master-timing generator, or clock. Microprocessors can have various clock speeds and are usually capable of certain minimum and maximum speeds.

**Instruction set.** Each type of CPU made generally has its own instruction set, although some instruction sets may be compatible between different microprocessors. The CPU is designed so that a specific operation is performed when the CPU control logic decodes a particular instruction. The instruction set is a "table" showing what each combination of binary signals do. Some instruction sets are deliberately designed to accomplish certain tasks better than others.

**Data bus size.** The choice of data bus size depends on the intended application of the computer. Microprocessors may use 8-, 16-, or 32-bit data buses. Some microprocessors may use even larger data buses.

**Hardware requirements.** Each type of CPU is unique in its hardware construction and requirements. Some CPUs produce special signals during I/O operations. Other CPUs require different voltages for the chips. Various CPUs' internal operations may even differ because of their design differences.

**Electronic symbols.** Figure 2-2 shows the symbol for a microprocessor with the pins labeled to indicate the signals present at each pin. Symbols vary according to the different types of microprocessors.

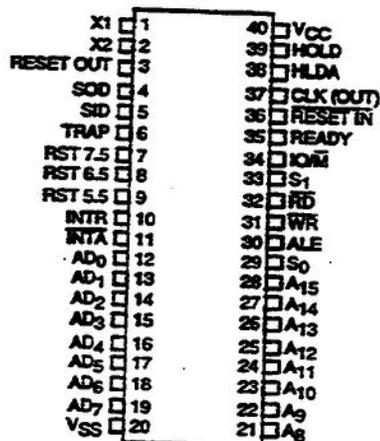


Figure 2-2. Symbolized microprocessor.

## Self-Test Questions

After you complete these questions, you may check your answers at the end of the unit.

### 608. Computer classifications and terminology

1. Who invented the analytical machine?
2. What are the three generations of computers?
3. What predecessor of the present-day computer was the first all-electronic machine, performed like a calculator, and could not store data?
4. During what year were transistors first used in the construction of computers?
5. What is considered the first true computer?
6. When did the first small computer become commercially available?
7. Define the term "mainframe."
8. What distinguishes minicomputers from mainframes?
9. Briefly describe the relationship between "data" and "program."
10. Why is programming called meticulous work?
11. What is the technical language of the computer profession?

12. Match the descriptions in Column A with the computer terms in Column B. Column B items may be used only once.

*Column A*

- \_\_\_(1) Often referred to as "software."
- \_\_\_(2) A single one or zero, stored in a flip-flop.
- \_\_\_(3) A master timing generator within the computer.
- \_\_\_(4) Generally refers to a unit of eight bits of data.
- \_\_\_(5) Refers to a group of conductors used to transfer signals.
- \_\_\_(6) A sequence of digits that tells the computer what operation to perform.
- \_\_\_(7) Ultimately depends on the design of the computer, but generally is made of more than one byte.
- \_\_\_(8) A label, symbol, or other set of characters used to designate a location or register where data is stored.
- \_\_\_(9) Used as a general term, it can denote numbers, letters, symbols, etc., that can be reproduced or processed by a computer.
- \_\_\_(10) Mechanical, magnetic, electrical, and electronic components of a computer.

*Column B*

- a. Bit.
- b. Data.
- c. Byte.
- d. Word.
- e. Instruction.
- f. Program.
- g. Clock.
- h. Bus.
- i. Address.
- j. Hardware.

13. What are the three principal buses in a computer?

**609. Internal circuit functions**

1. What are the two general classes of a computer's basic components?
2. What component within a computer translates information to be processed into computer language?
3. What controls the input/output (I/O) section of a computer?

4. Match the descriptions in Column A with the information pertaining to a computer's basic functional operation (or block diagram) in Column B. Column B items may be used only once.

*Column A*

- \_\_\_\_ (1) The functional block of a computer block diagram that applies to data which can be provided from punched paper tape, magnetic tape or disk, a modem, light pen, a mouse, a touch-sensitive video terminal, or even a speech recognition system.
- \_\_\_\_ (2) The functional block of a computer block diagram that applies to data which can be displayed on a video screen, recorded on magnetic tape or disk, or sent to a printer or modem. It can also produce the proper response through a recorded message of voice synthesis.
- \_\_\_\_ (3) The section of a computer which provides communications capability between the computer and a peripheral device.
- \_\_\_\_ (4) The electronic circuit which connects the peripheral device to the computer's I/O system.
- \_\_\_\_ (5) The section of a computer that is used for storing information.
- \_\_\_\_ (6) The section of a computer that can simulate subtraction, multiplication, and division.
- \_\_\_\_ (7) The section of a computer that decodes instructions.

*Column B*

- a. Arithmetic logic unit (ALU).  
 b. Control unit.  
 c. Input/output.  
 d. An interface.  
 e. Output.  
 f. Memory.  
 g. Input.

### 610. Internal control functions

1. What are the four functional units of a computer?
2. What is the "brain" of a computer system?

3. What must a central processing unit (CPU) be able to do?
4. Why are interrupt requests important?
5. What are the three functional units within a central processing unit?
6. What components within a central processing unit serve as temporary storage units?
7. What is the function of the program counter?
8. What happens to an instruction once its memory location is identified?
9. Define the flag register's operation.
10. What is the function of a computer's control unit?
11. Where does the control unit of a computer obtain the instructions it needs to tell the central processing unit the exact steps to follow in the processing of data?
12. What functions does a computer's control sequencer perform?
13. What does the control circuitry of a computer use to maintain the proper sequence of events required for any processing task?
14. What are some of the general differences that distinguish various microprocessors?

## 2-2. Types of storage and devices

For a computer to function efficiently, it must have some alternate means of storing and retrieving program instructions and data other than its primary storage. There are various types of auxiliary storage devices in use today ranging from fixed subsystems installed in the computer to portable devices such as magnetic disks and tapes. For a computer to make use of these devices, it is interfaced with a disk controller. In this section, we discuss various types of storage media and how the computer makes use of them.

### 611. Disk subsystems

On most computer systems, the disk subsystem (fig. 2-3) is composed of a disk controller (storage controller), disk unit, and disk pack. These components sometimes have different names. Let's look at some information about components common to all computers.

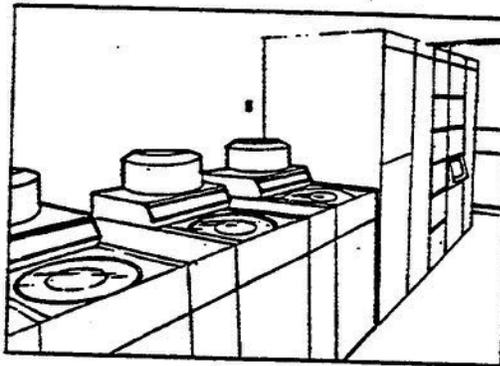


Figure 2-3. Magnetic disk subsystem.

**Disk controller.** Most disk controllers serve five basic functions:

- (1) Interpret and execute commands from the channel attached to the CPU.
- (2) Provide a primary and alternate path for data between the I/O interface and the attached storage devices.
- (3) Translate data appropriately as it is transferred between the storage devices and the I/O interface.
- (4) Furnish operation status information to the channel.
- (5) Check the accuracy of data transfer, parity, or packet.

Your disk controller may serve other functions, such as providing a direct means of influencing the disk subsystem. For this reason, it is important for you to become thoroughly acquainted with your model.

**Disk unit.** Basically, there are two variations of disk units: those with fixed read/write heads and those with movable read/write heads. Most disk units use movable read/write heads. A movable head is one read/write head per platter on an arm that must "float" to the correct track before reading or writing to it. The

access time on a fixed head unit is much faster than that on a movable head unit. Access time is lost on the movable head unit when it searches for the correct track the data is stored on. In later systems this time was negligible, so a type of movable head unit, called removable disk packs, became more widely used. These removable disk packs are not used much nowadays due to inherent problems with the disk packs becoming contaminated and causing the head to crash. Instead, non-removable disk packs with factory-sealed head disk assemblies (HDAs) containing movable read/write heads are being used.

Since the disk is in constant motion, the heads and medium must be kept separated a finite distance. The head usually travels on a small cushion of air. This is why it is important to maintain a dust/dirt-free environment. A small piece of dirt or a dust ball in the air cushion space causes what is commonly called a head crash. This results in damaged disk packs and, possibly, a damaged disk unit, or both.

Most disk units have an operator's control panel through which the operator controls the unit. Since there are many different disk units and the names may vary, you should become thoroughly familiar with the manufacturer's reference manual before you operate any disk unit. Now let's look at the most popular disk storage device, the hard disk.

## 612. Hard disk

The hard disk, or hard drive, is an aluminum-core metal oxide coated computer hardware device that contains a magnetized surface used to store programs and information. Although the hard disk can be installed as an external device, it is often installed in a computer's central processing cabinet. In *microcomputers*, the hard disk is typically called *Drive C*. The hard drive is a must-have storage device for today's large programs. As technology advancements continue, hard drives will be available in a wider range of sizes and capacities.

Hard disks store anywhere from a few megabytes of information to several hundred megabytes of information, depending on how they are manufactured. They are very durable because they are completely enclosed in air-tight protective cases to prevent contaminants from entering and damaging the surface of the disk.

A hard disk must be configured correctly in order to effectively store programs and information that can later be retrieved and processed. The four steps to configuring disk storage include installing, formatting, partitioning, and repartitioning. Before discussing these steps, we first need to understand the files system concept.

**Files system concept.** A file system is an arrangement of files stored on a segment of a hard or flexible (floppy) disk. A disk is divided into smaller, more manageable sections called *blocks*. These blocks are sized as 512, 1024, or 2048 bytes in capacity and are defined when the disk is installed. The blocks are

numbered beginning with zero up to the number of blocks that can fit on the disk or portion of the disk. A file system is made up of these blocks.

Four different types of blocks make up a file system. The file system's first block is called Block 0, which is the *boot* block, and contains the information needed to load, or boot, the file system. Block 0 resides only on the disk's boot partition.

The second block is called Block 1, or *superblock*, and is an index to the file system that contains the file system's size and the size of the next two types of blocks: the *i-list* and *data* blocks. The *i-list* is a variable number of blocks that contain *i-nodes*. An *i-node* is actually a table of contents that contains information about a file. The most important piece of information the *i-node* contains about a file is the disk address. The disk address tells the computer system the actual physical locations of a file on the disk. Each file on a disk has one related *i-node*. If you have a very large file, the *i-node* is used as a pointer to data blocks that serve as indexes for the true data blocks.

The rest of the file system blocks are allotted as data blocks and are used to store the actual contents of the particular file, except for the index function they provide in the case of very large files. A *directory* on a disk drive serves as an index to an index. Let's look at an example of how this file system works (fig. 2-4).

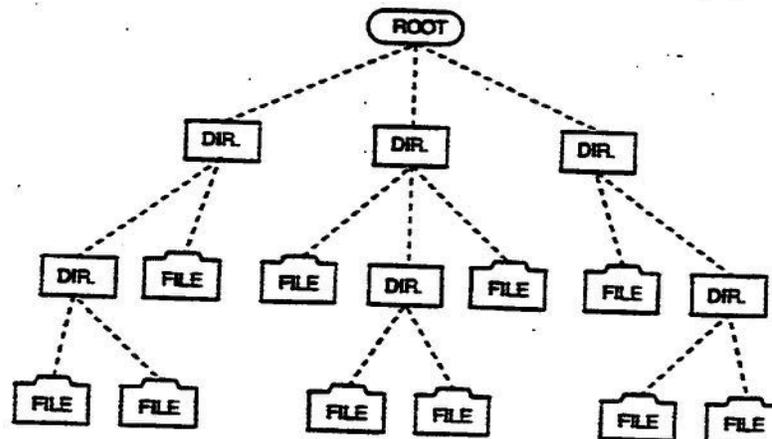


Figure 2-4. File system block diagram.

Say, for example, you want to display a file called "work." The first thing the system does is look in the directory for the particular filename. The directory contains the number of the *i-node* corresponding to this file. The directory informs the system to look at the *i-node* corresponding to the filename. For this example we say it is *i-node* 519. The system goes to *i-node* 519 of the *i-list* and reads the data block addresses for the file. The system then goes to the data block addresses (where the file contents actually reside) and retrieves the contents from the various locations. Finally, the system arranges the contents in the correct order and displays them on your computer's video screen.

You can easily understand how important and critical it is to configure disks so as to properly store and retrieve the information files. Let's look at the first step in configuring the hard drive.

**Installation.** We are not going to describe the step-by-step procedures for installing hard drives as this is normally a maintenance function. Suffice to say that if your system runs out of space on existing hard disks, installation of another disk drive is warranted. Most computer systems contain installation instructions in permanent memory that guide you through the installation procedures.

**Formatting.** Before a computer system can utilize a new hard disk, you must format the disk. Just as a surveyor must map out the property boundaries before a new neighborhood subdivision is built, a computer's disk operating system must map an addressing scheme on the magnetic surface of the disk. The system divides the disk into sections called tracks and sectors that the disk controller can access. Tracks run around the disk in a circular direction. Sectors divide the disk into sections much like pie-slices.

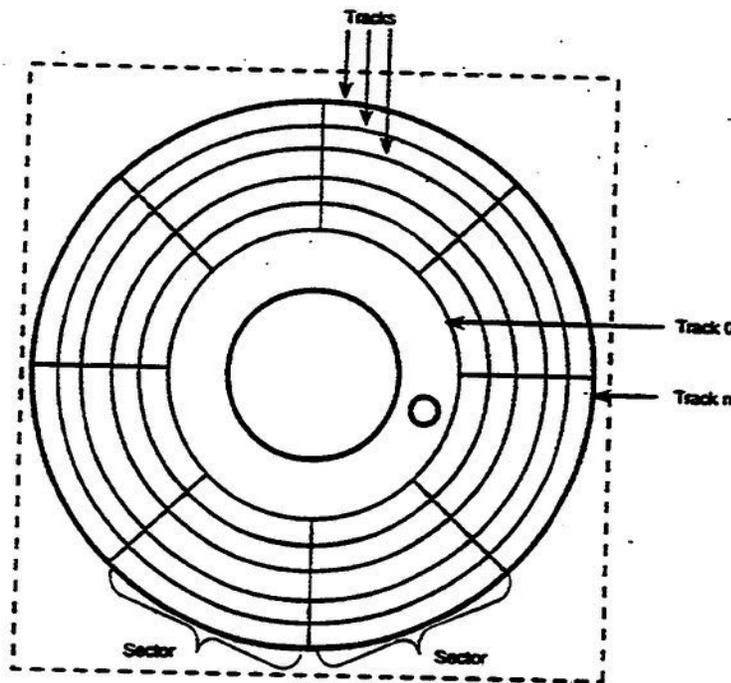


Figure 2-5. Disk formatting.

**Partitioning.** In addition to formatting disks, the disks must also be partitioned. Partitioning determines the block size (512, 1024, or 2048 bytes), the file systems that will reside on the disk, the number of blocks for each file system, and the number of i-nodes assigned. This is like saying the subdivision planners will designate one part of the neighborhood for a country club, another for the homes,

a third for a park, etc. A total of 16 partitions are possible; however, it is rare to use all 16.

Partitioning a disk is risky business. If you make a mistake and choose an incorrect number of blocks for a file system, you're headed for trouble. In addition, you must give careful consideration to block size. First, you determine the average size of files in the file system. If a file system generally consists of small files, you fix the block size at 512 bytes. Likewise, if a file system generally consists of large files, then make the block size 2048 bytes. The default block size is 1024 bytes. The decision you make on block size determines how efficiently your computer system uses the disk space and processing time.

Fortunately, the procedures for performing disk partitioning is beyond the scope of this course. Even if the initial partitioning works and the hard disk is up and running, new situations may develop that may require you to change the configuration.

**Repartitioning.** Repartitioning a hard drive allows you to make changes to the current partition setup and requires more steps than initial partitioning. You first need to off-load all information and then decide how you want to change the size of your file systems and if you need to add more. To enlarge the size of one partition, you must reduce the size of another. As applied to our subdivision example, this would be like tearing down four homes in order to enlarge the area of the park.

**Optimizing.** Another area of disk management includes making the most efficient use of available storage space. Periodically, you need to inspect the disk and remove unnecessary files. You also need to use some type of disk maintenance utility to clear up any checkerboarded areas and fragmented files.

Checkerboarding is a condition that exists when the disk has unused or blank areas scattered throughout the different storage areas. Fragmented files are those that have portions of information scattered throughout the file system storage area. Optimizing the disk regroups scattered information into logical areas and defragments the fragmented files.

Proper hard disk management ensures an efficient computer system. When configured correctly, the system easily retrieves files from storage, the disk contains enough room for data storage, and the amount of time it takes to access the disk is low.

### 613. Portable disk systems

Hard disks are not intended to be used to store infinite amounts of data for indefinite periods of time. Data to be maintained in this fashion is either created or stored on auxiliary storage devices. Also, there is a great need to have the ability to transport data without the benefit of a network environment for electronic transmission. Transportable auxiliary storage devices satisfy this need. There are

several of these types of devices available to us, including magnetic and optical disks.

**Magnetic disk packs.** Magnetic disk packs have become a popular input-output medium because of their direct access and large capacity. With direct access, a record can be retrieved without sequential searching through a file or a series of files; however, data can be stored or accessed randomly or sequentially. Disk packs are removable and interchangeable. They have the flexibility to meet present needs and, yet, allow for growth.

The magnetic disk is a thin disk of metal coated on both sides with magnetic recording material. Data is stored in the form of magnetic spots in concentric tracks residing on each surface of the disk. These tracks are accessible for reading by positioning read/write heads between the spinning disks.

Disks permit the immediate access to specific areas of information without the need to examine each record, as in magnetic tape operations. Independent portable disks can be used with interchangeable disk packs. The disk packs are mounted on a single unit which can be readily removed from the disk drive and stored in a library of disk packs.

Read/write heads are mounted on an access arm and arranged like teeth on a comb that moves horizontally between the disks. Two read/write heads are mounted on each arm with one head servicing the bottom surface and the other head servicing the top surface. Thus, it is possible to read or write on either side of the disk.

With proper file organization, minimum access time is required for the retrieval of a disk record. The concept of removable disk packs means that only those disk records required for a particular application need be in use. Data records for other applications can be removed and stored.

Each pack is divided into cylinders. A cylinder of data is that amount of information that is accessible with one position of the access mechanism. Since the movement of the access mechanism requires a significant portion of the time needed to access and transfer data, the storing of a large amount of data in a single cylinder can save time in processing by minimizing the amount of access time.

**Flexible disks.** Also known as floppy disks, flexible disks are among the newest enhancements of computer systems in both performance and price. A disk coated with a metallic oxide is housed in a plastic jacket that has openings for the drive hub, read/write head, and position sensing. Flexible disks come in various sizes and sectioning methods. The type used on your computer may vary with manufacturer. Figure 2-6 shows the parts of a flexible disk.

Floppy disks are used to transfer and store information that can be retrieved by the computer system. They are inserted into the computer through a floppy disk drive that is normally installed in the computer chassis, although it may be attached as an external device. The opening of the drive corresponds to the width of the disks used. The two sizes available are the 3.5 inch and the 5.25 inch. The 3.5 inch disk

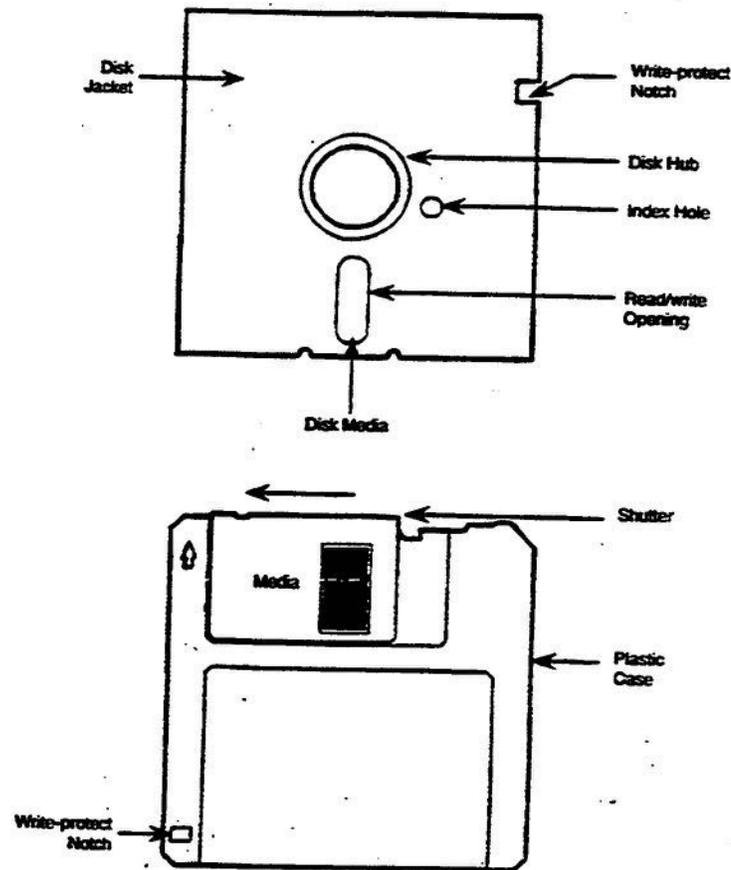


Figure 2-6. Flexible disks.

has a low-density storage capacity of 720 kilobytes or high-density capacity of 1.44 megabytes. The 5.25 inch has a low-density capacity of 360 kilobytes, a double-density capacity of 720 kilobytes, or a high-density capacity of 1.2 megabytes.

Both types can be physically write protected to prevent loss of data. This is accomplished on 5.25 inch disks by placing a tab over a notch in the protective jacket and on 3.5 inch disks by positioning a write-protect switch built into the plastic case to the open position.

Flexible disks are able to withstand the stress of repeated use; however, you need to take certain precautions to prevent accidental damage (fig. 2-7). Some of these precautions are the following:

- **Bent diskettes.** Diskettes are flexible which enables them to rotate freely within the disk drives. Bending them reduces the disks' rotating ability.
- **Creased diskettes.** Folding a disk or placing heavy objects on it can cause a permanent crease that ruins the disk.
- **Warped diskettes.** Store disks in temperature controlled conditions of 50-120° Fahrenheit and a range of 20-80% relative humidity. Exposing a

diskette outside these ranges (such as in direct sunlight in your automobile) can cause the disk to warp and render it useless.

- **Dented diskettes.** Pressure from ball-point pens or pencils can dent the recording surface of the disk. Therefore, label disks with a felt tip pen or fill out the labels before you attach them to the disk jacket.
- **Contamination of diskettes.** Foreign substances, such as fingerprints, smoke, moisture, and dust, can accumulate on the recording surface of a disk and cause contamination. When these substances accumulate on the surface of a disk, a "head crash" may occur. A head crash occurs when the surface of a disk comes in contact with the drive read/write heads and may destroy data on the disk, the disk itself, or the disk drive.

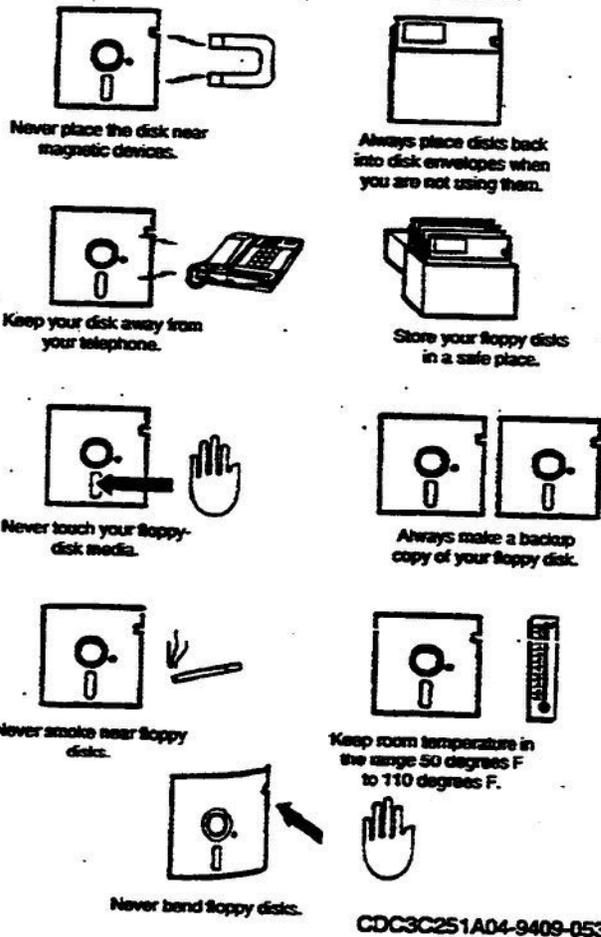


Figure 2-7. Care of floppy disks.

**Disk cartridges.** Cartridges are the next step up in disk systems. The disk cartridge is a removable disk mounted in a protective cartridge. A cartridge drive may support only the removable disk, or it may include one or more fixed disks as well as the removable one. Disk cartridges come in both the magnetic style and the optical. Removable hard disk systems are not used much in mainframe

computer systems anymore, but smaller versions are still used in tactical (theater) applications. Figure 2-8 shows a disk cartridge in use.

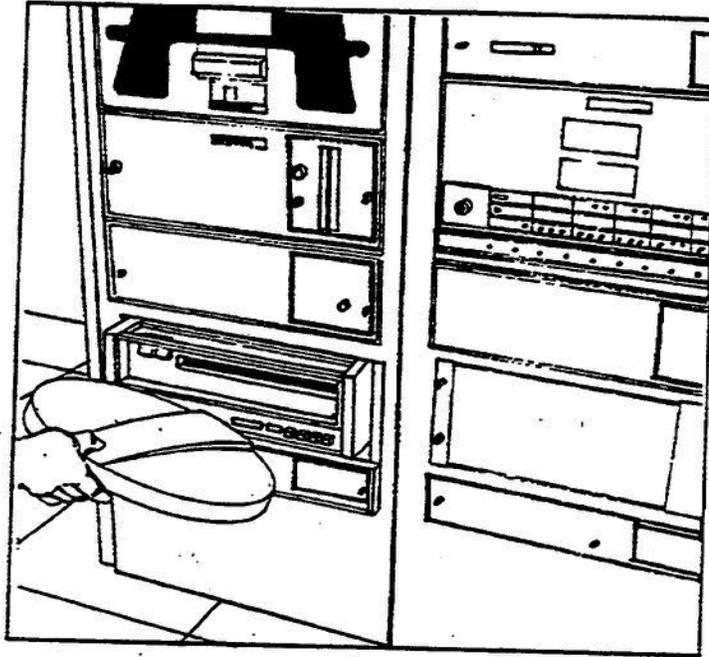


Figure 2-8. Disk cartridge.

**Optical disks.** These disks (fig. 2-9) are becoming less expensive to operate and, thus, are being incorporated into more and more new systems today. Optical disk units use a laser light to read pits and levels on the disk surface rather than magnetic spots as do magnetic disk units. Optical disks come in three types: compact disk-read only memory (CD-ROM), write once, read many (WORM) and fully rewritable units. Optical technology will be the next step in computer systems since speed and density can be increased to the gigabyte level. One of the most promising and successful types is the CD-ROM.



Figure 2-9. Optical disk.

**CD-ROM.** Like the very popular audio CDs, CD-ROM for computer use is becoming more wide spread (fig. 2-10). The disks are rigid, durable and reliable. CD-ROMs store information by means of laser technology. Using laser optics, CD-ROMs store data optically, not magnetically, and have proven to be a reliable storage medium. There is one drawback to using CD-ROM, however, and that is

they can only be used to retrieve data previously stored on them, hence the read-only memory. CD-ROMs are used primarily to archive important data that is not expected to change over long periods of time.

The greatest advantage of CD-ROMs is their storage capacity as compared to other type storage devices. A single disk can store hundreds of megabytes of data, more than enough to store an entire set of encyclopedias!

**WORM.** WORM disks allow you to write data on them once, but only once. After that, the disk behaves just like a CD-ROM.

**Others.** The third type, erasable optical or floptical, can be erased and loaded with new data just like magnetic disks. This technology is still quite new and relatively expensive when compared to other disk storage types.

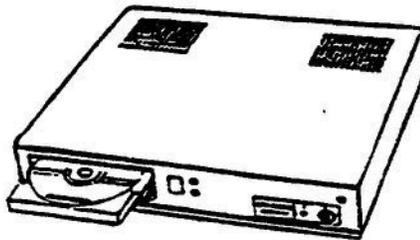


Figure 2-10. CD-ROM device.

#### 614. Magnetic tape systems

Magnetic tape devices (fig. 2-11) are input and output devices. They record and retrieve information by feeding magnetic tape through a read/write head. The read/write head reads information stored in the form of magnetized spots and writes information by magnetizing areas in parallel tracks along the length of the tape. The writing on magnetic tape is destructive; that is, the new information erases the old information.

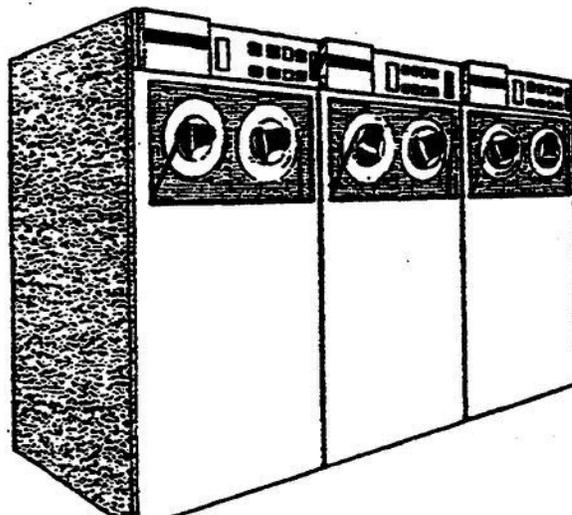


Figure 2-11. Magnetic tape units.

Magnetic tape records are not restricted to any fixed record size, words, or blocks. Blocks of records (which may be a single record or several records) are separated on the tape by an interblock gap, a length of blank tape averaging about 0.6 inches to 0.75 inches. This interblock gap is automatically produced at the time of the writing on the tape and provides the necessary time for starting and stopping the tape between blocks of records. Blocks of records are read into or out of buffer units.

**Magnetic tape unit.** The magnetic tape subsystem serves a multiple purpose—it acts as an auxiliary storage device, input device, and output device. As the Air Force inventory of computer (magnetic) tapes continues to grow at a rapid pace, you, as a systems controller, deal with them more and more. For this reason, it is helpful to understand the magnetic tape unit and how it works. Magnetic tape units vary, not only from one computer system to another, but also with models within a system. Still, there are some basic principles that remain the same. Every magnetic tape unit has a *tape controller* and *associated tape components*.

**Tape controllers.** Tape controllers may be hardware, software, or both. In any case, a tape controller serves as a controlling monitor of the tape units. Some common controller functions include the following:

- a. Monitoring for transfer errors between the input/output (I/O) controller and tape units.
- b. Monitoring parity errors on tape units.
- c. Allowing an operator to re-designate a particular unit's logic identifier.
- d. Allowing an operator to backspace or advance one record or more on the tape.

Your magnetic tape controller model probably allows you to do much more; therefore, it is very important that you study the manual that comes with it.

**Magnetic tape components.** Magnetic tape components also vary from one model to another. We'll discuss some areas common to all units. Magnetic tape units are composed of a tape transport system, read/write area, and an operator's control panel. The tape transport is much like an audio tape recorder. While most home tape recorders operate with a tape speed of 7.5 inches per second, magnetic tape transports operate at tape speeds above 100 inches per second. Since tape reels have hundreds of feet of tape, they are too heavy to accelerate to the proper operating speed in a short time. A tape transport accelerates only a small portion of the tape to the required speed during operation. A separate drive, which accelerates at a slower speed, drives the tape reels.

The most common type of tape drive system uses drive capstans to drive the tape (fig. 2-12). The drive capstan is always rotating at the proper speed. The tape passes between the drive capstan and a pinch roller. To accelerate the tape, it is only necessary to move the pinch roller a short distance to force the tape against the drive capstan.

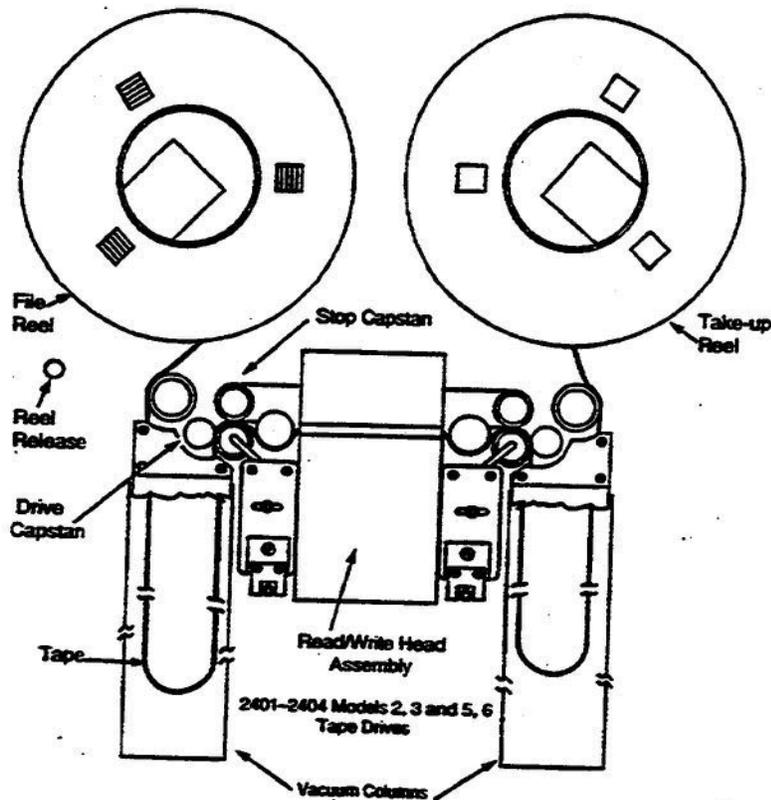


Figure 2-12. Magnetic tape unit assembly.

In a tape unit, the computer may start and stop the tape movement at specific intervals and at high frequency. The main problem is bringing a tape up from rest to high speed as quickly as possible and returning it from high speed to rest as quickly as possible. To do this, the tape unit uses a buffering arrangement.

Buffers depend on a tape reservoir system in which a short length of tape is always available for acceleration. Two reservoirs are commonly used. One is used with the supply reel and one with the take-up reel. During the tape start, the feed reservoir must be able to supply tape fast enough to prevent breaking the tape while the take-up reel is attaining speed. The most common method of buffering uses vacuum chambers.

In the vacuum chamber, magnetic tape is passed through a low-pressure chamber between the reel and the tape drive mechanism. Higher air pressure outside the chamber forces the tape into a loop where the length is sensed to control the reel motor. For example, if the loop in the feed chamber becomes too small, the motor starts to feed more tape; when the loop is too large, the motor stops.

The read/write mechanism (fig. 2-13), or head assembly, consists of read heads, write heads, erase heads, tape cleaners, and a pneumatic pad. The head assembly contains seven, nine, or more write heads and an equal number of read heads mounted side by side so that tape first passes under the write heads and then under

the read heads to detect any errors before the computer takes any action. Each read and write head consists of an individual wire-wound core.

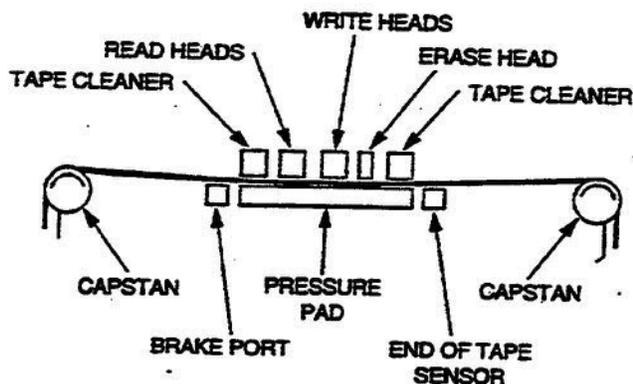


Figure 2-13. Head assembly.

As current flows through the windings in a write head, a magnetic field is induced across a write gap at the bottom of the head. This causes information to be written on the tape. As the tape passes under this write head, the magnetic particles on the tape are aligned by the head in the same direction as in the head wire. When current flow through the coil winding reverses, the magnetic particles on the tape are realigned in the opposite direction. As the tape passes under the gaps on the read heads, a read operation senses the alignment of the magnetic particles on the tape, amplifies them, and sends them to the control logic section for processing.

To write to tape, the computer issues a write command, which causes the tape station to start moving tape forward (from the supply reel to the take-up reel). After a short pause to let the tape reach full speed, input data is written to the tape in the form of magnetized spots. After the entire block is recorded (fig. 2-14), the computer sends a stop command to stop the tape. Subsequent data blocks are written by this same process. The blank space between the blocks is called a gap.

To read from tape, the computer issues a read command, and, again, the tape begins to move forward. If a reading or mechanical error occurs, either a printout is received on the monitor printer or an indicator lights on the operator console of the computer and an alarm sounds.

Erase heads have a steady DC current flowing through them during a write operation. This aligns all particles on the tape in the same direction before the tape passes under the write heads. Therefore, all information is erased from the tape before new information is written. During a read operation, the erase heads are not supplied current, and consequently, the prerecorded information is not destroyed.

The tape cleaner contacts the tape and removes foreign particles from the tape before recording. The scraper blades dislodge any foreign material on the tape. The particles are then removed from the head area by vacuum. A pneumatic pad maintains precise contact pressure between the tape and the head gaps. This

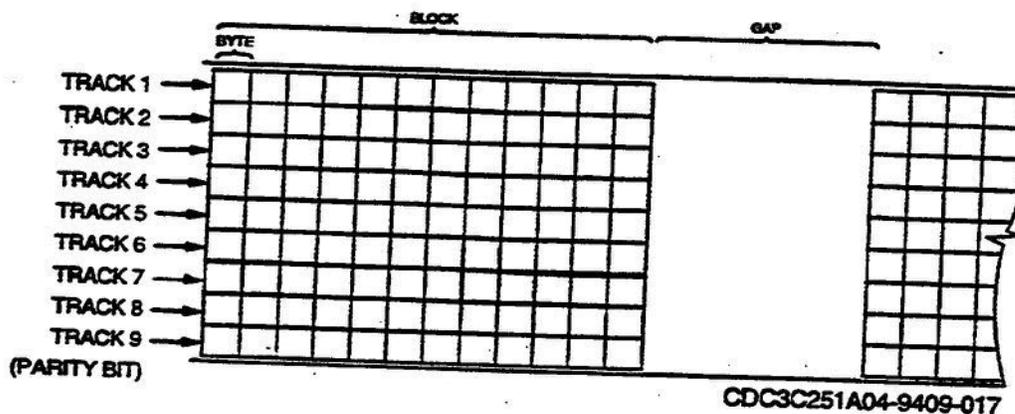


Figure 2-14. Magnetic spots and gap.

contact pressure is provided by means of air pressure, which minimizes head and tape wear by blowing the tape against the heads.

The control logic section consists of the circuit boards and other circuits that control read and write operations of the unit and detect error conditions. A power supply provides the AC and DC power for the tape station. The motors need AC power, and the control logic boards need DC power.

**Cartridge recorders.** There are many cartridge systems available; however, they usually are not interchangeable. A cartridge (fig. 2-15) contains about 300 feet of 1/4-inch tape with a storage capacity of over 2.5 million bytes. They are quite useful where higher performance is required and sequential access is still acceptable. Cartridge recorders are used primarily to store back-up information for systems and database files.



Figure 2-15. Magnetic tape cartridge.

## 615. Paper tape systems

Data from main storage can be converted to a tape code and punched into a blank paper tape as the tape moves past the punch mechanism. The paper-tape reader

reads the punched holes as the tape moves past the reading unit. Paper tape may also be punched as a by-product of a cash register or some other device.

Paper tape devices have been used as an I/O device on computer systems for many years, but they're not used much today on modern computer systems. Paper tape punches and readers are used to input and output on many different types of systems. Some devices house the punch and the reader in the same cabinet, while others are housed separately.

**Tape punch.** The punch is an output device used as auxiliary storage or data preparation for input.

**Tape reader.** The reader is the input device for data prepared on the punch. The reader functions much as a magnetic tape unit functions in the read mode. It has the same components except in the read/write area. The paper tape reader cannot change any information that has previously been punched on the tape. A new tape has to be made if data is to be changed or sequenced.

---

### Self-Test Questions

After you complete these questions, you may check your answers at the end of the unit.

#### 611. Disk subsystems

1. What three components comprise an average disk subsystem?
2. Which part of a disk subsystem translates data as it is transferred between storage devices and the input/output interface?
3. Which type of disk unit has a faster access time—those with fixed heads or those with movable heads?

#### 612. Hard disks

1. Are hard disks internal or external storage devices?
2. What is the common identification normally given to a microcomputer's internally installed hard drive?

3. What makes hard disks durable?
4. What are the four steps for configuring disk storage space?
5. What is a file system, as it relates to computer storage?
6. What makes up a file system?
7. Match the descriptions in Column A with their corresponding terms in Column B. Column B items may be used once or more than once.

*Column A*

- \_\_\_(1) It is the boot block.
- \_\_\_(2) Serves as an index to an index.
- \_\_\_(3) Tells the computer the physical location of files on a disk.
- \_\_\_(4) An index to a files system.
- \_\_\_(5) It is the superblock.
- \_\_\_(6) A table of contents containing file information.
- \_\_\_(7) A variable number of blocks containing i-nodes.
- \_\_\_(8) Contains disk address information.
- \_\_\_(9) Contains a files system's size information.
- \_\_\_(10) Used to store the contents of specific files.
- \_\_\_(11) Contains the information needed to load a file system.

*Column B*

- a. Block 0.
- b. Block 1.
- c. i-list.
- d. i-nodes.
- e. Disk address.
- f. Data blocks.
- g. Directory.

8. What is the first step in configuring a new hard disk?
9. What must be done to a newly installed hard disk before the computer can use it?
10. What changes does formatting make to a hard disk?

11. What configuration factors are established when a hard disk is partitioned?
12. What impact does determining the correct block size for a hard disk have on a computer's efficiency?
13. How do you change a hard disk's configuration once it has been partitioned?
14. How does enlarging one partition of a hard disk impact other existing partitions?

### **613. Portable disk systems**

1. How does the data access method used with magnetic disk systems differ from that used with magnetic tape systems?
2. What advantage do portable disk packs have over fixed disks?
3. Are floppy disk drives installed as internal or external storage devices?
4. What is the storage capacity of a 3.5 inch floppy disk?
5. What is the storage capacity of a 5.25 inch floppy disk?
6. What can happen if you allow a floppy disk to become contaminated with foreign substances?
7. What is a disk cartridge?

8. How is data read from an optical disk as opposed to a magnetic disk?
9. Give the three types of optical disks.
10. What is one disadvantage of compact disk-read only memory (CD-ROM)?
11. For what purpose are CD-ROMs primarily used?
12. What is the greatest advantage of CD-ROMs over other types of storage devices?
13. How do write-once read-many (WORM) disks differ from CD-ROMs?
14. How do floptical disks differ from CD-ROMs and WORM disks?

#### **614. Magnetic tape systems**

1. How do magnetic tape systems record and retrieve data?
2. How are blocks of records separated on magnetic tape?
3. What purposes do magnetic tape units serve?
4. What common controller functions are provided by magnetic tape controllers?
5. What are the components of a magnetic tape unit?

6. What part of a tape drive system commonly drives the magnetic tape?
7. How many tape reservoirs are commonly used on a tape unit?
8. What are the components of a magnetic tape head assembly?
9. What are two ways that the computer notifies an operator of a mechanical or read error?
10. How do tape units erase data from a magnetic tape?
11. What prevents prerecorded information from being destroyed during a magnetic tape read operation?

### **615. Paper tape systems**

1. Describe the principle behind using paper tape systems as data storage devices.
2. What part of a paper tape device is used for computer output?
3. Can a paper tape reader alter the information on a previously punched tape?

### **2-3. Peripheral Devices**

Peripheral devices are used to input data into the computer and output information in a usable form. Peripherals may operate either under the control of the CPU or as separate, stand-alone devices. Devices under control of the CPU are referred to

as *on-line*; devices not under CPU control are considered *off-line*. Many devices can operate in either mode, with the mode being switchable. For example, some terminals can be switched off-line and used to input data and record on tape and, then, switched back on-line to read the data from the tape into the computer.

Since the computer is an electronic device and input/output units are primarily electromechanical devices, the computer can operate much faster. To enable the computer to operate as nearly as possible at full capacity, the transfer of data between I/O devices and main storage usually takes place independently through an intermediary known as a channel.

The I/O channel and associated control and connecting units provide for the buffering (temporary storage), coordination, and transfer of data. This allows the CPU to work without communicating directly with I/O devices, which in turn, allows I/O operations to proceed while data is processed. In other words, the computer can carry on high-speed computations while input data is being received and while output data is being transferred.

The channel that performs this function may be an independent unit complete with necessary logic and storage capabilities (in effect, a small special purpose computer capable of performing only I/O operations) or it may share CPU facilities and be physically integrated with the CPU. In some cases a channel is shared by several low-speed devices such as readers, punches and terminals. In other cases, a channel may accommodate higher data rates and be limited to only one data transfer operation at a time.

Most I/O devices are automatic; once started, they continue to operate as the stored program directs. These devices can transmit data or receive data from the main memory section of the CPU.

I/O devices may be used for both input and output or for just one of these functions. More than one form of input and output may be used, and different forms may be combined. For example, punched tape may be used as input and a printer as output. The specific form of input and output depends on the configuration of your computer system and the functions it is designed to perform. We will consider each of the standard I/O devices separately.

## 616. Consoles and terminals

Some large computers are controlled by an operator from a system console like the one in figure 2-16. The operating system, diagnostic routines, and other functions may be loaded or performed via the console. A system console can include, but is not limited to the following: a keyboard, a monitor, a printer, dials, switches, and alarm lights. The primary function of the console is to provide a direct communication between the operator and the operating system.



Figure 2-16. Computer console.

**Consoles.** A console is a unit used by an operator for all manual communications with the computer. It also provides a display from the computer. The configuration of the console arrangement differs with the computer size and model used. In some large computer systems, a line printer is used for printing information to speed up the overall performance of the system. These system "logs" contain all system activities, including records of any operator intervention and information generated by users. Consoles are becoming predominant in most Air Force systems as they provide the necessary audit trails for security and historical purposes.

Large computer systems are usually equipped with a system console that has two video units and one keyboard; however, these devices may be more numerous, with several display consoles being used for controlling independent programs simultaneously.

The unique thing about consoles is that they are designed for one computer system. Consoles are not normally for processing data. Since they are connected directly to the computer's system board, the operator has final control over the system hardware. No user of the system has higher access or more control than the operator console area.

**Terminals.** Terminals, such as the one in figure 2-17, have both input and output capabilities and provide a means to communicate with the computer in an interactive or conversational manner. The input device is usually a typewriter-like keyboard, but other keyboard formats and input techniques are available, including numeric key pads and predefined or programmable function keys.



Figure 2-17. Computer terminal.

Numeric key pads have the same key layout as adding machines. Experienced operators can enter numeric data much faster using a numeric pad. Programmable function keys are associated with the functions that have been programmed and stored in the computer. When a function key is depressed, its associated operation is performed. Some terminals include combinations of two or all three of these keyboard formats. Figure 2-18 (page 37) is an example of a keyboard layout with all three formats. Your keyboard layout may be slightly different. Most personal computer keyboards contain either 84 keys (including 10 function keys) or 101 keys (including 12 function keys).

Terminal output capabilities include alarms, lights, and video. The *visual display unit* (VDU) is one of the most diversified units in the computer field. A VDU can display data in the form of characters and pictures in both monochrome (one color) or full color. The type of VDU that you use is based on the capabilities of your computer system and your data processing needs.

A visual display unit is a tool used to transfer information to and from a computer. Inputting information to this unit can be accomplished by using a keyboard, mouse, digital pad, or light pen. Before the data is released to the computer, the operator can edit or erase an entire input and reenter it. Output information is written on a cleared display or added to an existing display under operator or program control. Once information is displayed, it is available for as long as needed.

The VDU screen is divided into tiny squares called "pixels." The term, pixel, is derived from the words "picture element" and refers to the smallest addressable area on the monitor screen. It's the pattern generated by the pixels which are turned on that make up the picture or display. Each pixel is controlled by a small portion of the computer's random access memory (RAM). The information for the picture is stored in binary code in the computer's memory and is very easy to alter. For instance, you can change the color or shape of a picture and enlarge, reduce, stretch, or rotate it. By storing the picture on a disk or tape, you can get exactly the same picture every time.

The quality of the picture depends on how many pixels there are in the screen (resolution) and how much memory the computer has, since each pixel needs its own little bit of memory. Realistic looking pictures are made up of many thousands of tiny pixels. Larger pixels need less computer memory. Types of terminals used in conjunction with today's computers are referred to as "smart terminals" and "dumb terminals."

**Dumb terminals.** At one time, all terminals connected to a computer system were of this type. A dumb terminal takes all of its capability from the CPU. Another term once used for this kind of terminal was SLAVE. It doesn't matter which term you apply to your computer system. Following, we discuss some of the applications in which these terminals are used.



In *Analysis*, the terminal is used as an analysis tool. The operator and the machine operate together to achieve information retrieval, input data screening and updating, tape file editing, and on-line programming or debugging. The terminal is also used as a teaching device.

Performance *monitoring* is being used more and more by workcenter personnel as an aid to improving system effectiveness. Present large-scale computing systems, particularly those involving remote terminals and telecommunications devices, have become very complex. Even monitoring a system to determine if it is functioning normally can be complicated. However, performance monitoring has become invaluable for obtaining information about the inner workings of a system. It is used to fine tune system operation in order to complete assigned jobs in a timely manner, yet at the same time allow the system to complete on-line operations.

Computer operating systems being manufactured today can be described as generic systems in the sense that they are designed for use in a large variety of versions, each catering to a different hardware configuration and different type of workload. In any given environment, it is necessary to create a specific version of the generic system. This involves selecting the desired system components from the total set of components that make the generic system. It also involves choosing settings of the many and varied parameters that govern the actions of the system. These decisions can be very difficult. Performance monitoring provides data on which to base the decisions and can be used later to determine how well founded they are. It can also play a similar role when making such decisions as hardware configuration changes and layout of data in a disk store.

Performance monitoring involves the counting and, perhaps, timing of selected events, either continuously or on a sampling basis. This can be done with hardware as well as software devices. To add such data collection facilities to an existing operating system may be very awkward. For Air Force systems, it is now a requirement to predict what events need to be monitored and, then, to build the necessary facilities into the system at the beginning.

With monitoring, tabulation and analysis of events can be very time consuming. However, the major difficulty lies in deciding which events to monitor in order to obtain the required data and, then, interpreting the results.

*Batch terminals.* The primary purpose of a batch terminal is to allow users access to the computer system just as they would via the system's local I/O devices, but to do so from locations remote to that system. The simpler batch terminals are normally capable of either transmitting or receiving, but cannot perform both functions at the same time.

*Interactive terminals.* The primary purpose of an interactive (or time-sharing) terminal is to allow the user of a computer system to use it in a mode (called "interactive" or "conversational") that is characterized by relatively fast response time to each individual user request—thus the term "interactive."

**Smart terminals.** Smart terminals, also called "intelligent" terminals, were designed to operate independently from the mainframe. If the mainframe computer becomes inoperable and unable to communicate with its various terminals, serious problems develop. To reduce such dependence upon the mainframe's availability, intelligent terminals were developed. In short, an intelligent terminal has its own processor and operating system. Smart terminals, however, must use a "terminal emulation" program to give themselves the appearance of a dumb terminal in the eyes of the mainframe computer.

**Microcomputers.** Microcomputers (fig. 2-19) have been developed to the point that they are both intelligent and inexpensive. Their capabilities include all the previous uses of consoles, dumb terminals, and smart terminals. Microcomputers will eventually replace all other types of terminals in use.

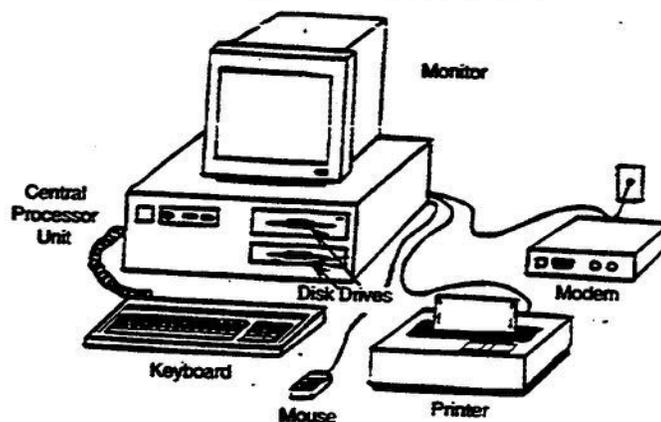


Figure 2-19. Microcomputer.

## 617. Printers

Today, there are many different kinds of printers in use. The development of new printers is almost a daily event. In this lesson, we discuss the differences between printers and tell how the more common ones operate.

**Impact printers.** Printers provide the permanent visual record (hard copy) from a data processing system. The impact printer can be distinguished from other types of printers by the method it uses to put letters and characters on a page. If the print mechanism actually strikes the page to create an image, the printer is considered an impact printer. The serial printer is the most common type.

**Serial printers.** Serial printers print one character at a time. There are four basic methods of operation for serial-impact printers. The simplest method uses a mechanism similar to a typewriter. The characters are on a ball or cylinder that rotates until the right character is in position. The ball or cylinder is then forced against a carbon ribbon and paper. Maximum print speed for this type of printer is about 30 characters per second because of the printer dynamics. Typewriter-like printers are noisy, but print quality is excellent.

A second type of serial-impact printer uses a *print wheel*. The characters are located around the periphery of the wheel. An armature impacts the hammer onto the paper, forcing the paper onto the selected character. Since the print wheel is always rotating, even during hammer impact, hammer timing is critical. Printer dynamics limit print speed to about 35 characters per second.

A third type of serial-impact printer uses a disk with the characters on "petals." Printing occurs when the selected character is in position and all motion is stopped. A small solenoid deflects the petal against the ribbon and paper. This unit is quieter and easier to service than other types of serial-impact printers. Print quality is excellent, and print rates up to 120 characters per second are available with this print mechanism.

The fourth type of serial-impact printer is the *dot matrix* printer (fig. 2-20 and 2-21). This printer was developed to increase printer speed without adding to the complexity or cost of conventional printers. Dot matrix printers can attain print speeds well over 100 characters per second. The print mechanism (fig. 2-22) consists of pins arranged in a solenoid-actuated matrix. The dot matrix heads are simple and contain few moving parts. The simpler matrix size is 5 by 7 pins, but other sizes are becoming available on some newer printers.

The seven pins in the print head hit the paper multiple times to form each character. Because the characters are not solid, the print quality of a dot matrix printer is not as good as that of the other types. The versatility and low expense of the dot matrix printer makes it the most popular for most data processing applications.

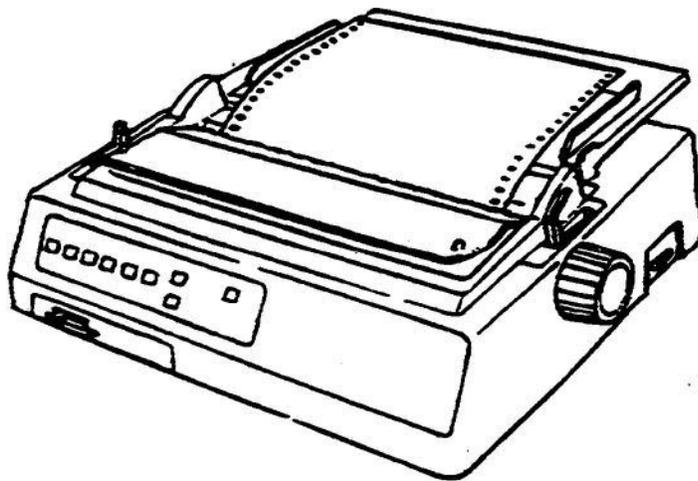


Figure 2-20. Dot matrix printer.

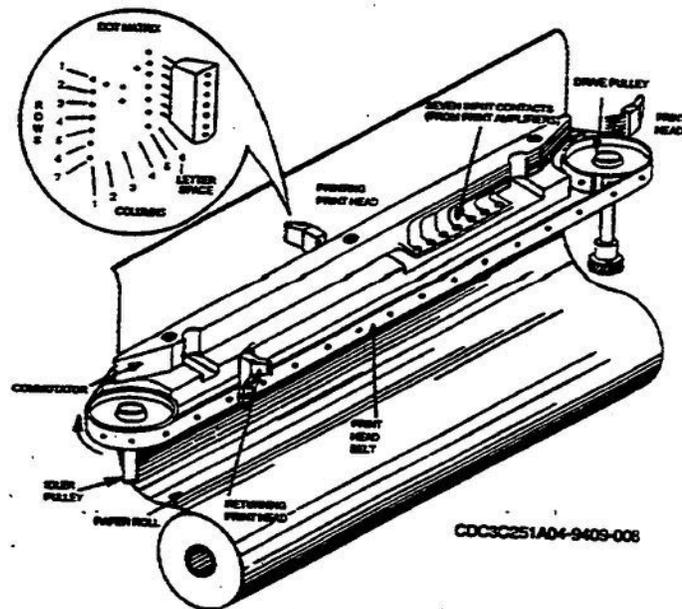


Figure 2-21. Dot matrix printer (internal view).

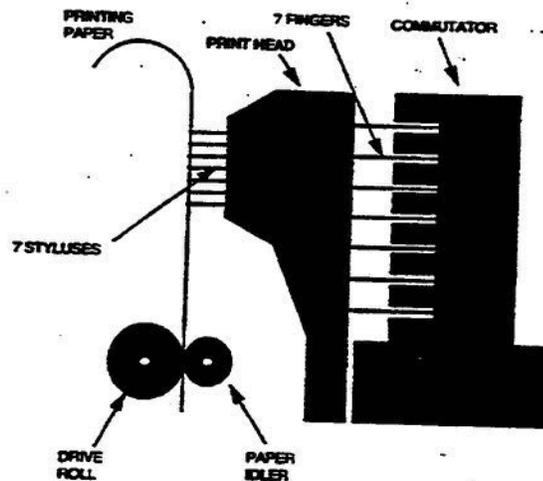


Figure 2-22. Dot matrix printhead.

**Line printers.** Line printers are, in general, much faster than serial printers, but are also considerably more expensive. There are three principal types of impact line printers: drum, chain, and matrix scanning.

**Drum printers.** Drum printers are similar to the serial print-wheel printer. In place of the narrow print wheel and single hammer, a drum covers the full width of the paper, with a hammer for each column position. The drum has one full character set for each print position and one character in each position is printed per revolution. Each rotation of the drum, therefore, prints one line. Drum printers can attain speeds of 35 lines per second or 2,100 lines per minute.

**Chain printer.** The second kind of impact line printer is the chain printer. One or more character sets on a chain or belt moves past a row of hammers. The hammers are activated when the desired characters come into position. A line is printed as each character set moves past the bank of hammers. Thus, chain printers are capable of higher print speeds than drum printers. They print from 200 up to 3,000 lines per minute.

**Matrix scanning.** The third type of impact line printer uses a matrix scanning mechanism that prints the characters in dot matrix form. One or more hammer banks with a hammer for each print position oscillates horizontally to produce a line of dots. The paper advances vertically one dot row at a time until the complete line matrix is scanned. The dots overlap to provide good print quality.

With the exception of paper feed and ribbon mechanisms, all other functions of matrix scanning printers are performed electronically. Advantages of dot matrix printers include reliability, easier maintenance, and font and character sets that are easily interchangeable by programming.

**Nonimpact printers.** Nonimpact printers also employ various techniques. The most popular nonimpact line printer is an electrostatic device.

**Electrostatic device.** This printer consists of one or more linear arrays of conducting nibs. The surface of the paper is coated with a nonconducting dielectric material that holds an electrical charge applied by the writing nibs. The paper is then exposed to a toner to make the characters visible. Characters are formed in the dot matrix format. Electrostatic printers have a print speed range from 500 to 3,600 lines per minute.

**Ink jet.** The ink jet system technique directs a high-velocity stream of ink at the paper. The ink stream is broken into microscopic droplets, which are then given a charge and passed between electrodes that produce an electric field. The ink droplets are deflected to form the characters on ordinary paper. Ink drops that are not charged are directed to a return channel for reuse. Lines are printed at about 180 lines per minute. Recent technology has advanced this type of printer to include full color printing.

**Electrographic printing.** This technique is also referred to as *laser* and *xerography* printing. Electrographic printers (fig. 2-23) are similar to electrostatic

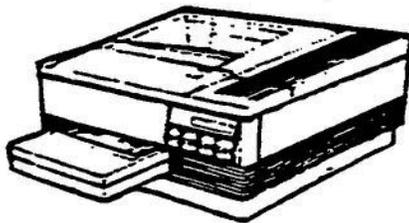


Figure 2-23. Common laser printer.

printers except the electrical charge is deposited on a drum to form the output image. Ordinary paper is passed over the drum in the presence of a toner to

transfer the image to the paper. These devices have a print speed range of 4,000 to 18,000 lines per minute.

**Graphic devices.** These devices are used for both passive (providing output only) and interactive work. An example of a passive graphic device is a plotter, or a microfilm developer. Interactive graphic devices are units with both graphics output and input. Some graphic input devices include light pens, joy sticks, and digitizers.

**Associated printer devices.** A few other devices warrant discussion before we conclude this session on printers. The first is the *VFU* (vertical-format unit). The VFU is mounted on the side of most older-model printers. The shaft from the paper-feed clutch extends into the VFU to turn a sprocket wheel in correspondence with paper advance. When a format tape is engaged with the sprockets, it is moved between a set of photodiodes and lamps. The tape is prepunched with holes in a channel for various format choices. The holes in each channel indicate the line of the form at which the skipping paper-feed cycle is to terminate. As the tape moves over the photodiodes, a pulse is generated by each hole. The output from the selected channel is synchronized with the output of the paper-drive pulse generator to signal the end of the cycle. At the end of the printing cycle, the stationary sets can be split into single sheets by means of decollators and bursters.

**Decollators.** As the name implies, decollators separate multi-part paper into single parts. Many production requirements make it necessary to use very long strings of multi-part paper. When preparing these output documents for delivery after printing, it is necessary to dislodge the different plies of paper. A decollator is designed to save countless man-hours by doing this separation process quickly and accurately. The decollator is illustrated in figure 2-24.

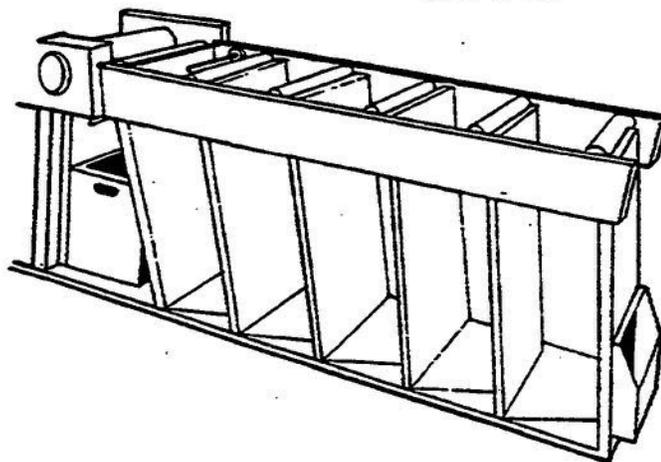


Figure 2-24. Decollator.

**Bursters.** Bursters (fig. 2-25), on the other hand, are designed to separate forms. Many printer outputs require the use of multi-part continuous forms. After a form-print operation is complete, a burster can save endless man-hours of work by

separating forms before distribution to customers. Bursters also remove the tractor feed edges of continuous-feed paper and carbons if carbonless paper is not used.

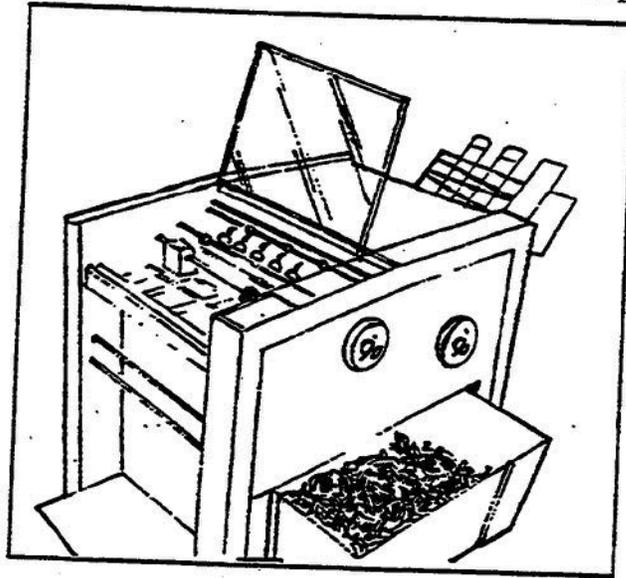


Figure 2-25. Burster.

Although you may not work directly with some of the above peripherals, many of the subscribers your facility supports do. For this reason, the systems controller must be aware of their existence and have some knowledge of their capabilities.

### 618. Optical read units

Optical character recognition (OCR) is defined as a high-speed process of recognizing and translating machine-printed or hand-printed words, letters, symbols, and numbers into information that the computer can process. The data is directly machine readable while still being readable by people.

OCR was developed by using the techniques of optical mark reading. Optical mark readers generally use a single photocell to detect the presence or absence of an optical mark in a given marking position. Early optical character readers used a matrix of photocells onto which characters from a defined stylized font were decoded into a particular character. The advent of "flying spot" electronic and laser scanners made OCR reliable enough that it is now an acceptable and viable means of inputting data.

**Magnetic ink-character readers.** Magnetic ink-character readers are machines that read documents inscribed with special magnetic ink characters at high speeds and interpret them for the system. They are used extensively in finance operations to process checks at electronic speeds.

**Optical read units.** Optical read units are strictly input devices. They are used extensively to enter data by exposing the characters to a light source. The light source reflects the characters onto a photosensitive unit and sends the data to the

control unit. If the data is to be corrected or reviewed, it is sent to a visual display unit.

There are two types of optical read units: the optical character reader and the optical scan unit. The *optical character reader* (fig. 2-26) reads one character at a time, while the *optical scan unit* (fig. 2-27) reads (scans) an entire document before processing through the use of a laser reading device.

**Optical-character readers.** Optical-character readers can read some hand-printed or machine-printed numeric digits and alphabetic characters from documents. The read-and-recognition operation is automatic and takes place at electronic speeds. Optical-character reading is used intensively in postal systems and for reading insurance premiums, notices, and invoices.

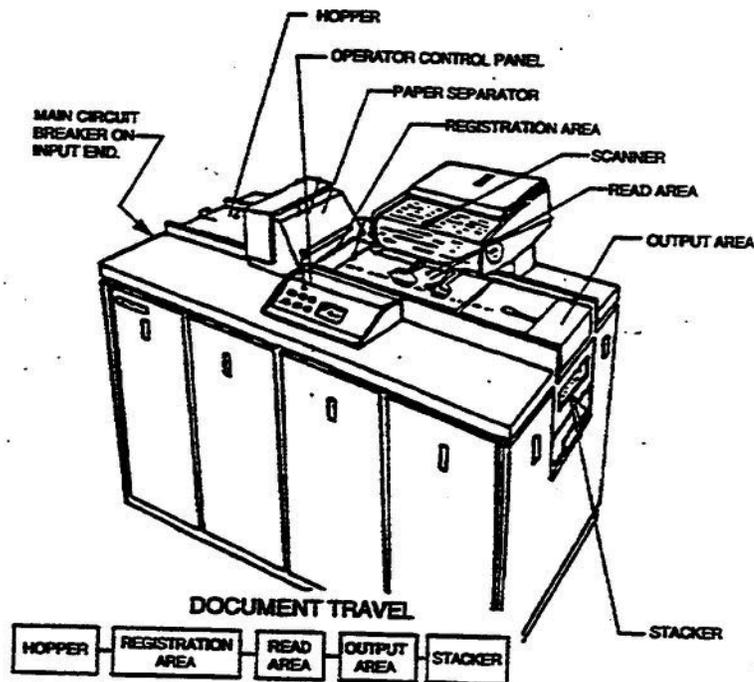


Figure 2-26. Optical character reader.

Another example of this technology is the *Optical Mark Reader* which scans documents for black areas. This is used in testing and survey work.

**Scanning.** Before a character can be converted into a machine representation, it must pass through two systems: an optical scanning system and a recognition system (the analysis and decoding of optical output). The optical system not only governs the speed and flexibility of the unit, but also determines the range of inks and paper-surface qualities needed. The scanning process determines the presence or absence of a mark by the amount of light reflected from the area being scanned.

**Fonts.** The problems of reading printed characters by a machine increase drastically with significant increases in the number of characters in the set. Obviously, the least expensive readers are those designed to read only a single

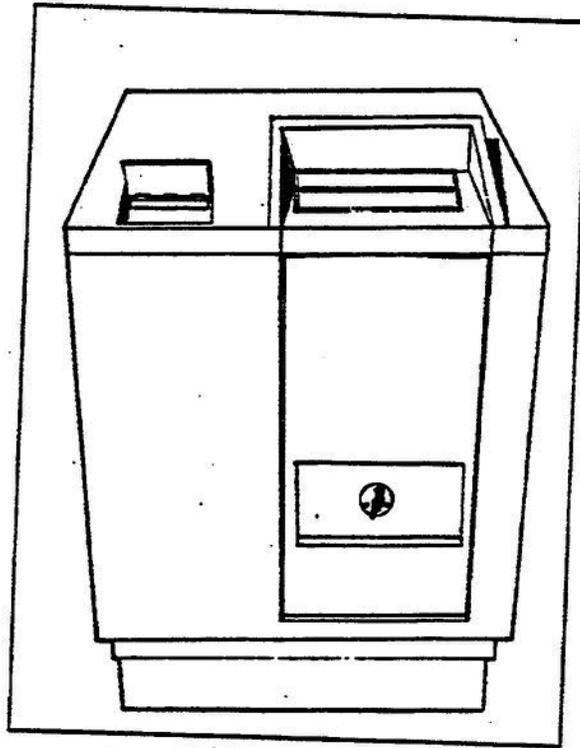


Figure 2-27. Optical scan unit.

stylized uppercase type such as the OCR-B. Stylizing a type font reduces the number of similarities between characters and makes their identification easier for the machine.

**Applications.** OCR has developed as an alternative solution to the data input bottleneck problem of the past. OCR enables the source documents themselves to pass information directly to the computer. The time frequently lost between source documents and computer-processed information is reduced or eliminated. The exact savings depends on the method of document creation and the system that OCR can replace.

**Quality.** Good print quality is the key to reading performance and subsequent overall OCR system performance. The quality of the printing affects both the total process time and the error recovery effort on-line and off-line. Also, the sensitive physics of light impose specifications of paper quality, background inks, and print quality on OCR forms no less rigid than those of the scanning machines themselves. Cleanliness requirements are also stringent; there must never be stray marks.

## 619. Monitors

The term "monitor" usually refers to the entire casing that contains the computer's video display screen. There are many ways to classify monitors, but the most common is to describe them in terms of color capabilities. This classification

separates monitors into three distinct categories: monochrome, gray-scale, and color.

**Monochrome.** Monochrome monitors actually display two colors, one color for the background and one for the foreground. The colors can be black and white, green and black, or amber and black. Monochrome monitors generally produce a sharper image than color monitors, but color monitors are useful for certain types of graphic displays.

**Gray-scale.** A gray-scale monitor is a special type of monochrome monitor capable of displaying different shades of gray.

**Color.** Color monitors can display from 16 to over one million different colors. They are sometimes called RGB monitors because they accept three separate color signals; red, green, and blue. Color monitors are further categorized as CGA, EGA, VGA, and SVGA monitors.

**CGA.** Color Graphics Adapter (CGA) is a graphics display system for certain types of personal computers (PCs). First introduced in 1981, CGA was the first color graphics system designed for PCs and primarily for use with computer games. It does not produce sharp enough characters for extensive graphics programs, but it does accept 16 colors. It also has a graphic resolution of 640 by 200 pixels. Pixels are single discrete points or elements that, when put together, form a video image. CGA has been superseded by EGA, VGA, and SVGA systems.

**EGA.** Enhanced Graphics Adapter (EGA), another graphics display system, was first introduced in 1984. EGA supports 16 colors from a palette of 64 colors and provides a resolution of 640 by 350 pixels. This system's performance is better than CGA, but not as high as VGA or SVGA.

**VGA.** Video Graphics Array (VGA), a graphics display system developed for IBM® PCs, is available to other computers by adding a printed circuit board to support the system. In graphics mode, the resolution is either 640 by 480 pixels with 16 colors or 320 by 200 pixels with 256 colors. This is significantly better than the systems we already discussed. However, VGA uses analog signals rather than digital signals used by CGA and EGA. Consequently, a monitor designed for one of the older standards cannot use VGA. The majority of the add-on boards contain one 15-pin port designed to connect to an analog monitor. Some types of these add-on boards, however, have an additional 9-pin port that enables you to connect to a digital monitor.

**SVGA.** Super Video Graphics Array (SVGA) is a graphics display system designed to provide higher resolution than any of those standards already described. There are two types of SVGA: one with a resolution of 800 by 600 pixels, and the other with a resolution of 1024 by 768 pixels. The Video Electronics Standards Association (VESA) recommends SVGA become the industry standard for video displays.

### Self-Test Questions

After you complete these questions, you may check your answers at the end of the unit.

#### 616. Consoles and terminals

1. In large computer systems, how does an operator gain access to load programs such as operating systems and diagnostic routines?
2. What are some items found on a system console?
3. What is the primary function of a system console?
4. What component associated with a system console is used to provide an audit trail for system monitoring?
5. What purpose does a computer terminal serve?
6. What part of a computer terminal provides input to a computer?
7. What part of a console or terminal can display data in the form of characters or pictures?
8. What are the tiny squares on a video display unit called?
9. What determines the quality of a picture displayed on a video display unit screen?
10. What are two types of computer terminals?

11. What type of computer terminal takes all of its capabilities from the central processing unit?
12. Name some ways that a computer terminal can be used in analysis.
13. How does monitoring play a part in computer operation?
14. What is the purpose of a batch terminal?
15. What is the primary purpose of an interactive computer terminal?
16. What makes a smart terminal intelligent?
17. Is a microcomputer considered a smart terminal or a dumb terminal?

## 617. Printers

1. For what purpose are printers used?
2. What is the term used to describe the permanent visual record created by a printer?
3. How can you distinguish an impact printer from other types?
4. How many characters are printed at one time on a serial printer?

5. Which type of serial impact printer makes the paper strike the print mechanism instead of making the print mechanism strike the paper?
6. What type of printer was developed to increase printer speed without adding to the complexity or cost of conventional printers?
7. What makes dot matrix printers the most popular choice for use with data processing?
8. How does the speed and cost of line printers compare to that of serial printers?
9. Name three principal types of impact line printers.
10. How many lines of characters are printed for each revolution of the drum on a drum printer?
11. What are some advantages that dot matrix printers have over drum or chain printers?
12. What is the most popular type of nonimpact printer?
13. Which type of nonimpact printer uses a high-velocity stream of ink to produce print characters on ordinary paper?
14. What other names are used to describe an electrographic printer?
15. How is the operation of a laser printer similar to that of an electrostatic printer?

16. Give one example of a passive graphic device and one example of an interactive graphic device.

17. What device is mounted on older printers to control the advance of the paper?

18. What printer device separates long strings of multi-part paper?

19. What printer device can separate forms?

### **618. Optical read units**

1. What is optical character recognition (OCR)?

2. What is a popular use for magnetic ink-character readers?

3. Name some of the common uses of optical-character readers.

4. Are optical read units considered input or output devices?

5. Which type of optical read device reads an entire document before processing?

6. How does the scanning process of an optical reader determine the presence or absence of a mark?

### **619. Monitors**

1. What are the three categories of monitors?

2. How many colors do monochrome monitors display?
3. List the four categories of color monitors.
4. What are the two major characteristics of color monitors?

## 2-4. Operating Systems and Computer Memory

The operating system is the most important program that runs on a computer. Every general-purpose and network computer must have an operating system to run other programs.

### 620. Operating system functions

Here we discuss the functions of an operating system, their different classifications, and some of the more popular operating systems used today.

**What is an operating system?** Operating systems are computer programs that recognize input from a keyboard, send output to the display screen, keep track of files and directories on a disk, and control peripheral devices such as disk drives and printers. In order to accomplish these tasks, it interacts with application software programs and all other components of the computer system.

For large systems and networks, operating systems are responsible for even greater tasks. They control and direct processes and ensure other programs and users running at the same time do not interfere with each other. The operating system also handles security by ensuring unauthorized users do not access the system.

**Operating system elements.** The primary elements of an operating system are the supervisor, input/output (I/O) manager, file manager, command processor, and transient utility programs. When a computer is powered on, the operating system automatically loads itself. Certain essential elements (the most frequently used functions) transfer into the computer's random access memory (RAM). Utility programs (less frequently used functions) reside on the system's hard disk and are loaded into RAM as needed. Let's review the functions of the operating system elements beginning with the supervisor.

**Supervisor.** The supervisor, or executive, is the master control program of the operating system. It must be resident in RAM before processing can begin. Once installed, it is responsible for the integrity of the system. It coordinates,

interrelates, and monitors all aspects of the system software. In other words, the supervisor regulates the flow of all input and output activity. It coordinates and executes all other program activities. The supervisor runs at the highest priority and is protected from interference from user programs.

**I/O manager.** This portion of the operating system instructs the computer hardware on how to decode keyboard signals and then displays the appropriate characters on the video display screen. It also encodes and decodes all data transferred between other programs and peripherals.

**File manager.** All information stored on a disk is organized into either a data file or program file. The file manager handles the naming, loading, and saving of files and the retrieval of stored information. It also maintains a record of everything stored on a particular disk. A directory lists every file on the disk according to its filename, the type of information it contains, the date and time it was created, and the size (number of bytes) of the file.

**Command processor.** This portion of the operating system interprets commands and instructs the computer to perform various functions. Commands begin with a key word or filename. A key word instructs the operating system to execute a specific type of command and a filename instructs the operating system to load a specific program file into RAM before it executes the program. When the system is ready for a new command, it displays a prompt on the video display that serves as a visual indication of where to enter the next command on the screen. As each command is entered, the command processor checks it for validity.

**Transient utilities.** These are programs that are pre-assembled and specialized to perform frequently required tasks. File updating, sorting, and backing-up of files are a few tasks accomplished with transient utility programs. These programs are usually stored on the computer system's hard disk and called into RAM by the supervisor when needed. Many of them occupy unused space in RAM (even after they accomplish their task) until they are displaced by other programs. For this reason, these are called terminate-and-stay-resident (TSRs) programs. This capability prevents the supervisor from having to call them into RAM every time they are needed.

**Classification of operating systems.** Operating systems are classified as one of four types: multi-user, multi-processing, multi-tasking, or real-time.

**Multi-user.** Operating systems with this classification allow two or more users to run programs at the same time. Some operating systems permit hundreds or even thousands of users to operate concurrently.

**Multi-processing.** Under this classification, operating systems allow multiple users to run two or more programs at the same time. Each program being executed is called a process. Multi-processing uses more than one processor to accomplish user tasks. For example, one such system uses 4 CPUs that are separate, but have equal access to information and attached devices and share one operating system.

**Multi-tasking.** Operating systems designed for multi-tasking allow a single processor to run more than one task. The terms *multi-tasking* and *multi-processing* are often used *incorrectly* as though they are interchangeable terms. Each carries its own different technical meaning.

**Real-time.** Real-time operating systems respond to input instantaneously. General purpose systems such as MS-DOS® and UNIX® are not real time. Real-time operating systems are designed to perform such tasks as navigation, in which the computer must react to a steady flow of new information without interruption.

As a type of systems software, operating systems provide a software platform on top of which other programs, called *application programs*, can run. The application programs must be written to run on top of a particular operating system. The choice of operating system, therefore, determines to a great extent the number and type of application programs the computer system can run.

As a user, you interact with the operating system by inputting a set of commands through the keyboard. For example, to copy a file on a computer using the MS-DOS® operating system you input the command, COPY, along with other information (such as filename and destination) to direct the operating system to perform the copy function.

## 621. Popular operating systems

Many different operating systems are used today to run a variety of personal computers, mainframes, and work stations. Here we discuss some of the most popular ones.

**MS-DOS®.** Microsoft Disk Operating System was originally developed in the 1980s by the Microsoft Corporation for the International Business Machine Corporation (IBM). Because of its suitability, it has become the predominant operating system for all IBM® and IBM®-compatible PCs. MS-DOS® is a single-user, single-tasking operating system, but one whose capabilities are expandable by adding integrated software programs to the individual computer system.

**OS/2®.** Object-oriented Software (GUI) is a relatively powerful operating system developed by IBM® and Microsoft Corporation that runs only on computers containing a 80286 or greater microprocessor. OS/2 is generally compatible with MS-DOS®, but contains many additional features. Some of these features can be made available to MS-DOS® systems with the addition of operating environment programs such as Microsoft Windows®.

**UNIX®.** The American Telephone and Telegraph (AT&T) UNIX® System V operating system runs on a variety of personal computers, mainframes, minicomputers, and workstations. It has become the defacto standard for minicomputers and workstations due to its powerful multi-user and multi-tasking capabilities. An example of this system is the 3b2 at your own wing that provides

many users with electronic mail and file transfer services. UNIX® is the operating system of choice for many software development projects because of its flexibility and usability with such a wide array of computers. We cover network operating systems in more detail later when we discuss computer networking schemes.

This ends our discussion on operating systems. Remember that these are the most critical programs installed within a computer system and must function correctly in order for all other programs to run. Choose operating systems wisely by comparing your processing and tasking needs with the different systems classifications and various brands of operating systems available.

## 622. Computer memory

Two factors that greatly affect how efficiently a computer system processes information is the amount of memory it contains and how well this memory is managed. The amount of memory available also limits the size and number of programs that can run on the computer. Regardless of how much memory a system has, you will not recognize its full potential if you manage this valuable resource poorly.

**What is memory?** Memory refers to the microchips installed in a computer that store information for current or later use. Computers have to store instructions or data when processing information or performing operations. Don't confuse memory with storage devices designed to simply store information to be retrieved later into memory. A hard disk containing 250 megabytes of storage space does not imply that the system has a memory of 250 megabytes.

Computer memory is measured in kilobytes or megabytes of information. One kilobyte equals 1024 bytes and one megabyte equals 1,048,576 bytes. Therefore, if your computer contains 640 kilobytes of memory, it can hold 655,360 bytes at one time. Software programs require a minimum amount of RAM to work properly. You can generally find these memory requirements identified on the software packaging.

In general, the more memory you have, the more data you can store in memory at one time. You can increase the amount of memory on your system by plugging additional memory boards into specialized slots inside your computer. For example, you might add a 2-megabyte memory board to a system that has 1 megabyte of memory on its main system board; the system would then have 3 megabytes of memory.

Computers contain two types of memory, Read-Only-Memory (ROM) and Random-Access-Memory (RAM). Physically, ROM and RAM are microchips located on the computer's system board.

**Characteristics of memory.** The storage of information is not restricted to the memory unit of a computer. The arithmetic logic unit, the control unit, and the input/output units all may require some form of information storage.

A memory unit stores binary information in groups called WORDS, each word being stored in a memory register. A word in memory is an entity of "n" bits that moves in and out of storage as a unit. The storage location within memory for one word is often referred to as a "cell." A memory word may represent a number, an instruction, a group of alphanumeric characters, or any binary coded information.

The communication between a memory unit and its environment is achieved through two control signals and two external registers. The control signals specify the direction of transfer required; that is, whether a word is to be stored in a memory register or whether a word previously stored is to be transferred out of a memory register. One external register specifies the particular memory register chosen out of the thousands available; the other specifies the particular bit configuration of the word in question. The control signals and registers are shown in the block diagram in figure 2-28.

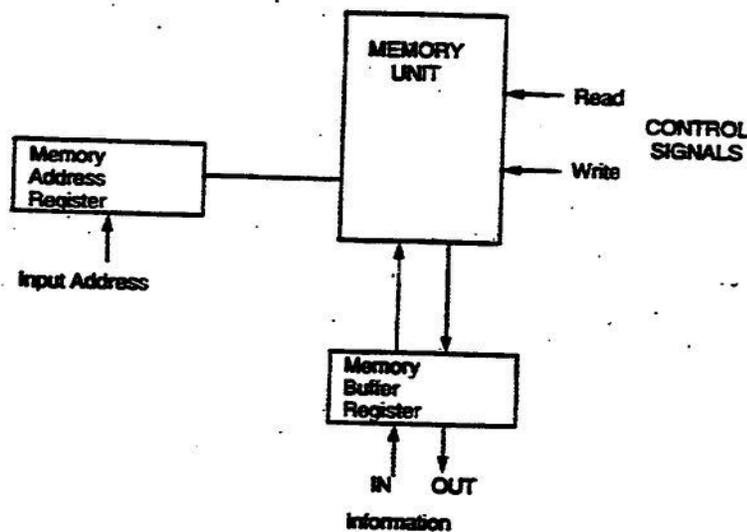


Figure 2-28. Memory unit block diagram.

The memory ADDRESS REGISTER specifies the selected memory word's location. Each location in memory is assigned an identification number starting from 0 up to the maximum number of locations available. The number of available memory locations is usually expressed in kilobytes. To communicate with a specific memory location, its location number, or ADDRESS, is transferred to the address register. The internal circuits of the memory unit accept this address from the register and open the paths needed to transfer the word in or out of memory.

The two control signals applied to the memory unit are called READ and WRITE. A write signal specifies a transfer-in function; a read signal specifies a transfer-out function. Upon accepting one of the control signals, the control circuits inside the memory unit provide the desired function.

**Classifications of memory.** Memory may be classified in a number of ways to distinguish their characteristics. Three appropriate classifications are (1) volatility, (2) mode of access, and (3) construction.

**Volatility.** Volatility of a memory element is an expression of its ability to retain stored data when power is removed from the equipment containing the storage device. A *volatile* memory is cleared when the power is turned off and the information is lost. A *non-volatile* memory does not depend upon applied voltages or currents to retain the information; therefore, the stored data is not affected by loss of power. Non-volatile memory includes magnetic core memory and certain types of semi-conductor memory chips.

Another characteristic associated with non-volatile memory is that it can be erasable or non-erasable. Some memory types, once programmed, can never be changed. These devices are called *non-erasable* because the data contained within them is permanent. A non-volatile device which can be reprogrammed is called an erasable device.

**Mode of access.** The time period required by the memory device to provide data to the CPU is very important if the CPU is to function at its designed speed. The references to timing which follow assume that a valid read/write request has been made and that the memory address is specified. When reading data from a memory chip, the time involved between the instant at which a memory chip receives a valid address and the instant at which the output data is placed on the data line (retrieved) is called the *access time*. When writing data into a memory chip, access time is the time between the instant at which the memory chip receives a valid address and the instant at which the storage is completed. The minimum time from the beginning of one access to the beginning of the next is often important and is called the *cycle time*.

This mode of access classification deals with the way information can be written to or retrieved from memory. Each word is stored at a memory location. The mode of access describes the manner in which access to the locations is achieved.

In a *sequential access storage* unit, the access time depends on the location of the desired word in the storage medium. You can relate this to hunting a song on a cassette tape. If you are playing a song at the beginning of the tape and want to access a song in the middle of the tape, you have to sequentially go through all the songs in between. Magnetic tapes are examples of sequential access storage media.

A *random access storage* unit is one in which the time required to read or write a word is independent of the location. Random access can be related to finding a specific song on a compact disc (CD) player. Most CD players allow you to advance directly to any song you want to play without going through all the songs in between. Random access memories are faster than sequential access memories because any memory location can be addressed directly. An example of a random access unit is the magnetic core storage unit, or hard disk.

**Construction.** The classification by construction relates to both volatility and mode of access.

**RAM** is volatile and, as the name implies, uses the random access mode. In common usage, this type of memory is read-write memory. It can have data read from it and data written into each memory location. While many types of memory are random access, the term RAM has come to be used to describe semiconductor read-write RAM. A symbol for one type of RAM chip is shown in figure 2-29.

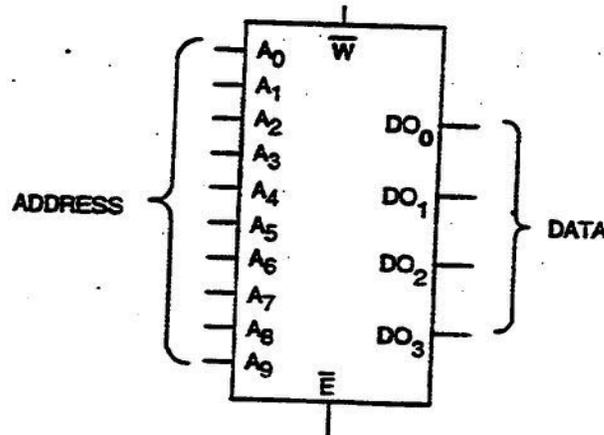


Figure 2-29. 1024 by 4, random access memory (RAM).

This device is capable of storing 1024 4-bit data words. Terminals  $A_0$  through  $A_9$  are the address inputs and enable us to address any memory location from  $000000000_2$  ( $0_{10}$ ) to  $111111111_2$  ( $1023_{10}$ ) a total of 1024 memory locations.  $DO_0$  through  $DO_3$  are the data terminals which provide the ports for input and/or output of 4-bit data words.  $\bar{E}$  is the enable terminal. When it is at a low logic level, it selects the address represented by the binary count present on the address terminals. When the enable terminal is at a high logic level, the device is in the tri-state or inhibited mode and no address is selected.  $\bar{W}$  is the read/write terminal.

When  $\bar{W}$  is at a low logic level, the 4 data bits present at the  $DO$  terminals are written into the memory location established by the address. Conversely, when  $\bar{W}$  is at a high logic level, the 4 data bits stored at the memory location addressed are read out to the  $DO$  terminals.

**ROM** is a random-access memory device just like the RAM memories. The major difference is that the ROM is non-volatile. ROM is the non-erasable type of non-volatile memory discussed earlier. Many ROMs have data, 1s and 0s, permanently hard-coded in when they are manufactured. For that type, each different ROM has a separate printed wiring pattern. For example, one type of ROM might contain the instruction set for a particular type of computer. The number of users for that particular ROM chip may be limited. Rather than make up an original printed wiring pattern for each type of ROM, it is more practical to let the user determine the exact pattern of the ROM.

The two most popular types of alterable ROM are the *programmable read-only memory (PROM)* and the *erasable PROM (EPROM)*.

The user or the factory can program PROMs; however, in either case, special equipment is required. PROMs can be programmed once, but cannot be reprogrammed. The PROM, once programmed, is still the non-erasable type of non-volatile memory. The advantage of PROM is that the manufacturers of several different types of computers can each use the same type of PROM to contain the instruction set for their computer. If changes to an instruction set are necessary, a new chip does not have to be designed—the PROMs for the new instruction set are simply programmed with the new information.

A second, more versatile type of PROM falls into the erasable nonvolatile category of ROM. The erasable PROM can be programmed once, erased, and then reprogrammed. Various types of these devices are erasable PROMs (EPROMs), ultraviolet EPROMs, electrically erasable PROMs (EEPROMs), and electrically alterable PROMs (EAPROMs). The main difference between these types is the methods for writing and erasing data.

**ROM versus RAM.** ROM and RAM differ not only in their construction, but also in the ways they are used. Each performs its own specific functions within a computer.

**Read-only memory.** ROM is memory that contains basic operating instructions used by the computer. ROM literally means that these instructions are permanent and can only be read and not changed by the operator. Examples of operating instructions contained in ROM are system input/output information that recognize the devices connected to the system, system self-test routines used to ensure the system is ready for processing, and system configuration information used to identify those devices connected to the system.

**Random-access memory.** RAM is memory that is installed on the computer's system board or on additional peripheral cards or boards. When programs are processed by the CPU, information is loaded into RAM from disks, tapes, your keyboard, or other input devices. RAM temporarily loads and stores the instructions and data while processing is occurring and only while power is applied to the computer. Once the computer is turned off, all instructions and data are lost. Unlike ROM, RAM may be changed or written to as well as read from.

As an example of how RAM is used, let's compare a computer to the top of your desk. In order to perform a specific task, you (the CPU) need to pull files from your file cabinet (disks, tapes, or optical devices) and place them on top of your desk (RAM) so you can sift through (process) the information. You pull only those files needed to complete the task and then return them to the file cabinet once you're finished with them. In this analogy, RAM acts as the desktop by temporarily storing the information while it is being used.

Today's computer programs contain dozens of files necessary to perform a task. For this reason, RAM size plays an important role in your system's ability to

handle certain programs. If RAM is not large enough to hold all the files, the task cannot be completed. RAM itself can be broken down into several components: conventional, upper, expanded, and extended memory.

*Conventional memory* is the basic type of memory found on all computers and refers to the first megabyte of memory located within the system. Computer programs use conventional memory without the special instructions needed to use other types of memory. Operating systems and device drivers use some conventional memory and the remaining memory is available for other programs. In conventional memory, the first 640 kilobytes is where application programs and current work files are temporarily stored so that processing can occur. The remaining memory between 641 kilobytes and 1 megabyte refers to *upper memory*.

As programs were designed to require more than 640 kilobytes of memory, the need for memory management programs was recognized. A memory management program allows the computer to use its memory more efficiently by loading device drivers and portions of other programs into the upper memory area of conventional memory. The remaining memory areas we discuss require some form of memory management.

*Expanded memory*. One way to add memory in excess of 640 kilobytes to your system is to install expanded memory. Most computers can accommodate expanded memory, which consists of two parts: an expanded-memory board, which must be installed on your computer, and an expanded-memory manager program. Expanded memory increases the capacity of the computer to perform more powerful and detailed applications by making the computer think it is actually conventional RAM.

A program designed to use expanded memory does not have direct access to the information in expanded memory. Instead, expanded memory is divided into 16 kilobyte segments called pages. When a program requests information that is in expanded memory, the expanded-memory manager *maps*, or copies, the appropriate page to an area in the upper memory called *page frames*, which are 64 kilobytes each in size. The program then gets the information from the page frame.

Some programs are unable to use expanded memory because they were not designed to interact with an expanded-memory manager; however, because expanded memory was introduced before extended memory, more programs are designed to use expanded memory than to use extended memory. Because an expanded-memory manager allows programs to access a limited amount of information at one time, expanded memory can be slower and more cumbersome for programs than extended memory.

*Extended memory* is any memory beyond 1 megabyte. This type of memory is available only on systems with 80286 or higher processors. Most programs that use conventional memory cannot use extended memory because the numbers, or *addresses*, that identify memory locations within extended memory to the

programs are beyond the addresses most of them can recognize. Only the addresses in the 640 kilobytes of conventional memory are recognized by all programs. Programs need special instructions (such as MS-DOS® EMM386 extended memory manager) to recognize the higher addresses in extended memory.

Extended memory is fast and efficient for programs that can use it. However, many programs are not designed to take advantage of extended memory. To use extended memory more efficiently, you need to install an extended-memory manager program. An extended memory manager prevents different programs from using the same portions of extended memory at the same time. Some operating systems (such as certain versions of MS-DOS®) have the capability to run in extended memory, thereby leaving more conventional memory available to other programs.

*Upper memory.* Most computer systems have an upper memory area. This area resides immediately adjacent to the 640 kilobytes of conventional memory. The upper memory area is not considered part of the total memory of your computer because programs cannot store information in this area. Upper memory is normally reserved for running your system's hardware, such as your monitor.

Information can be mapped (or copied) from another type of memory to parts of the upper memory area left unused by your system. These unused parts are called *upper memory blocks*. One use of this mapping process is to run those programs designed to use expanded memory that we described earlier.

If your system has a 80386 or 80486 processor and extended memory, some operating systems (such as MS-DOS®) can use the upper memory area, thereby freeing up more conventional memory on your computer for use by other programs. These operating systems have certain commands that enable you to store some device drivers and programs outside of conventional memory, usually in extended memory. It then maps these device drivers and programs into the upper memory area, where they can run successfully. The number of device drivers and programs you can run in the upper memory area depends on how much of the upper memory area is left unused by your system and the expanded memory page frame, if you're using one.

*Cache memory.* Systems that have a hard drive and extended or expanded memory can make use of disk-caching programs to reduce the amount of time the computer spends reading data from the hard drive. These disk-caching programs (such as Microsoft Corporation's SMARTDrive®) accomplish this by reserving portions of extended or expanded memory for their own use. This area of memory is called *cache memory*. SMARTDrive® uses cache memory to store information as the processor reads it from the hard drive. When a program attempts to read information from the hard drive, SMARTDrive® supplies it a whole block of data at a time directly from its cache memory instead, which is much faster than reading it from the hard drive. SMARTDrive® always copies new or modified

information to the hard disk, so there is no danger in losing the data when you turn off the computer.

If you do not use a disk-caching program such as SMARTDrive®, a secondary buffer *cache* can be useful. MS-DOS® uses a secondary *cache* to store the contents of files that programs are currently using. When a program requests part of a file stored on the disk, the operating system provides the information requested. The operating system then stores the next portion of the file in a secondary *cache*, if one is identified. When the program requests the next portion of the file, it is supplied more quickly from the *cache* instead of retrieving it from the hard disk.

You specify a secondary buffer *cache*, and its size, as part of your "buffers" command contained in the CONFIG.SYS file within the operating system program.

### 623. Memory management

Our previous discussions concerning computer memory indicate the importance of managing this resource very carefully in order to optimize a computer's ability to handle instructions and manipulate information. There are basically three ways to manage computer memory effectively: freeing conventional memory, freeing extended memory, and freeing expanded memory.

Portions of this lesson were prepared using materials provided by Microsoft Corporation from their *Microsoft® MS-DOS® Operating System version 5.0 User's Guide and Reference* manual, ©1991. The example we use in our discussion of memory management includes the disk operating system, MS-DOS®, and memory management programs created by Microsoft Corporation. Permission to use these materials is gratefully acknowledged.

**NOTE:** All computers and operating systems do not subscribe to the following management recommendations. Consult the reference material provided with your particular system and programs before attempting these techniques. Also, these recommendations should only be employed by those persons qualified to do so because incorrect system modifications result in system failure. The information provided here is designed only to introduce you to the principles of memory management, not to give you instructions or empower you to make changes to your computer system.

**Freeing conventional memory.** All computer programs require conventional memory in order to run. If a program fails to run because of sufficient memory, the problem is most often caused by a shortage of *conventional* memory.

You can make more conventional memory available to other programs by minimizing how much is used by the operating system, installable device drivers, and other memory-resident programs. A computer program can use only the conventional memory that is available at the time the program is started. If

memory-resident programs are already using portions of memory, then the program you initiate cannot use those same portions of memory.

There are several methods available to free conventional memory for use by programs:

- Run MS-DOS® in extended memory instead of conventional memory, if your system contains extended memory. When MS-DOS® runs in extended memory, it uses the *high-memory area* (the first 64 kilobytes in extended memory).
- Streamline your CONFIG.SYS and AUTOEXEC.BAT files so that they don't start unnecessary memory-resident programs. When the operating system starts, the CONFIG.SYS file tells it which devices to install, which installable device drivers to use, and contains commands that determine how MS-DOS® uses memory and controls files. The AUTOEXEC.BAT file is a batch program that defines the characteristics of each device connected to your system and can also include any MS-DOS® commands you want to carry out when you start your computer.
- Run device drivers and other memory-resident programs in the upper memory area instead of in conventional memory. Remember, only systems containing a 80386 processor or greater allow access to these memory addresses.

**Freeing extended memory.** Some computer programs require the use of extended memory in order to run. Insufficient extended memory, just like insufficient conventional memory, can cause these programs to fail. Some ways to prevent this are listed below:

- Make sure your system contains as much physical extended memory as the program needs to run.
- Make sure your CONFIG.SYS file contains a device command to enable your extended-memory manager program. Remember, most programs need an extended-memory manager in order to use this memory area.
- Reduce the amount of extended memory you allocate for each of your device drivers by changing the device command for that driver. Also, disable unnecessary device drivers.
- Make sure your CONFIG.SYS and AUTOEXEC.BAT files don't start unnecessary programs that use extended memory, just like you did for freeing conventional memory.
- If you are running MS-DOS® in extended memory, you can move it to the conventional memory area at the expense of losing conventional memory space.

**Freeing expanded memory.** Some computer programs require expanded memory in order to run. These programs can fail if there is insufficient expanded memory for them to run. Some ways to avoid this are the following:

- Make sure your system contains enough expanded memory to run the programs.
- If your system contains physical expanded memory, ensure your CONFIG.SYS file contains a device command to enable the expanded-memory manager program supplied with the added memory board.
- Disable unnecessary device drivers and programs in your CONFIG.SYS file that use expanded memory. You may also disable *cache-memory* programs or reduce the amount of expanded memory they are allotted.
- You may install an expanded-memory emulator program that can use extended memory to simulate expanded memory. Programs can then use this *simulated* expanded memory as if it were physical expanded memory. Expanded-memory emulator programs are used only on systems that contain extended memory and do not contain expanded memory. Do not use expanded-memory emulator programs if you are using another expanded-memory manager program.

This concludes our discussion on operating systems and computer memory. In this section we explored some of the functions of operating systems and introduced some of the popular systems in use today. We also covered the characteristics and classifications of computer memory and then introduced you to a few memory management techniques. Hopefully, we've aroused your curiosity and dampened some of your fears about computers and peripheral devices to the point you will want to learn all you can about them. The fact is, they're here to stay, and they do help us provide better service our customers.

---

### Self-Test Questions

After you complete these questions, you may check your answers at the end of the unit.

#### 620. Operating system functions

1. Define operating systems.
2. What additional responsibilities are carried out by operating systems utilized in large computers and networks?
3. What are the five primary elements of an operating system?

4. Briefly describe how operating system elements are distributed when the computer system is initialized.
5. Where within the computer must an operating system's executive be located before processing can begin?
6. Match the functions in Column A with their associated operating system elements in Column B. Column B items may be used once or more than once.

*Column A*

- \_\_\_(1) Coordinates, interrelates, and monitors all aspects of system software.
- \_\_\_(2) Displays a command prompt on the video screen.
- \_\_\_(3) Terminates and stays resident in RAM until needed.
- \_\_\_(4) Maintains a record of all data stored on a disk.
- \_\_\_(5) Instructs computer hardware how to decode keyboard signals.
- \_\_\_(6) Interprets commands and instructs the computer to perform various functions.
- \_\_\_(7) Handles the naming, loading, saving, and retrieving of files.
- \_\_\_(8) Responsible for system integrity.
- \_\_\_(9) Encodes and decodes all data transferred between other programs and peripheral devices.
- \_\_\_(10) Regulates the flow of all input and output activity.
- \_\_\_(11) Checks the validity of input commands as they are entered.
- \_\_\_(12) Performs file updating, sorting, and back-up.
- \_\_\_(13) Occupies unused space in RAM after being executed until displaced by other programs.
- \_\_\_(14) Coordinates and executes all other program activities.

*Column B*

- a. Executive.
- b. Input/output (I/O) manager.
- c. File manager.
- d. Command processor.
- e. Transient utilities.

7. What are the four classifications of operating systems?
8. Which classification of operating systems allows two or more users to run programs at the same time?
9. Which classification of operating systems allows single users to run more than one program at the same time?
10. What determines, to a great extent, the number and type of application programs a computer can run?

### **621. Popular operating systems**

1. What is the predominant operating system used with IBM® and compatible personal computers?
2. What is the defacto standard operating system used for mainframe computers and work stations?

### **622. Computer memory**

1. What factor determines the number and size of programs that can be run on a computer?
2. Where within a computer is memory located?
3. What is the unit of measure for computer memory?
4. How can you increase the memory capacity of a computer?

5. What are the two types of computer memory?
6. In what fashion are groups of binary information stored in computer memory?
7. What does a memory word represent?
8. How is communication achieved between a computer's memory and its environment?
9. Define the purpose of memory control signals.
10. Briefly describe the functions of external memory registers.
11. Define the purpose of a memory address register.
12. How is a specific memory location accessed?
13. What are the functions of the read and write control signals that are applied to a computer's memory unit?
14. How is memory classified?
15. What is the difference between volatile and nonvolatile memory?
16. Which is the faster access method, sequential or random? Why?
17. Is RAM volatile or nonvolatile?

18. What is the major difference between RAM and ROM?
19. What are the two most popular types of alterable ROM chips?
20. Describe the difference between PROMs and EPROMs.
21. What type of programs are contained in ROM?
22. What is the purpose of RAM?
23. Why is the capacity of a computer's RAM important?
24. Match the statements in Column A with their associated types of RAM in Column B. Column B items may be used once or more than once.

*Column A*

- \_\_\_ (1) Memory existing above one megabyte.
- \_\_\_ (2) Requires a memory management program.
- \_\_\_ (3) Memory added to a computer by installing memory circuit boards.
- \_\_\_ (4) Memory residing between 640 kilobytes and one megabyte.
- \_\_\_ (5) Not considered part of total memory.
- \_\_\_ (6) Programs require special instructions to recognize its higher memory addresses.
- \_\_\_ (7) Refers to the first megabyte of system memory.
- \_\_\_ (8) Increases processing capacity by making the computer think that it is conventional RAM.
- \_\_\_ (9) Normally reserved for running system hardware.
- \_\_\_ (10) Temporarily stores portions of files currently in use.
- \_\_\_ (11) The basic type of memory found on all computers.

*Column B*

- a. Conventional.
- b. Expanded.
- c. Extended.
- d. Upper.
- e. Cache.

### 623. Memory management

1. What is the most likely cause of program failures attributed to insufficient memory?
2. Give three ways to free conventional memory.
3. What are three ways you can free extended memory?
4. Give three ways to free expanded memory.

---

### Answers to Self-Test Questions

#### 608.

1. Charles Babbage.
2. Mainframes using vacuum tubes; smaller computers using transistors; and computers using silicon chips.
3. The electronic numerical integrator and calculator (ENIAC).
4. 1953.
5. The Manchester University Mark I.
6. 1976.
7. A mainframe is a computer with its parts mounted on frames inside large metal cabinets.
8. Minicomputers are designed for one particular type of work.
9. Data is the physical form of information. A program gives a computer the precise instructions it needs to process data.
10. Because each computer understands only a limited set of *very exact* instructions.
11. Computerese.
12. (1) f, (2) a, (3) g, (4) c, (5) h, (6) e, (7) d, (8) i, (9) b, (10) j.
13. The address bus, the data bus, and the control bus.

#### 609.

1. Memory and decision elements.
2. The input unit.
3. The program currently being executed.
4. (1) g, (2) e, (3) c, (4) d, (5) f, (6) a, (7) b.

**610.**

1. The central processing unit, input unit, output unit, and memory.
2. The central processing unit.
3. It must be able to call instructions from memory, decode their binary contents, and execute them.
4. They allow the processor to execute the system program without requiring the program to monitor the status of each peripheral device.
5. The registers, arithmetic logic unit, and control circuitry.
6. Registers.
7. It identifies the memory location of the next instruction to be performed.
8. The instruction is placed into the instruction register, decoded by the instruction decoder, and then executed.
9. The flag register indicates specific conditions after an arithmetic or logic operation.
10. The control unit directs and coordinates the step-by-step operations of the computer.
11. From stored programs.
12. It receives input from the decoder, generates control signals in the proper order to accomplish instructions, and manages the control bus.
13. Clock inputs.
14. Clock speeds, instruction sets, data bus size, hardware requirements, and symbology.

**611.**

1. A disk controller, disk unit, and a disk pack.
2. The disk controller.
3. Fixed-head disk units.

**612.**

1. Hard disks can be installed as either internal or external storage devices.
2. Drive C.
3. They are completely enclosed in air-tight cases that protect their surfaces from being damaged by contaminants.
4. Installing, formatting, partitioning, and repartitioning.
5. A file arrangement on a segment of a hard or floppy disk.
6. Small sections of a disk called "blocks."
7. (1) a, (2) g, (3) e, (4) b, (5) b, (6) d, (7) c, (8) d, (9) b, (10) f, (11) a.
8. Install it.
9. The disk must be formatted.
10. Formatting a hard disk maps an addressing scheme on the surface of the disk and divides it into tracks and sectors.
11. Partitioning determines the size of the blocks to be used, the file systems that will reside on the disk, and the number of blocks to be used for each file system.

12. The decision you make on block size determines how efficiently your system utilizes disk storage space and processing time.
13. You repartition the hard disk.
14. To enlarge the size of one partition, you must reduce the size of another partition.

**613.**

1. Data can be accessed from disks either randomly or sequentially. Data stored on magnetic tape can only be accessed sequentially.
2. Portable disk packs can be readily removed from the computer and stored elsewhere. This means that only those files required for a particular application need be in use.
3. Floppy disk drives can be installed internally to the computer or attached as an external device.
4. A 3.5 inch low-density floppy disk can store up to 720 kilobytes of data and a high-density disk can store up to 1.44 megabytes of data.
5. A 5.25 inch low-density floppy disk can store up to 360 kilobytes of data, a double-density disk can store 720 kilobytes of data, and a high-density disk can store up to 1.2 megabytes of data.
6. A head crash may occur and destroy data on the disk, the disk itself, or the disk drive.
7. A removable magnetic or optical disk mounted in a protective cartridge.
8. Optical disk units use a laser to read pits in the disk rather than a read/write head to read magnetic spots.
9. Compact disk-read only memory (CD-ROM), write once read many (WORM), and fully rewriteable.
10. They cannot be written on and can only be used to retrieve data previously stored on them.
11. To archive important data that is not expected to change over long periods of time.
12. CD-ROMs have a larger storage capacity than other types of storage devices.
13. You can write data *once* to a WORM disk.
14. Floptical disks can be erased and reloaded with new data.

**614.**

1. They use a read/write head to read magnetized spots on a tape and to magnetize areas of parallel tracks along the length of a tape.
2. By sections of blank (unrecorded) tape, called interlock gaps, that are produced on the tape at the time the data is recorded.
3. As auxiliary storage, input, and output devices.
4. Tape controllers provide the means to monitor transfer errors between an input/output controller and tape units and to monitor parity errors on tape units. They also allow access for an operator to redesignate logic identifiers and to back-space or advance one record on the tape.
5. A tape transport system, a read/write area, and an operator's control panel.
6. The capstan drives the magnetic tape.
7. Two; one with the tape unit's supply reel and one with the take-up reel.
8. Read heads, write heads, erase heads, tape cleaners, and a pneumatic pad.
9. Issuing a printout or by lighting up an indicator and sounding an alarm.

10. Erase heads use a steady DC voltage to align all magnetic particles on the tape in the same direction.
11. The erase heads are not supplied current and therefore, cannot realign the magnetic particles on the tape.

**615.**

1. Data is converted to a tape code and is recorded on blank paper tape in the form of punched holes. To retrieve data, a paper tape reader detects the punched holes and converts the tape code into data usable by the computer.
2. The tape punch.
3. No.

**616.**

1. The system console.
2. A keyboard, monitor, printer, various dials, switches and alarm lights.
3. To provide a direct communication between the operator and the operating system.
4. A printed log.
5. It provides a means for an operator to communicate with the computer in an interactive or conversational manner.
6. The keyboard section.
7. The video display unit.
8. Pixels.
9. The screen's resolution capabilities and the amount of computer memory available to support the display.
10. Smart terminals and dumb terminals.
11. A dumb terminal.
12. A computer can be used to retrieve information, screen and update input data, edit tape files, perform on-line programming or debugging, and serve as a teaching tool.
13. Monitoring is a means of obtaining information about the inner workings of a system.
14. To access the computer system from locations remote from the system.
15. It allows the user to interface the computer in a conversational mode.
16. A smart terminal is considered intelligent because it has its own processor and operating system and does not have to rely solely on the central processing unit.
17. Smart terminal.

**617.**

1. Printers are used to provide a permanent visual record from a data processing system.
2. Hard copy.
3. If the printer creates an image by striking the paper with a print mechanism, it is considered an impact printer.
4. One.
5. Serial impact printers using a printwheel mechanism.

6. The dot matrix printer.
7. Versatility and low cost.
8. Line printers generally operate faster but are also more expensive.
9. Drum, chain, and matrix scanning.
10. One line of characters.
11. Dot matrix printers have higher reliability, easier maintenance, and easier font and character changes.
12. The electrostatic printer.
13. Ink jet.
14. Laser and xerographic printers.
15. The laser printer also uses an electrical charge to produce a character image.
16. Examples of passive graphic devices include plotters and microfilm developers. Examples of interactive graphic devices include light pens, joy sticks, and digitizers.
17. The vertical format unit (VFU).
18. A decollator.
19. A burster.

**618.**

1. A high-speed process of recognizing and translating machine-printed or hand-printed words, letters, symbols, and numbers into information that can be computer processed.
2. Magnetic ink-character readers are used by financial organizations to process checks at electronic speeds.
3. They are used intensively in postal systems, and for reading insurance premiums, notices, and invoices.
4. Strictly input devices.
5. Optical scan unit.
6. Marks or absence of marks are determined by the amount of light reflected from the area being scanned.

**619.**

1. Monochrome, gray-scale, or color monitors.
2. Two; one for the background, and one for the foreground.
3. Color graphic adapter (CGA), enhanced graphics adapter (EGA), video graphics array (VGA), or super video graphics array (SVGA).
4. The number of colors they support and level of resolution.

**620.**

1. Operating systems are computer programs that recognize input, send output, track files and directories, and control peripheral devices.
2. They control and direct processes, ensure other programs and users running at the same time do not interfere with each other, and also handle system security.

3. A supervisor (executive), input/output (I/O) manager, file manager, command processor, and transient utility programs.
4. The most frequently used elements are loaded into the computer's RAM and the remaining elements reside on the hard disk to be called into RAM as needed.
5. In RAM.
6. (1) a, (2) d, (3) e, (4) c, (5) b, (6) d, (7) c, (8) a, (9) b, (10) a, (11) d, (12) e, (13) e, (14) a.
7. Multi-user, multi-processing, multi-tasking, or real-time.
8. Multi-user.
9. Multi-processing.
10. The type of operating system.

**621.**

1. MS-DOS®.
2. UNIX®.

**622.**

1. The amount of available memory.
2. Within a computer's microchips.
3. Kilobytes or megabytes.
4. By installing additional memory boards.
5. Read-only memory (ROM) and random-access memory (RAM).
6. A memory unit stores binary information in the form of words.
7. A number, an instruction, a group of alphanumeric characters, or any binary coded information.
8. Through the use of two control signals and two external registers.
9. Control signals specify whether a word is to be stored or transferred out of a memory register.
10. One register specifies the use of a specific memory register and the other specifies a memory word's particular bit configuration.
11. A memory address register specifies the location of a selected memory word.
12. The memory location's address is transferred to the address register, the memory unit accepts the address from the register, and then the memory unit opens the paths needed to transfer the word into or out of memory.
13. The read control signal specifies a transfer-out function and the write control signal specifies a transfer-in function.
14. According to its volatility, mode of access, and construction.
15. Volatile memory is cleared when power is removed from the computer.
16. Random memory access is faster because memory locations can be accessed directly.
17. Volatile.
18. ROM is nonvolatile.
19. Programmable read-only memory (PROM) and the erasable programmable read-only memory (EPROM).

20. PROMs can only be programmed once while EPROMs can be erased and reprogrammed.
21. Basic operating instructions such as system input/output information, self-test routines, and system configuration information.
22. RAM temporarily loads and stores instructions and data needed while processing is occurring.
23. If RAM is not large enough to hold all the files needed to accomplish a task, then the task cannot be completed.
24. (1) c, (2) b, c, d, e, (3) b, (4) d, (5) d, (6) c, (7) a, (8) b, (9) d, (10) c, (11) a.

**623.**

1. A shortage of conventional memory.
2. You can free conventional memory by: running MS-DOS® in extended memory; streamlining your CONFIG.SYS and AUTOEXEC.BAT files so that they do not start unnecessary memory-resident programs; and by running device drivers and other memory-resident programs in upper memory.
3. You can free extended memory by: reducing the amount of memory allocated for device drivers; streamlining your CONFIG.SYS and AUTOEXEC.BAT files so as not to start unnecessary extended-memory programs; and, if MS-DOS® is running in extended memory, you can move it to conventional memory.
4. You can free expanded memory by: disabling unnecessary device drivers and programs, disabling or reducing cache-memory allocations; or by utilizing expanded-memory emulator programs.

**Do the Unit Review Exercises (URE) before going to the next unit.**

## Unit Review Exercises

**Note to Student:** Consider all choices carefully, select the *best* answer to each question, and *circle* the corresponding letter.

21. (608) Which of the following is considered the *first* true computer?
  - a. ENIAC.
  - b. Ferranti Mark I.
  - c. Analytical engine.
  - d. Manchester Mark I.
  
22. (608) What designates a location or registers where data is stored in the computer?
  - a. The hardware.
  - b. An address.
  - c. The clock.
  - d. A bit.
  
23. (608) The three principal buses in a computer are the
  - a. address bus, the data bus, and the information bus.
  - b. address bus, the data bus, and the control bus.
  - c. address bus, the byte bus, and the control bus.
  - d. charter bus, the data bus, and the control bus.
  
24. (608) A computer's internal synchronization signals are provided by its
  - a. clock.
  - b. data bus.
  - c. operating system.
  - d. instruction registers.
  
25. (608) A sequence of digits that tells the computer what operation to perform refers to
  - a. instructions.
  - b. directions.
  - c. hardware.
  - d. software.

26. (608) What is the computer term that is *generally* made of more than one byte and, in some cases, is the largest group of bits treated *as a unit* throughout the computer's central processor?
- Bit.
  - Word.
  - Program.
  - Instruction.
27. (609) The functional unit of a digital computer that is associated with printers, modems, or video screens is
- primary storage.
  - control.
  - output.
  - input.
28. (609) What is required to make interfacing functional?
- Software.
  - Hardware.
  - The control unit.
  - The input/output unit.
29. (609) Which interfacing format transfers data many bits at a time?
- Shift transfer.
  - Bus transfer.
  - Parallel.
  - Serial.
30. (609) Which functional section of a digital computer decodes instructions?
- Memory.
  - Control.
  - Input/output.
  - Arithmetic logic unit (ALU).
31. (610) What lets the central processor execute the system program without requiring the program to monitor the status of each peripheral device?
- Hardware.
  - Instructions.
  - Control words.
  - Interrupt requests.

- 
- 
32. (610) Which is the part of the CPU that identifies the location, in memory, of the next instruction to be performed?
- Control unit.
  - Decoder unit.
  - Program counter.
  - Instruction register.
33. (610) What indicates specific conditions after an arithmetic or logic operation?
- Accumulator.
  - Flag register.
  - Read-only memory (ROM).
  - Random access memory (RAM).
34. (610) A central processing unit fabricated into a single integrated circuit is called a
- microprocessor.
  - microcomputer.
  - minicomputer.
  - co-processor.
35. (611) The disk subsystem component that checks data transfer accuracy is the disk
- unit.
  - pack.
  - controller.
  - input/output unit.
36. (611) What are the two variations of disk units?
- Floating read units and floating write units.
  - Floating head units and movable head units.
  - Fixed read head units and fixed write head units.
  - Fixed read/write head units and movable read/write head units.
37. (612) The four steps for configuring hard disks include installing, formatting, partitioning, and
- repartitioning.
  - compressing.
  - addressing.
  - labeling.

38. (612) What magnetic disk configuration step divides the disk into tracks and sectors?
- Installing.
  - Formatting.
  - Segmenting.
  - Partitioning.
39. (612) You can change the size of partitions on a magnetic disk by
- repartitioning.
  - reinitializing.
  - reformatting.
  - optimizing.
40. (613) A 3.5 inch high-density magnetic disk has a storage capacity of up to
- 360 kilobytes.
  - 720 kilobytes.
  - 1.2 megabytes.
  - 1.44 megabytes.
41. (613) When we discuss optical disk units, what does WORM mean?
- Write once, read many.
  - Wide optical read mechanism.
  - Window optical read mechanism.
  - Write only revolving mechanism.
42. (614) Vacuum chambers are used on a magnetic tape unit to
- decelerate the tape.
  - accelerate the tape.
  - buffer tape movement.
  - prevent motor damage.
43. (614) Magnetic tape cartridge recorders are primarily used
- as primary storage devices.
  - in place of hard disk systems.
  - to store system back-up and database information.
  - where random access storage retrieval capability is needed.

44. (615) Which of the following is *true* about paper tape units?
- The punch is an input device.
  - The reader is an input device.
  - After the tape is punched, information can easily be changed.
  - Paper tape punches and readers are always housed in the same cabinet.
45. (615) Which of the following auxiliary storage devices cannot alter data once it has been stored?
- Floptical disk systems.
  - Magnetic disk systems.
  - Magnetic tape systems.
  - Paper tape systems.
46. (616) Which part of a computer system includes a keyboard, monitor, dials, switches, and alarm lights?
- Central processing unit (CPU).
  - Visual display unit (VDU).
  - Console.
  - Printer.
47. (616) The tiny squares that make up a picture on a visual display unit are called
- dots.
  - cubes.
  - pixels.
  - prompts.
48. (617) What is the *most common* type of printer in use today?
- Laser.
  - Serial.
  - Nonimpact.
  - Electrographic.
49. (617) Three types of impact line printers are the drum, chain, and
- matrix scanning.
  - printwheel.
  - ink jet.
  - laser.

50. (617) What is the difference between electrographic printers and electrostatic printers?
- Electrostatic printers use ordinary paper.
  - Electrostatic printers deposit an electrical charge on a drum.
  - Electrographic printers deposit an electrical charge on paper.
  - Electrographic printers deposit an electrical charge on a drum.
51. (617) What device is used on older model printers to advance paper feed?
- Burster.
  - Skipper.
  - Decollator.
  - Vertical-format unit (VFU).
52. (617) Which printer device is used to separate forms?
- VFU.
  - Burster.
  - Skipper.
  - Decollator.
53. (618) Which of the following uses a recognition process to translate machine-printed words into information that the computer can process?
- Bursters.
  - Decollators.
  - Optical read units.
  - Magnetic tape units.
54. (618) The device that reads an entire document before processing it is an optical
- scan unit.
  - mark unit.
  - page reader.
  - character reader.
55. (619) The first color graphics display system designed for personal computers was the
- gray-scale.
  - video graphics array (VGA).
  - color graphics adapter (CGA).
  - enhanced graphics adapter (EGA).

- 
56. (619) Which color graphics display system offers the highest degree of resolution?
- Video graphics array (VGA).
  - Color graphics adapter (CGA).
  - Enhanced graphics adapter (EGA).
  - Super video graphics array (SVGA).
57. (620) Each of the following is a computer operating system element *except*
- peripheral controller.
  - input/output manager.
  - transient utilities.
  - supervisor.
58. (620) The computer operating system element responsible for encoding and decoding all data transferred between programs and peripherals is the
- input/output manager.
  - command processor.
  - transient utilities.
  - supervisor.
59. (620) Multi-user operating systems are those that
- allow two or more users to run programs at the same time.
  - allow a single program to run more than one task.
  - respond instantaneously to user input.
  - use more than one processor.
60. (620) The classification of operating systems normally used for navigation applications is
- multi-processing.
  - multi-tasking.
  - multi-user.
  - real-time.
61. (621) Which of the following operating systems works only on computers with a 80286 processor or greater?
- MS-Windows®.
  - MS-DOS®.
  - UNIX®.
  - OS/2®.

62. (621) The operating system that works on personal, mainframe, and minicomputers is
- OS/2®.
  - UNIX®.
  - MS-DOS®.
  - MS-Windows®.
63. (622) The memory storage location for one word is called the
- ROM.
  - cell.
  - track.
  - sector.
64. (622) Which section of a memory unit specifies the selected memory word's location?
- Stack pointer.
  - Storage register.
  - Control register.
  - Address register.
65. (622) Which classification of memory is cleared when power is turned off?
- Volatile.
  - Non-volatile.
  - Magnetic core.
  - Magnetic disk.
66. (622) The major difference between read-only memory (ROM) and random access memory (RAM) is that ROM is
- non-volatile.
  - expandable.
  - extendable.
  - volatile.
67. (622) A computer's conventional memory is located in
- its first megabyte of memory.
  - the memory area above 1 megabyte.
  - the first 64 kilobytes of upper memory.
  - the memory area between 641 kilobytes and 1 megabyte.

68. (622) Extended memory is
- a. memory above 1 megabyte.
  - b. part of the upper memory.
  - c. the same as expanded memory.
  - d. available only on computers with 80386 processors or greater.
69. (622) What type of computer memory reduces the amount of time the computer spends reading data from the hard disk?
- a. Expanded.
  - b. Extended.
  - c. Cache.
  - d. Upper.
70. (623) If a computer program fails to run because of insufficient memory, the most likely cause is a shortage of
- a. conventional memory.
  - b. expanded memory.
  - c. extended memory.
  - d. upper memory.
71. (623) In order to use extended computer memory, you must
- a. run MS-DOS® in extended memory.
  - b. run MS-DOS® in conventional memory.
  - c. install an extended memory manager program.
  - d. load all device drivers in conventional memory.

**Please read the unit menu for Unit 3 and continue. →**

## Unit 3. Protocols, Standard Systems Architectures, and Digital Interface Devices

	<i>Page</i>
<b>3-1. Protocols</b> .....	<b>3-2</b>
624. Error control techniques .....	3-2
625. Protocol types .....	3-5
626. Asynchronous modem protocols .....	3-23
<b>3-2. Standard Systems Architectures</b> .....	<b>3-28</b>
627. Open System Interconnect (OSI) .....	3-29
628. Standard Network Architecture (SNA) .....	3-34
<b>3-3. Digital Interface Devices</b> .....	<b>3-43</b>
629. Systems interface devices .....	3-43
630. Access interface devices .....	3-51

**A** COMPUTER network in its simplest form is a system of computers, terminals, and peripherals linked together in some manner by a series of telecommunications circuits. This has developed from the experimental systems of what seems like just a few years ago to become a central issue in computing at all levels—from small personal work stations to large mainframe computers and powerful desktop microcomputers. The long-established form of computer networks that the military is incorporating retains most of the processing power in a central location and uses a telecommunications network to allow remote users to access the central services.

Most computer networks extend over several geographically separate sites, and consequently, use military circuits or circuits belonging to, or leased from, a commercial agency to interconnect them. By knowing the background of standard network architectures and having a basic knowledge of protocol and equipment interface requirements, maintaining the communications system used can be simplified. This unit provides you with a look at these areas to help you better understand our computer communications networks.

## 3-1. Protocols

Let us imagine, for a minute, that no traffic laws exist. There are no red lights, stop signs, or railway signals. People can drive on any side of the road and as fast as they wish. That will not work very well, will it? We need laws or rules to ensure that traffic moves safely and efficiently from one point to another. The same is true for data communications. In data communications, these rules, called *protocols*, ensure that data is passed from one terminal to another without error and in the most efficient means possible.

This section is devoted to describing asynchronous protocols and two of the most common layer 2 protocols: IBM's character-oriented binary synchronous communications (BSC) and IBM's bit-oriented synchronous data link control (SDLC). Also introduced are the two standard packet-oriented protocols, the Internet Protocol (IP) and Transmission Control Protocol (TCP) which interface upper layer and subnetwork layer protocols. BSC was first introduced in 1964 and is the most pervasive protocol in the industry. Newer, more powerful protocols are becoming increasingly popular. SDLC is a member of this group of protocols.

Two other members of this new group are HDLC (high-level data link control) and ADCCP (advanced data communication control procedure). Although these new protocols vary in their versatility and capability, they are similar in principle and understanding. SDLC is a good springboard to understanding other protocols. There are many protocols available for many uses, but most provide some form of error control. For this reason, we look at some of the ways error control is performed.

### 624. Error control techniques

One of the most important functions of a protocol is error control. You should remember that error control is simply an *attempt* to provide faultless communication. Errors inevitably slip through but, with certain techniques, we can catch nearly all of them.

Error detection is usually done by one of the following four methods depending upon the functions and code: (1) vertical redundancy check (VRC); (2) longitudinal redundancy check (LRC); (3) checksum; and (4) cyclic redundancy check (CRC).

**Vertical redundancy check (VRC).** As each incoming character is received, VRC checks it for odd parity. (BSC ASCII uses odd parity.) Odd parity means that there is always an odd number of "1" bits in the bit pattern for each character. Outgoing characters are checked prior to transmission for odd bit count. If the count is even, the transmitter inserts a "1" bit in the parity bit position to make it odd. Thus, all characters transmitted have an odd bit count. A receiver detects an error when a character contains an even number of "1" bits.

Figure 3-1 illustrates both VRC and LRC checking. With VRC, the "1" bits are added vertically and the parity bits are added at the bottom. For example, the vertical row on the far left is 1100101. This character has an even number of "1" bits so a "1" parity bit was added to make the character odd. Note the third vertical row from the left. Because the character 1010001 has an odd number of "1" bits already, a "0" parity bit was added so it would remain odd. VRC is about 90 percent effective in error detection.

LONGITUDINAL CHECK									
V	1	0	1	0	0	1	1	0	1
E	1	0	0	1	0	1	0	0	0
R	0	1	1	0	0	0	0	0	1
T	0	0	0	1	1	1	0	1	1
I	1	0	0	0	1	0	0	1	0
C	0	0	0	1	1	0	1	0	0
A	0	0	0	1	1	0	1	0	0
L	1	1	1	0	0	1	1	0	0
C	1	1	0	0	1	1	0	0	0
H	1	1	1	0	0	1	1	0	0
E	1	1	1	0	0	1	1	0	0
C	1	1	1	0	0	1	1	0	0
K	1	1	1	0	0	1	1	0	0
K	1	1	0	0	0	1	0	1	0
VERTICAL PARITY BITS									

Figure 3-1. VRC/LRC checking (odd parity).

**Longitudinal redundancy check (LRC).** Where VRC checks characters for odd parity, LRC checks an entire horizontal line within a block for odd parity, the count being made at both the transmitter and receiver. This method is usually done in addition to VRC to improve error control. When the control characters ITB, ETB, or ETX appear, indicating the end of a block, this count becomes the BCC (block check character) and is transmitted to the receiver. In figure 3-1, the extra bits, where required, are shown on the right of the illustration. Again, if we look at the top row of bits, we can see 10100110. This obviously has an even number of "1" bits so the "1" parity bit was added to make it odd.

At the receiver, the BCC count is compared with the transmitter's BCC count. When they are equal, the previous block was error-free. When they are unequal, the receiver requests a retransmission of the previous block. VRC/LRC is only available with the ASCII character set. It is not available with EBCDIC. VRC, when combined with LRC, is about 98 percent effective.

**Checksum.** If an extended ASCII code is used, there is no parity bit to use for error control. The extra bit is part of the code. For this reason, another error detection method was developed.

There are different methods for calculating a checksum, but we will look at the method used by one of the most popular PC to PC protocols, XMODEM.

XMODEM sends data in blocks of 128 characters. These characters are put one on top of the other and added together just like any addition problem. The answer is then divided by the binary equivalent of the number 255. Now, we do something that may seem a little strange. After doing all this, we disregard the answer. That's right, we disregard the answer, but we keep the remainder and that becomes our checksum. Here is a simplified example using decimal numbers rather than binary.

476	<i>Answer ignored</i>	
773		255)2846
901		<u>2550</u>
+696		296
2846		<u>255</u>

41 *is the remainder and  
becomes the checksum*

**Cyclic redundancy check (CRC).** The CRC concept treats the binary string of ones and zeros contained in the frame address, control, and information fields as a single, long binary number and uses that number as a dividend in a division problem. The dividend is divided by the binary divisor represented by a standard binary sequence. This digital operation results in a binary quotient and a remainder (which, because of the size of the divisor, may contain up to 16 bits).

This division process takes place in the transmitter at the sending end of the link. The 16-bit remainder is attached to the frame (packet) as the BCC (block check character). When the frame is received at the distant end, the BCC is divided by the same divisor as used by the transmitting end. The remainder computed at the receiver is compared to the BCC received with the data over the link. If they are the same, it is presumed that the data was correct. If they differ in any bit, it is assumed that an error was made in transmission, and the receiver asks the transmitter to repeat the errored frame (packet). This is an example using decimal numbers instead of binary numbers:

#### AT THE TRANSMITTER

MESSAGE = 108436 = DIVIDEND

DIVISOR = 41

DIVIDEND/DIVISOR = 108436/41

QUOTIENT = 2644

REMAINDER = 32

TRANSMITTED MESSAGE = 10843632

**AT THE RECEIVER**

RECEIVED MESSAGE = 10833632

DIVIDEND/DIVISOR = 108336/41

QUOTIENT = 2642

REMAINDER = 14

Since the remainder of the received message (14) does not equal the remainder of the transmitted message (32), an error has occurred. Also, remember that the quotient itself is not transmitted since this would effectively double the number of bits transmitted.

CRC is more effective than parity or checksum for a data block greater than 512 characters. It can be applied in either the hardware or software and is about 99 percent effective in most applications.

**625. Protocol types**

Protocols generally fall into one of three categories: character-oriented, bit-oriented, or packet-oriented. Although all of them are used to achieve the same goal (getting a message from a source to its destination), they each perform different services throughout a transmission path. For message traffic to successfully reach its destination, it may have to interface more than one of these protocols.

**Character-oriented protocol (COP).** BSC (binary synchronous communications) is a character-oriented protocol. It uses characters to control the link and to represent information. With BSC and other line protocols of this type, "handshaking" is the term commonly used to describe the interaction between stations. Typically, the following information is exchanged:

- Message available for transmission.
- Start of text transmission.
- Acknowledgment or rejection of the text.
- Detection of errors.
- Retransmission after error detection.
- End of transmission.

**Handshaking sequence.** A simplified handshake sequence is summarized in figure 3-2. Before transmitting any control/message block, the sending location transmits normally three or four synchronizing characters (SYN) to synchronize the two locations. A message exchange is initiated when a location sends an enquiry (ENQ) to another location. If the other location can accept a message, it acknowledges (ACK) the enquiry. Throughout the handshake sequence, each acknowledgment is alternately numbered one and zero.

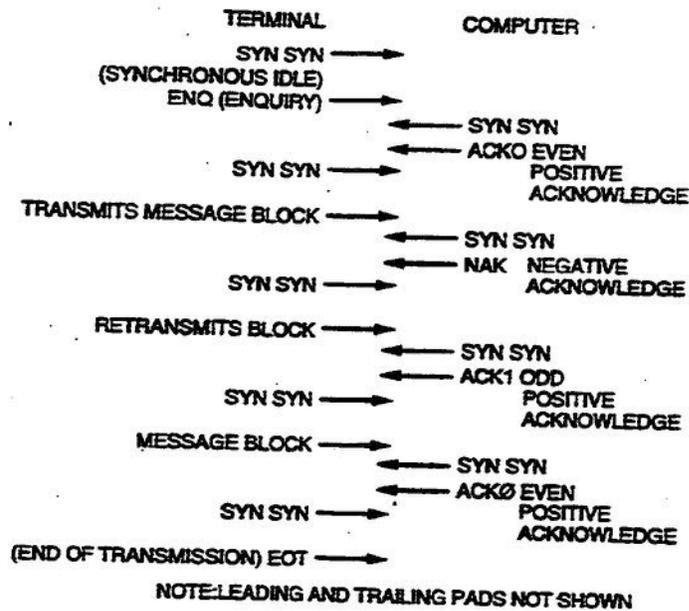


Figure 3-2. BSC handshaking sequence.

As shown in figure 3-2, SYN followed by ENQ is transmitted to synchronize and initiate the exchange; the response is ACK 0, even positive acknowledgment. The sender then transmits SYN characters followed by the message block.

When the computer looks at the message block, it detects an error and transmits a negative acknowledgment (NAK). The terminal then retransmits the message block. This time it is error free, and the computer transmits ACK 1, each message block, and the terminal sends SYN followed by the message. The transmission is error free and the computer responds with ACK 0, even positive acknowledgment. Since the terminal has no more messages to transmit, it sends the end of transmission (EOT) character. Unless the computer has something to transmit to the terminal, this completes the exchange and the computer disconnects from the terminal. The use of odd and even ACKs provides a sequential check for the series of replies.

The line protocol characters used in the above example represent only a few of the BSC control characters. Table 3-1 lists the BSC control characters along with their meanings and functions.

**Message block.** A message block (frame) format is shown in figure 3-3. Each transmission in BSC can contain up to three elements:

- (1) A header.
- (2) The text.
- (3) A trailer (BCC).

The control characters used to identify these elements are as follows:

- SOH, indicating the header follows.
- STX, indicating the text follows.

- ETX, indicating the end of text, or ETB, indicating the end of the message block with the trailer following.

Character	Meaning	Functions
SYN	Synchronizing Character	Establishes and maintains character synchronization prior to the message block and during transmission. Also used as time fill in the absence of control characters and data.
STX	Start of Text	Transmitted before the first data characters.
ETB	End of Transmission Block	Indicates the end of the text block starting with STX or SOH. BCC is sent after ETB, requiring the receiver to respond with ACK, NAK or optionally WACK or RVI.
US/TS	End of Intermediate Transmission Block	Divides a message for error checking purposes without the turnaround required by ETB. BCC follows ITB and resets the block check count to zero. STX or SOH is not required for following text blocks, but STX is required if a header is followed by text.
ETX	End of Text	Terminates a block begun with SOH or STX and the end of a sequence of blocks. BCC immediately follows ETX, requiring a receiver status reply.
EOT	End of Transmission	Concludes transmission, resets all stations to control mode (neither transmitter nor receiver). Also a non-transmit response to a poll and an abort signal for a malfunction.
ENQ	Enquiry	Bids for the line in a point-to-point and multipoint connection and requests last acknowledgment retransmission or a preceding block to be ignored.
ACK	Affirmative Acknowledgment	Previous block accepted and error-free, receiver ready for next block. Also a positive response to selection (multipoint) or line bid (point-to-point).
SOH	Start of Heading	Transmitted before the header characters. These contain information such as the routing and priority of the message.
NAK	Negative Acknowledgment	Previous block unacceptable and retransmission required. Also a negative response to a selection or line bid.
TTD	Temporary Text Delay	Transmitter not ready to commence transmission but wants to maintain connection. Sent two seconds after message received to avoid three second timeout, also initiates an abort.
RVI	Reverse Interrupt	Sent to a transmitter by a receiver in place of ACK indicating the receiver has a high priority message waiting transmission.
WACK	Wait Before Transmit Positive Acknowledgment	Previous block accepted and error-free, but receiver not ready for next block. Will continue to respond with WACK until ready to receive. Also a positive response to a text or heading block selection sequence (multi-point), line bid (point-to-point), or identification line bid sequence (switched network).
DLE	Data Link Escape	Prefix for control characters during transparent mode. Control characters have no control meaning unless prefixed by DLE.
DLE EOT	Disconnect Sequence for a	Transmitted on a switched line when all message exchanges are complete. Can be transmitted at any time to cause a disconnect.
Pad ( )		Added before (leading pad) and after (trailing pad) a transmission. This ensures the first character is not sent until the other station is prepared to receive, and the last character is properly transmitted before turnaround is initiated or the transmitter turns off.

Table 3-1. BSC Control Characters.

As previously described, two or more SYN characters are used to establish character synchronization between sender and receiver. The number of SYN

characters used varies with different networks and applications. The three shown in figure 3-3 are one accepted pattern. The message block follows the SYN characters.

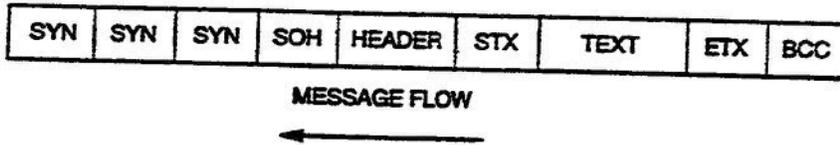


Figure 3-3. BSC message block format.

A message consists of one or more blocks of information. Each block contains a text and trailer; the first block contains the header. Typically, a header contains a character, or characters, that identify the originating or receiving location and the priority. This could be called the address. A SOH character identifies the characters that follow as the header.

The text portion of the message block is identified by a preceding STX character. A short message may only require a single block, but a long message is broken into a number of blocks. In the example shown in figure 3-3, the message consists of a single block and is followed by the end of text (ETX) control character. In a multi-block message (fig. 3-4), the last block of text is followed by ETX. The trailer consists of the block check character (BCC). This character contains a count for error checking. As the block is transmitted, both the sender and receiver generate a count from the block. At the end of the block, the receiver compares his or her block count with the sender's BCC character. If the two do not agree, a NAK character is transmitted to the sender, requiring the block to be retransmitted. If the error persists, a preset number of attempts is made to obtain an error-free block, then the transmitter aborts.

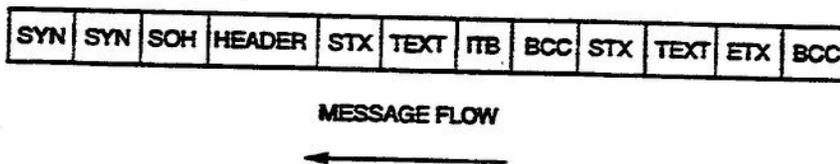


Figure 3-4. A long multi-block message.

**Long messages.** Long messages are broken into a series of blocks for transmission, as illustrated in figure 3-4. Each block of text, except the last, is followed by an end of transmission block (ETB) character or an end of intermediate transmission block (ITB) character. ETB requires a response from the receiver and causes line turnaround and the BCC to be sent and compared. ITB divides the message for error-checking purposes and does not require a response from the receiver. After the first ITB, an SOH is not required before each block of text. The last intermediate block is followed by an ETX or ETB character. As each intermediate block arrives, its BCC is compared with the receiver's BCC. If an error is detected in any intermediate block, no action can be taken until ETB is received, then all intermediate blocks must be retransmitted.

**Data transparency.** If we want to transmit raw data as text as in a real situation, we cannot know what characters are being transmitted. In the previous discussion of control characters, no text could have included control characters since the receiver would detect it as a control character and not text. The transparent mode removes the code restriction for line and message control characters and allows transmitting many forms of raw data within the standard message format. This provides greater versatility in the variety of coded data that can be handled. Raw data that contains a character identical to a control character is recognized as data (text) by the receiver. It does not see the (transparent) data as a control character.

However, some type of character is required to delineate control characters and information characters. This is accomplished by prefixing the previously discussed control characters with a data link escape (DLE) character. Only when the DLE prefix is present does the receiver recognize a control character.

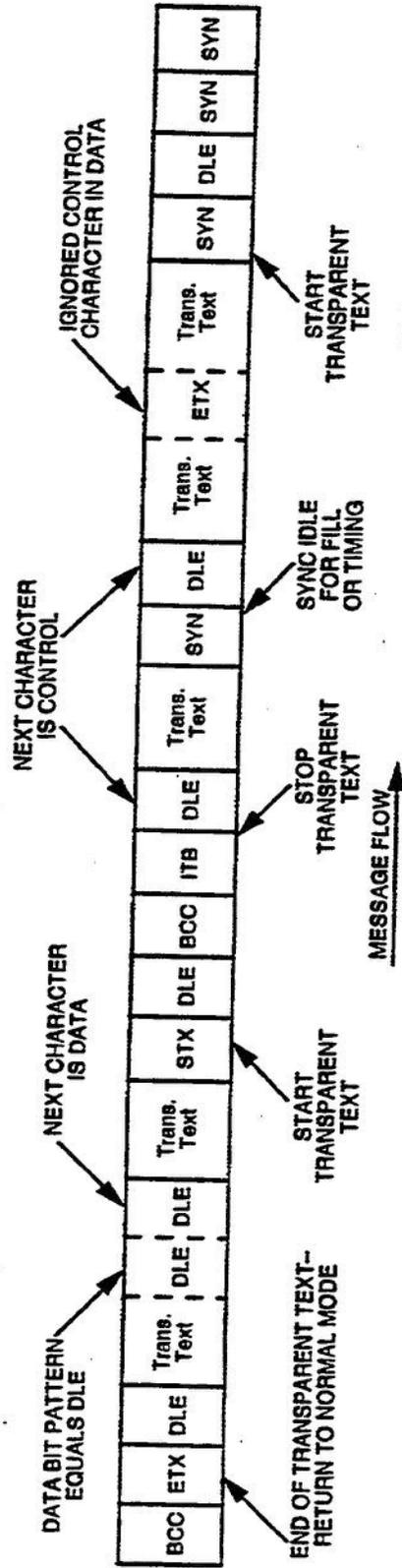
Figure 3-5 (next page) illustrates the format for blocks of transparent text. The transmission is initiated and synchronized as before, but DLE STX is transmitted instead of STX. Until the receiver sees DLE STX, the receiver handles the text as pure binary data and ignores control characters unless they are prefixed by DLE. When a DLE appears, the control character that follows is recognized and the appropriate action performed. When a binary bit pattern equals DLE, a DLE is inserted before it to show that the second DLE is data and not the start of a control sequence. The receiver discards the first DLE and recognizes the second as data.

All replies, enquiries, and headers are transmitted in normal mode. SYN SYN SYN appears before the message to establish synchronization, and DLE SYN may occur in the message as required for fill or time-out purposes. SYN SYN SYN may also appear after DLE ITB to establish synchronization prior to the next message block. If the next intermediate block is transparent, it must start with DLE STX. Table 3-2 (page 11) shows the control characters used in transparent mode. Note that DLE ETB and DLE ETX cancel transparency.

**Retransmission.** Normally, a receiver replies to a transmission with ACK—data accepted, continue sending. When an error is detected, the response is NAK—data not accepted, retransmit previous block. Retransmission of the previous block is normally attempted three or four times. When the transmitter sends a message block and receives no reply or a garbled reply, the transmitter can send ENQ requesting a retransmission. When there is no reply, a time-out occurs 3 seconds after the last response from the receiver.

**Error checking.** While error checking is often assumed to be a bit counting process, BSC actually uses three error detection methods:

- (1) Format detection — a check for control characters in proper sequence.
- (2) Time-out sequences — a check for continued transmission to prevent an indefinite tie-up.
- (3) Transmission error detection — a check that the received bit count equals the transmitted bit count.



CDC3C251A04-9409-038

Figure 3-5: Transparent text message blocks.

<b>DLE STX</b>	Starts the transparent mode for the following message
<b>DLE ETB</b>	Terminates a transparent block, returning the data link to normal mode, and requests a reply.
<b>DLE ETX</b>	Terminates the transparent text, returning the data link to normal mode, and requests a reply.
<b>DLE SYN</b>	Maintains sync and provides a time fill to prevent a timeout.
<b>DLE ENQ</b>	"Disregard this block of transparent data." Returns the data link to normal mode.
<b>DLE DLE</b>	A data bit pattern may appear in the transparent text that is identical to a DLE. A DLE character is inserted to identify the data bit pattern. The first DLE is discarded and the second is treated as data.

Table 3-2. BSC Transparent Mode Control Characters.

**Bit-oriented protocol (BOP).** Whereas character-oriented protocols are based on characters, bit-oriented protocols are based on bits and the position of the bits. The most widely used bit-oriented protocols in the industry are IBM's SDLC and HDLC. Architecturally, BOP differs from COP in that it uses positional significance of the bits, not control characters. For example, in COP, a start of header (SOH) is required after SYN is transmitted. SOH informs the receiver that a header is following soon after. In BOP, after the 8 synchronizing bits have been transmitted, the receiver knows that the next set of bits to follow is the header information. In BOP, information is sent by frames as well, but within each frame are fields that have specific functions.

All transmissions in a link follow the frame format shown in figure 3-6. A frame always contains control characters, but may not contain information (data). All frames are numbered in sequence, and each frame contains the sender's count of frames transmitted to a specific station along with the number of the next frame the sender expects to receive from that station. The frame consists of the following fields:

- **Flag** — the opening flag for the start of a frame.
- **Station Address** — identifies the outlying station that is in communication with the primary.
- **Control** — used by the primary to control secondary operation and by the secondary to respond to the primary.
- **Information** — the field containing the data to be transmitted without constraints on length or bit patterns.
- **Frame check sequence (block check)** — used to detect transmission errors.
- **Flag** — the closing flag for the end of a frame.

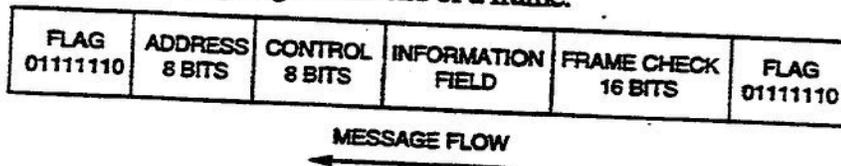


Figure 3-6. BOP frame format.

**Flag.** With a protocol that uses positional significance, not control characters, to identify the various elements of a message, the start of a frame sets the starting position for this bit stream. The flag field is the first character of a frame; the receiver uses it to count down the incoming bit stream to identify the fields within the frame.

The flag is a unique sequence of bits which never appear in a frame other than at the start and end of it. As shown in figure 3-6, the flag consists of 8 bits—a 0, six 1's, and a 0.

Zeros are inserted and deleted as required to prevent a flag bit pattern from appearing within the frame. This hardware function is called bit insertion or bit stuffing. When five 1's appear, a 0 is inserted in the bit stream after the last 1. The receiver detects the five 1's followed by a 0 and deletes the 0. The inserted and removed zeros are not included in the frame check sequence.

Each station in the data link is continuously looking for the flag bit sequence followed by the station's address. During lulls in message flow, a series of flags can be transmitted to keep the link active and synchronized.

**Station address.** Before a secondary station can accept a frame, it must recognize its valid address in that frame. A primary station only accepts frames with addresses of those secondaries the primary has authorized to transmit. The address field always contains the addresses of secondary stations, never those of a primary station.

**Control field.** The 8-bit control field contains the commands and responses required for control of a data link. The primary station uses the field to command a secondary that is selected by the address field to perform an operation. A secondary station uses it to respond to the primary. The control field has three formats that indicate the contents and purpose of the frame:

- (1) Information — the frame contains information.
- (2) Supervisory — the frame contains commands and responses.
- (3) Unnumbered — the frame contains commands and responses and may contain count information. Frame sequences are not counted.

**Control field information format.** This information format is used when data is transmitted between primary and secondary stations. It contains the count of frames sent ( $N_s$ ), the count of frames received ( $N_r$ ), and tells if the frame is polling (P) or a final (F) frame in a sequence.  $N_s$  and  $N_r$  are used for error checking by frame and are generated by both primary and secondary stations. P is used by a primary station to poll a secondary station and initiate transmission. F is used by a secondary station to inform a primary station when a frame is the last one in a transmission.

$N_s$  represents the number of the frame currently being transmitted from either a primary or secondary station. It tells the receiver that "the number of this frame that I am now sending you is  $N$ ."  $N_r$  represents the number of the frame that the sender next expects to get from the receiver. It tells the receiver that "the frame you sent me prior to  $N$  was error free, and your number of the next frame you send me will be  $N$ ."

This method of frame sequence counting provides for the detection of duplicated, missing, or erroneous frames. Frames detected as incorrectly received are retransmitted.

A 3-bit control field allows up to eight (0 to 7) frames to be transmitted without a response from the receiver; then, transmission must halt until the receiver acknowledges correct reception of the frames. Burroughs data link control (BDLC) can have a 9-bit control field, which extends the potential number of unacknowledged frames to 127. The sender stores unacknowledged frames in buffers (for possible retransmission) until they are acknowledged as being correctly received.

When a frame is received at a station, the  $N_s$  is compared with the station's  $N_r$ , the frame count the station is expecting to receive. If they are equal and the frame check sequence does not indicate an error, the frame is accepted and the station's  $N_r$  is updated. The  $N_r$  in the received frame is compared with the station's  $N_s$  (the count of

the last frame transmitted by the station); if they are equal, indicating correct transmission, the station clears its buffers. An incorrect transmission is corrected by retransmission of the frame.

When a primary station polls (P) a secondary station, it requests a response or a series of responses from the station. Final (F) is sent by the secondary station with the final frame transmitted as a response to a poll command. If a final frame is not accepted by a primary or secondary station, the primary station performs a time-out and waits for a response to the poll. When there is no response, it again polls the secondary station.

*Control field supervisory format.* A frame with a supervisory format in the control field contains no information. This format is used to regulate traffic and request retransmission. The frame format is the same as shown in figure 3-6, except the information field is interpreted to be of zero length.

These commands/responses are interspersed with information frames to supplement traffic control. The three command/responses are as follows:

- (1) *Receive Ready (RR).* When a station has received error-free information frames through the frame prior to Nr, it sends RR. This indicates it is ready to receive Nr.
- (2) *Receive Not Ready (RNR).* RNR is sent by the receiving station to indicate a busy condition. For example, this could happen when the station's buffers are full, and they cannot accept frames requiring buffer space. Frames with counts prior to Nr are accepted, but frame number Nr and subsequent frames cannot be accepted.
- (3) *Reject (REJ).* REJ is transmitted to request retransmission of a sequence frame or frames. This occurs when a frame is received out of sequence or contains an error in the frame check sequence. When a frame is received with an Ns count not equal to the station's Nr count, all frames prior to Nr are accepted, but the Nr frame is not.

No more than one REJ can be outstanding at any given time. REJ is transmitted only once for each incorrect condition. The condition is cleared when the receiving station accepts an information frame with an Ns count equal to the station's Nr count.

*Control field unnumbered format.* A station's Ns and Nr counts are not changed when the control field uses the unnumbered format, but the frame can contain information without regard to the send and receive sequence counts. Commands that change modes reset the frame sequence count to zero. Of the commands and responses previously discussed, only P and F are contained in this format. Table 3-3 contains a list of commands and responses available with SDLC, in addition to P and F.

## SDLC CONTROL FIELD NON-SEQUENCED FORMAT COMMANDS AND RESPONSES.

MNEMONIC	MEANING	FUNCTION
NSI	Nonsequenced Information	A command response to identify a nonsequenced information frame. It is not acknowledged.
SNRM	Set Normal Response Mode	A command to set the receiving secondary station into normal response mode. It stays in this mode until it receives a DISC or SIM command. The expected response is NSA. The Ns and Nr counts are set to zero. no unsolicited transmission is allowed.
DISC	Disconnect	This command places the receiving station off-line and terminates other modes. It stays off-line until an SNRM or SIM command is received. The expected response is NSA. While disconnected, the secondary station cannot receive or transmit information frames.
NSA	Nonsequenced Ack	The affirmative response to SNRM, DISC, and SIM. The primary station controls further transmission from the secondary station.
RQI	Request for Initialization	A response from a secondary station requesting a SIM command from the primary station. It is repeated if any command other than SIM is received.
SIM	Set Initialization Mode	A command to start certain procedures at the secondary station that will initialize the data link functions. The expected response is NSA. Nr and Ns counts in both stations are reset to zero.
ROL	Request On-Line	A response from a secondary station to show it is disconnected.
ORP	Optional Response Poll	A command inviting transmission from the addressed secondary station. A frame with ORP cannot contain information.
CMDR	Command Reject	A response from a secondary station in normal response mode indicating an incorrect command or that it cannot execute a correct command. It cannot act upon the command that caused the condition; in further responses it repeats CMDR, except when a mode-setting command is received. The frame containing CMDR has an information field containing the status information needed by the primary station to correct the condition.
GA	Go Ahead	A command used in the loop configuration that sequentially requests each station in the loop to transmit if it has anything to transmit. ORP GA is selective in the stations it asks to transmit, but still travels around the loop.

Table 3-3

Some examples of non-sequenced command functions are commanding a secondary station in a switched network to go off-hook, exchanging identification between primary and secondary stations, and polling secondary stations without changing frame sequence numbers. This format provides for data link management, activating and initializing secondary stations, reporting of procedural errors not recoverable by retransmission, and controlling response of secondary station.

**Information field.** An information frame is the vehicle for moving data between stations. It is unrestricted in content and format, and the line protocol does not recognize its contents because of transparency. However, buffering considerations and frame check sequence capacity may place practical limits on very long information fields. Information format frames are the only frames with both  $N_s$  and  $N_r$  counts. The particular control format transmitted in a frame identifies when the frame contains information.

The length of each information field is variable. In a single multi-frame transmission, each frame may have different lengths. Any type of code can be conveyed including EBCDIC, ASCII, baudot, binary coded decimal, packed decimal, and straight binary. The specific code used is identified as the operational code of the station address by the address field.

With no restriction on bit patterns, it is possible for 6 sequential bits to be 1's, simulating a flag bit pattern which the line protocol sees (non-transparent) as a closing flag to terminate the frame. This is prevented by the transmitter inserting a 0 after the fifth 1. The inserted 0 is then deleted by the receiver.

**Frame check sequence (block check) field.** This field is 16 bits in length and precedes the closing flag. When information is present in the frame, it follows the information field; otherwise, it follows the control field. Its purpose is to detect errors that occur during transmission.

All bits appearing between the opening and closing flags are included in the checking accumulation, except for the 0 bits that are inserted and deleted to prevent false closing flags. Both transmitter and receiver perform a cyclic redundancy check on the frame. The transmitter sends its computations to the receiver in the frame check sequence field. The receiver compares the computation with its own computation, and, if they are equal, the frame is error free. When they are not equal, the receiver does not accept the frame, does not advance its  $N_r$  count, and requests retransmission of the frame at the first opportunity.

**Closing flag.** The closing flag has the same bit configuration as the opening flag. It terminates the frame and completes the frame sequence check.

**Packet-oriented protocols.** These protocols govern the characteristics for transmitting messages through packet switching networks. In packet switching, messages are broken down into smaller groups of data and control characters, called "packets," which are individually addressed and transmitted through the

networks. Network channels are occupied only during actual transmission of the packets and then released to other traffic. The DoD standard packet-oriented protocols are the Internet Protocol (IP) and the Transfer Control Protocol (TCP).

**Internet Protocol (IP).** The Internet Protocol is designed to interconnect packet-switched communication subnetworks to form an internetwork (internet). The IP transmits blocks of data, called internet datagrams, from sources to destinations throughout the internetwork. These message sources and destinations, called hosts, may reside within the same subnetwork or connecting subnetworks.

**Applications.** The IP is limited to provide only the basic functions necessary to deliver a block of data. Each datagram is handled independently of any other datagram. To provide this service, the IP receives information from an upper layer protocol, packages it as an internet datagram, and passes it to a local subnetwork protocol to be sent across the local subnetwork. If the destination host is on the same subnetwork, IP sends the datagram through the subnetwork directly to its destination. If the destination is a host on a different subnetwork, IP sends the datagram to a local gateway. The gateway then sends the datagram through the next subnetwork to the datagram's destination host, or to another gateway, if necessary. Thus, datagrams move from one IP module to another through interconnected subnetworks until they reach their destinations.

When a datagram is being transmitted, it may pass through a subnetwork whose maximum packet size is smaller than the datagram size. If this happens, the IP provides that the datagram can be fragmented into smaller units, called datagram fragments, and reassembled into its original form by the IP module at the destination host.

**Upper layer services.** In order for the IP to perform data transmission, it must interface with upper and lower layer protocols. This interfacing involves providing certain services to the upper layers and receiving certain services from the lower layers. The upper layer services that the internet protocol provides are internetwork datagram service, delivery service, virtual network service, and error reporting.

- a. **Datagram service.** The Internet Protocol provides a datagram service between similar upper layer protocols in an internetworking environment. A datagram service is characterized by data delivery to the destination with non-zero probability; which means some data may possibly be lost or duplicated. The IP datagram service does not necessarily present data at its destination in the same sequence it was supplied by the source.
- b. **Delivery service.** IP delivers received data to a destination upper layer protocol in the same form as sent by a source upper layer protocol. IP discards datagrams when it lacks sufficient resources for processing them. IP does not, however, detect datagrams lost or discarded by the subnetwork layer. As part of the delivery service, IP insulates upper layer protocols from subnetwork-specific characteristics. For example, IP maps internetwork addresses supplied by upper layer protocols into local

addresses used by the local subnetwork. IP additionally hides any packet-size restrictions of subnetworks along the transmission path within the internetwork.

c. *Virtual service.* IP provides upper layer protocols the ability to select virtual network service parameters. For this function, it provides a general set of commands for the upper layer protocols to indicate the services desired. Thus, the upper layer protocols can select certain properties of the internet protocol and the underlying subnetworks in order to customize the transmission service that best fits their needs. Virtual network services available include subnetwork service quality parameters and service options. Subnetwork service quality parameters influence the transmission service provided by the subnetworks. These parameters include the following:

- Precedence: attempts preferential treatment for high importance datagrams.
- Transmission mode: datagram versus stream. Stream mode attempts to minimize delay and delay dispersion through reservation of network resources.
- Reliability: attempts to minimize data loss and error rate.
- Speed: attempts prompt delivery.
- Resource tradeoff: indicates relative importance of speed versus reliability.

Subnetwork service *options* include the following:

- Security labeling: identifies datagram for compartmented hosts.
- Source routing: selects set of gateway IP modules to visit in transit.
- Route recording: records gateway IP modules encountered in transit.
- Stream identification: names reserved resources used for stream service.
- Time-stamping: records time information.
- Don't fragment: marks a datagram as an indivisible unit.

d. *Error reporting service.* IP provides error reports to the upper layer protocols indicating errors detected in providing the above services. In addition, certain errors detected by lower layer protocols are passed to the upper layer protocols. These reports indicate several classes of errors including invalid arguments, insufficient resources, and transmission errors. The errors that IP must report to upper layer protocols are to be determined for each implementation.

*Lower protocol layer provided services.* The required interface services to be supplied to IP from lower layer protocols are transparent data transfer between hosts within a subnetwork and error reporting.

a. *Data transfer.* The subnetwork layer protocol must provide a transparent data transfer between hosts within a single subnetwork. Only the data to be delivered, and the necessary control and addressing information, are required as input from the IP. Intranetwork routing and subnetwork

operation is handled by the subnetwork layer itself. Data may not necessarily arrive in the same order as it was supplied to the subnetwork layer, nor is data guaranteed to arrive error free.

- b. *Error reporting.* The subnetwork layer provides reports to IP indicating errors from the subnetwork and lower layers as feasible. The specific error requirements of the subnetwork layer are dependent on the individual subnetworks.

*Basic model of IP operation.* The following illustrates how the internet protocol supports a datagram transmission from one upper layer protocol to another. In this example an upper layer protocol in Host A is sending data to a like protocol in Host B on another subnetwork. In this case, both the source and destination hosts are on subnetworks directly connected by a gateway.

- 1) Within Host A, the sending upper layer protocol passes its data to the internet protocol module, along with the destination internet address and other parameters.
- 2) The IP module prepares an internet protocol header and attaches the upper layer protocols data to form an internet datagram. Then, the IP module determines a local subnetwork address from the destination internet address. In this example, it is the address of the gateway connecting to the destination subnetwork. The internet datagram, along with the local subnetwork address, is passed to the local subnetwork protocol interfacing the gateway.
- 3) The subnetwork protocol creates a local subnetwork header and attaches it to the datagram forming a subnetwork packet. The subnetwork protocol then transmits the packet across the local subnetwork to the gateway.
- 4) The packet arrives at the gateway that connects the two subnetworks. Here, the local subnetwork header associated with Host A is stripped off and the remainder of the packet is passed to the gateway internet protocol module.
- 5) The IP module determines from the destination internet address in the IP header that the datagram is intended for a host in the subnetwork associated with Host B. The IP module then derives a local subnetwork address for the destination host. That address is passed along with the datagram to the subnetwork protocol for that local subnetwork for delivery.
- 6) The destination subnetwork's subnetwork protocol builds a local subnetwork header and appends the datagram to form a packet for the destination subnetwork. The packet is transmitted across this local subnetwork to the destination host, B.
- 7) The subnetwork protocol of the destination host strips off the local subnetwork header and hands the remaining datagram to the destination IP module.

- 8) The destination IP module determines that the datagram is intended for an upper layer protocol within this host. The data portion of the datagram and information from the IP header are passed to this upper layer protocol. Delivery of data across the internetwork is complete.

This discussion does not include all aspects of the Internet Protocol. Some of its particulars, such as internet naming and addressing, are covered later.

***Transfer Control Protocol (TCP).*** The Transfer Control Protocol is a reliable, ordered, full-duplex, flow controlled connection-oriented transport protocol for use in packet-switched communication networks and internetworks. TCP appears in the DoD protocol hierarchy at the transport layer and is designed to support a wide range of upper layer protocols so they can channel continuous streams of data to like upper layer protocols. TCP breaks down the data streams into portions which are encapsulated together with appropriate addressing and control information to form segments (the unit of exchange between like TCPs). In turn, TCP passes these segments to the DoD protocol hierarchy network layer for transmission through the communication system to the distant TCP.

***Network layer provisions.*** The network layer provides for data transfer between hosts attached to a communication system. Such systems may range from a single network to interconnected sets of networks forming an internetwork. The minimum required data transfer service is limited; data may be lost, duplicated, misordered, or damaged during transmission. As part of the transfer service, the network layer must provide global addressing, handle routing, and hide network-specific characteristics. As a result, upper layer protocols (including TCP) using the network layer may operate above a wide range of subnetwork systems. Additional services the network layer may provide include selectable levels of transmission quality such as precedence, reliability, delay, and throughput. The network layer also allows data labeling, needed in secure environments, to associate security information with data.

***TCP design and mechanisms.*** TCP was specifically designed to operate above the Internet Protocol (IP). Originally, TCP and IP were developed as a single protocol providing resource sharing across different packet networks. The need for other transport protocols to use IP's services led to their specification as two distinct protocols.

TCP builds its services on top of the network's potentially unreliable ones with mechanisms such as error detection, positive acknowledgments, sequence numbers, and flow control. These mechanisms require certain addressing and control information to be initialized and maintained during data transfer. This collection of information is called a TCP connection. The following describes the purpose and operation of the major TCP mechanisms.

- ***Positive acknowledgment with retransmission (PAR).*** TCP uses a PAR mechanism to recover from the loss of a segment by the lower layers. The strategy with PAR is for a sending TCP to retransmit a segment at timed intervals until a positive acknowledgment is returned. TCP uses a simple

checksum to detect damaged segments which are discarded without being acknowledged. These segments are then treated the same as lost segments and are retransmitted by the sender. TCP assigns a sequence number to each eight-bit-byte (octet) section of the data stream. These enable a receiving TCP to detect duplicate and out-of-order segments.

- *Flow control.* TCP's flow control mechanism enables a receiving TCP to govern the amount of data dispatched by a sending TCP.
- *Multiplexing.* TCP employs a multiplexing mechanism to allow multiple upper layer protocols within a single host and multiple processes in a upper layer protocol to use TCP simultaneously.
- *Upper layer protocol synchronization.* When two upper layer protocols wish to communicate, they instruct their TCPs to initialize and synchronize the mechanism information on each to open the connection. However, the potentially unreliable network layer can complicate the synchronization process. Delayed or duplicate segments from previous connection attempts might be mistaken for new ones. To prevent this, TCP uses a three-way handshake procedure with clock based sequence numbers during the connection opening process to reduce the possibility of these false connections. During this handshake, the two TCPs synchronize sequence numbers by exchanging three segments, thus the name three-way handshake.
- *Upper layer protocol modes.* An upper layer protocol can open a TCP connection in one of two modes, passive or active. With a passive open, an upper layer protocol instructs its TCP to be receptive to connections with other TCPs. With an active open, an upper layer protocol instructs its TCP to actively initiate a three-way handshake to connect to another upper layer protocol. Usually, an active open is targeted to a passive open. Over an open connection, the two upper layer protocols can exchange a continuous stream of data in both directions. When data exchange is complete, the connection can be closed by either upper layer protocol to free TCP resources for other connections.

Connection closing can happen in two ways— a graceful close or an abort. A graceful close uses a three-way handshake procedure to complete data exchange and coordinate closure between the TCPs. An abort does not allow this coordination and may result in loss of unacknowledged data.

*Basic model of TCP operation.* The following provides a walk-through of a full-duplex TCP connection opening, data exchange, and connection closing that might occur between two upper layer protocols, ULP A and ULP B.

Although not depicted, the DoD protocol hierarchy network layer transfers the information between TCP A and TCP B.

- 1) ULP B issues a PASSIVE OPEN to TCP B to prepare for connection attempts from other upper layer protocols in the system.

- 2) ULP A issues an ACTIVE OPEN to open a connection to ULP B.
- 3) TCP A sends a segment to TCP B with an OPEN control flag, called a SYN, carrying the first sequence number it will send to TCP B.
- 4) TCP B responds to the SYN by sending a positive acknowledgment, or ACK, marked with the next sequence number expected from TCP A. In the same segment, TCP B sends its own SYN with the first sequence number for its data.
- 5) TCP A responds to TCP B's SYN with an ACK showing the next sequence number expected from TCP B.
- 6) TCP A now informs ULP A that a connection is open to ULP B.
- 7) Upon receiving the ACK, TCP B informs ULP B that a connection has been opened to ULP A.
- 8) ULP A passes data to TCP A for transfer across the open connection to ULP B.
- 9) TCP A packages the data in a segment marked with the current "A" sequence number.
- 10) After validating the sequence number, TCP B accepts the data and delivers it to ULP B.
- 11) TCP B acknowledges all 20 octets of data with the ACK prescribed to the sequence number of the next data octet expected.
- 12) ULP B passes data to TCP B for transfer to ULP A.
- 13) TCP B packages the data in a segment marked with a "B" sequence number.
- 14) TCP A accepts the segment and delivers the data to ULP A.
- 15) TCP A returns an ACK of the received data marked with the sequence number of the next expected segment.
- 16) Suppose, however, this segment containing TCP A's ACK is lost by the network and never arrives at TCP B.
- 17) TCP B times out waiting for the lost ACK and retransmits the last segment. TCP A receives the retransmitted segment, but discards it because the data from the original segment has already been accepted. Although the segment is discarded, TCP A re-sends the ACK.
- 18) TCP B receives the ACK.
- 19) ULP A has sent all of its data and closes its half of the connection by issuing a CLOSE to TCP A.
- 20) TCP A sends a segment marked with a CLOSE control flag, called a FIN, to inform TCP B that ULP A will send no more data.
- 21) TCP B receives the FIN and informs ULP B that ULP A is closing.
- 22) ULP B completes its data transfer and closes its half of the connection.

- 23) TCP B sends an ACK of TCP A's FIN and its own FIN to TCP A to show ULP B's closing.
- 24) TCP A receives the ACK and FIN, then responds with an ACK to TCP B.
- 25) TCP A informs ULP A that the connection is closed.
- 26) TCP B receives the ACK from TCP A and informs ULP B that the connection is closed.

This concludes our discussion of bit-, character-, and packet-oriented protocols. The advantages offered by these rules for communication transmission are too numerous to discuss here. Suffice to say that without them, we would still be providing service to our data users with only point-to-point connections.

## 626. Asynchronous modem protocols

In the early days, asynchronous communication had little use for protocols. Usually an operator would simply look at the teletypewriter to see if the circuit was in use. If it was printing, the circuit was obviously in use, so the operator would wait to send a message. When the message finished printing, the receive operator would look it over and acknowledge it, usually by sending the sequence number back to the sender. Of course, if the message was garbled, the receive operator would request a retransmission of part or all of the message. Quite often, the sending operator would automatically retransmit important numbers such as dollar amounts or addresses to ensure accuracy.

Although data communications have become quite complex and automated, most asynchronous protocols remain simple. The parity bits and start/stop bits surrounding the characters are usually the only protocol used. Although ASCII codes define special control characters, they are not used unless a more complex protocol is used. Today's asynchronous transmission still remains character oriented.

In the 1960s and 1970s, synchronous transmission grew faster than asynchronous transmission because terminals became more intelligent and communications between terminals and computers grew in popularity. However, personal computers (PCs) created a renewed growth in asynchronous. PC users found that asynchronous was less expensive; yet, it was quite adequate for PC use.

With this growth, more complex PC protocols were needed. The more elaborate protocols allowed data to be transmitted in blocks; more features were created to perform additional checks on the blocks; this improved accuracy of transmission. Today, there are a significant number of the more elaborate PC or *modem* protocols.

**TTY emulation.** Teletype (TTY) emulation is one of the oldest and simplest protocols. This same protocol is known as TTY compatible, XON/XOFF, ASCII, and Async. There are many protocols of this type that have the same capabilities, but a different name. All have the same features and are used for keyboard/printer

terminals such as Model 33. Model 33 is used between terminals and nearly all mini mainframe computers.

TTY emulation has the following features:

- a. Uses an asynchronous ASCII code.
- b. Can operate with odd, even, or no parity.
- c. Operates at either half or full duplex (one way at a time or can send and receive at the same time).
- d. Has block parity option.
- e. Sends all data and text as one block.
- f. All data begins with an SOH (start of header) or STX (start of text) character.
- g. Data ends with EOT (end of transmission) or ETX (end of text).
- h. The receiver can either send an XOFF-transmitter pause, or XON-transmitter resume.
- i. Ten or more zeros in succession indicate a BREAK. If this occurs the transmitter stops transmitting.

**XMODEM.** XMODEM, or one of its many variations, is the most widely used asynchronous protocol today. Its primary use is for the transfer of data files between microprocessors, but it has many uses. The protocol designates one microprocessor as the sender and the other end as the receiver. If the receiver is ready to receive, it sends an ASCII NAK character every .10 seconds. When the sender sees the NAK, it starts sending blocks of 128 data characters surrounded by a header and trailer. The header is the SOH character mentioned earlier and the trailer is a checksum.

When the message is received, the protocol not only verifies the checksum, but ensures that the first character is an SOH, that 128 characters are received, and that the block number is 1 more than the last block received. Once this is verified, the receiver sends an ACK back to the sender. If any error occurs, a NAK is sent and the sender retransmits the block. An ETX character occurs at the end of the message and, if all went well, the receiver sends an ACK to complete the transmission. XMODEM operates half duplex so data cannot be sent both ways at the same time.

XMODEM is not 100 percent compatible with other versions. For instance, QMODEM is designed for packet networks. If used for satellite circuits, it degrades efficiency 25 percent to 50 percent due to circuit delay. Other variations are YMODEM, which sends data blocks of 1024 characters, and UMODEM, which is for UNIX computers.

**KERMIT.** Another asynchronous PC protocol is known as KERMIT. It was first developed by Columbia University and can be better or worse than XMODEM in

efficiency, depending on which version of KERMIT is used. Unfortunately, there are about 150 different versions of KERMIT, so compatibility problems still exist.

This is not the only problem with KERMIT. There is an extended packet form of KERMIT available; however, it is not compatible with KERMIT either. Here are some of the features of KERMIT:

- a. Asynchronous half-duplex operation.
- b. Uses a variation of checksum.
- c. Works with any code.

**BLAST.** BLAST stands for *B*Locked *A*Synchronous *T*ransmission and is available on a wide range of PCs, microcomputers, and operating systems. There are two major advantages to this protocol. For one, all versions are compatible with each other. This is an obvious advantage over other protocols. The second advantage is its efficiency on satellite circuits. It is much more efficient than XMODEM when used for this purpose. BLAST is a full-duplex asynchronous protocol, can work with any code, and uses CRC for error control.

A more recent protocol with similar capabilities to BLAST is *Relay Gold*. It has the same features as BLAST, but also has the capability of setting up two channels between PCs. This allows one channel to be used as an orderwire while the other channel passes data.

Only a few modem protocols can be mentioned here; however, we have discussed some of the most popular ones on the market today. Others are available which may suit your particular needs better; for now, you should be able to analyze them and make a good choice.

---

### Self-Test Questions

After you complete these questions, you may check your answers at the end of the unit.

#### 624. Error control techniques

1. What is error control?
2. List four error control methods employed by protocols.
3. What does "odd parity" mean?

4. How effective is VRC?
5. Which two error control techniques are usually done together?
6. Which error control method is employed by the PC to PC protocol and XMODEM?
7. When we use checksum and CRC, what do we use as the error control code?
8. What error control technique is most effective for data blocks greater than 512 characters?

### 625. Protocol types

1. What are the three general categories of protocols?
2. With BSC and other line protocols of this type, what is the term commonly used to describe the interaction between stations?
3. What are the three elements of a BSC transmission?
4. What control characters precede the text?
5. What is a BCC?
6. When we use data transparency, what is the use of the DLE character?

7. Within the character-oriented protocol, what response prompts the sender to retransmit a previous block of data?
8. What are the two most widely used bit-oriented protocols?
9. What is a flag field? What is it used for?
10. What character is recognized as a flag?
11. What information is in the control field?
12. Compare the bit configuration of the opening flag and the closing flag.
13. How does the internet protocol process a datagram through a subnetwork whose maximum packet size is smaller than that of the datagram?
14. What services does the internet protocol provide to upper layer protocols?
15. What aspect of the internet protocol provides the ability for upper layer protocols to customize transmission services?
16. What services does the internet protocol require from lower layer protocols?
17. What generated the separation of the internet and transfer control protocols?
18. List the services provided by the transfer control protocol.

19. Under the transfer control protocol, what connection closing method can cause unacknowledged data to be lost?

### 626. Asynchronous modem protocols

1. What caused a renewed growth in asynchronous protocols?
2. What type of parity does TTY emulation use?
3. What happens if TTY emulation protocol receives 10 or more zeros in succession?
4. How many characters are in a block when we use XMODEM?
5. What signal does a receiver send to the transmitter if an error occurs?
6. What are the two advantages of BLAST?
7. Which modem protocol is capable of setting up two channels between PCs?

### 3-2. Standard Systems Architectures

The two main architectures in use today are open system interconnect (OSI) and standard network architecture (SNA). Even though OSI is trying to become the international standard, SNA was created by IBM, the largest computer company in the world. Both have similarities, but are not completely compatible with each other. In this section, we discuss both of the architectures.

Some material used in lessons 627-628 was taken from *The Handbook of Data Communications and Computer Networks* with permission from Petrocelli Books, Inc.

## 627. Open System Interconnect (OSI)

As was mentioned earlier, protocols allow information transfer through the use of a common set of rules or standards. One of the most common protocol standards in operation today was developed by the International Standards Organization (ISO). ISO's main purpose in life is to standardize. ISO has developed a common code set, or protocol, which can be utilized by a network. In turn, the protocol allows heterogeneous subsystems to transfer information back and forth. The protocol developed by ISO was based on a model called the Open Systems Interconnect (OSI). Remember, ISO developed OSI. The model developed used the concept of layering. Seven layers were designed to accomplish different functions. All seven layers form the entire protocol system, which is generally referred to as the OSI suite of protocols.

**The OSI model.** The OSI model is broken or layered into seven distinct categories or layers. From top to bottom, they are listed below:

- (7) Application.
- (6) Presentation.
- (5) Session.
- (4) Transport.
- (3) Network.
- (2) Data link.
- (1) Physical.

Each layer has specific functions and areas of control. Let's take a close look at each.

**The physical layer.** The purpose of the physical layer is to provide a mechanical, electrical, functional, and procedural means to activate, maintain, and deactivate an actual physical connection. Physical layer components such as terminals, modems, and multiplexers have to be interconnected by means of a physical medium. Take the phone system for instance. The telephone jack that plugs into the wall is a mechanical standard the telephone industry has accepted. The electrical supervisory signals traversing that cable are industry standards as well. The same holds true for computer communication; a standard physical medium must exist in order to interconnect physical layer entities.

The physical layer is concerned with how bits of information are actually transmitted over the physical connection by establishing (1) data transmission techniques, (2) the data channel characteristics, and (3) sharing of the physical medium. Data transmission techniques of concern are asynchronous or synchronous timing, and the various types of data encoding employed (AM, FM, PM). The data channel characteristics of concern are the types of channels available (half-duplex, full-duplex, analog, digital, or without conditioning). The physical channel is a costly resource due to the nature of the hardware involved; therefore, sharing the physical medium must be employed. The three sharing

techniques employed at the physical layer are frequency-division multiplexing, time division multiplexing, and statistical time-division multiplexing.

**The data link layer.** The data link layer provides the grammar by which machines converse with one another and serves as a vehicle for information transmission. It defines the initialization and finalization procedures, decides who talks and who listens, and assures error-free data transmission.

Our person-to-person communication has features similar to the data link protocol. The protocol involved in calling someone in order to pass on information is a good example. There is an initiation procedure which gets your friend's attention and assures the proper person has been called. Information is then ready to be transferred as is shown in figure 3-7. During the information transfer, as well as during the entire conversation, a talker and listener are designated. If this designation does not take place and both parties talk at the same time, communications break down. To assure the integrity of the information transfer, error control is implemented. Finally, there is a polite and orderly way to end the conversation.

HUSBAND	WIFE	FUNCTION
	(THE TELEPHONE RINGS)	
	HELLO?	
HELLO DEAR		INITIALIZATION
	OH, IT'S YOU	
SORRY, BUT I WON'T BE HOME UNTIL 6 O'CLOCK		TALKER AND LISTENER DESIGNATION
CAN DINNER WAIT?		
	NO! DINNER CAN'T WAIT UNTIL 8 O'CLOCK!	
NO! I SAID 6 O'CLOCK		ERROR CONTROL
	WELL, I GUESS SO. PICK UP SOME KITTY LITTER ON THE WAY HOME. OK?	
OK, BYE		
	BYE	FINALIZATION

Figure 3-7. An example of a data link protocol.

This example has explained some of the more important features of the layer 2 protocol.

- Initialization and finalization procedures.
- Talker and listener designation.
- Error control to assure the integrity of the information transfer.

**Data link modes of operation.** The layer 2 protocols perform the above functions, but they also operate in three different modes:

- (1) Synchronous unbalanced mode.
- (2) Asynchronous unbalanced mode.
- (3) Asynchronous balanced mode.

The synchronous unbalanced mode is also known as "normal response mode" and is commonly used in multi-point environments. Figure 3-8 illustrates this mode of operation. The term "synchronous" implies that the primary (host) must poll the secondaries (terminal) in order for any information to be transmitted by the secondaries. The term "unbalanced" implies that there is a master/slave relationship between the primary and the secondary; the primary being the master and the secondaries being the slaves.

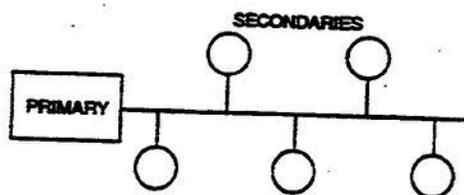


Figure 3-8. Synchronous unbalanced mode.

The asynchronous unbalanced mode (fig. 3-9) is normally referred to as the "asynchronous response mode." The term "asynchronous" implies that polling is not required; the secondary may transmit information to the primary without having been polled or asked by the primary. This mode of operation is generally used when there is a single terminal or when only one terminal is active on a host at any particular time.

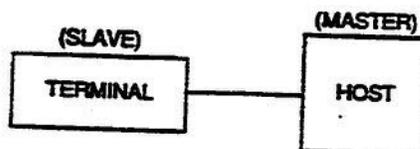


Figure 3-9. Asynchronous unbalanced mode.

The asynchronous balanced mode (fig. 3-10) is used for data communication between two host computers. The term "balanced" implies that there is no master/slave relationship; the computers are logically equal. This mode of operation is never used in the multi-point environment.

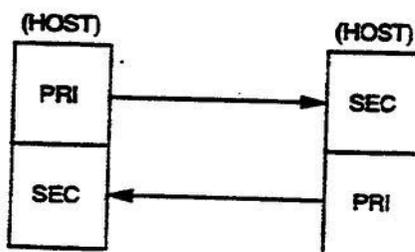


Figure 3-10. Asynchronous balanced mode.

Now that you have a firm understanding of protocols, it is easier for you to understand how they are organized into architectural layers. If protocols keep the flow of data in order, network architectures are what keep protocols in order. After all, if order is not kept, something may go wrong...may go wrong...may go wrong....

**The network layer.** The network layer protocol controls the operation of the entire network by providing an access (routing) path through a network by which the end users can communicate. The network layer protocol is concerned with describing the establishment and control of a communication path and with the routing of packets over multiple data link segments which comprise the network. This protocol is also concerned with flow control and addressing. The data link and physical layers are used by the network layer to control each link segment of a network.

The user in a computer network usually has dedicated access to only the first node (switch) of the network. Every host thinks that it is talking directly with every other host through some kind of dedicated communications channel. However, the host is really communicating with its dedicated node, which in turn, communicates with other nodes and, eventually, with the destination host.

The network layer protocol can provide two different types of services for delivering a packet across the network. These services are *virtual circuit* and *datagram* service. In the virtual circuit service, the network layer provides the transport layer with an assured and sequential delivery of all packets sent through the subnetwork. In the datagram service, the network layer does not provide an assured and sequential delivery of all packets sent.

Virtual circuit service can be compared to the telephone network. When using the telephone network, a full connection must be established first before any information can be sent from one point to another point. After all information is sent, the connections must be taken down. In virtual circuit service, a full connection must be established through the network before any packets can be delivered. After the route is established, all packets for that session must be sent through the network using the established route. The key in virtual circuit service is that all packets sent by a user for a given session travel the same route through the network.

Datagram, on the other hand, is like the postal service. When letters are sent through the postal system, they must carry a complete destination address for delivery. If letters having the same destination address are mailed at the same time, there is a possibility that not all the letters travel the same route to this destination. Letters may be lost by the carriers, etc. In datagram service, packets do not travel the same path or route through the network.

**The transport layer.** The transport layer, or layer 4, selects the route a message takes between two data terminals. Like the mail system, you are not sure how a letter gets to its destination as long as it gets there reliably. The transport layer, as

the name implies, determines how the message is transported and assures the end-to-end flow of complete messages.

Layer 4 also prevents messages from being lost or, in some cases, duplicated. If a message does get lost, layer 4 simply requests a retransmission. Since some computers run faster than others, layer 4 also prevents a fast computer from overwhelming a slow terminal. Flow control is also governed by layer 2, but only frames are controlled there. Layer 4 controls entire messages. In addition to the layer 4 functions mentioned, it can also multiplex several streams of messages from higher layers into one circuit and add appropriate headers to messages on broadcast circuits.

**The session layer.** The session layer controls the connection between machines or programs to exchange data. A session is only a temporary connection dictated by rules. During the sign-on or logon process, a user requests the establishment of a session between a terminal and a computer, but before the session can begin, such things as who transmits first and how long the transmitter will transmit must be agreed upon.

After layer 5 establishes a session, it maintains the session until the user requests that it be broken. This is usually done with the BREAK key or ESC key. If a session somehow gets broken unintentionally, layer 5 re-establishes the connection.

This layer has other important functions such as implementation of priorities to ensure timely delivery of important data. For commercial company billing purposes, it also keeps track of how long a user is on a network.

**The presentation layer.** Again, as the name implies, layer 6 deals with the way data is *presented* at the distant end of a connection. Suppose your screen format is different from the receiver's screen format. The presentation layer matches the message to the number of characters per line and the number of lines per screen.

Layer 6 also provides data compacting, code conversions, and even data encryption. Data encryption is done by hardware, but the rest is nearly always done by software.

**The application layer.** Layer 7 deals directly with the software application programs that interact throughout the network. Basically, it is the application program and user who are doing the communicating. Here are some of the elements of application that are being standardized in the application layer:

**Virtual terminal device (VTD)** — The definition of virtual terminal concepts such as graphics, hardcopy devices, and image terminals.

**Common application service elements (CASE)** — Standards for logon and password identification, plus restart, checkpoint, and backup procedures.

**Message oriented interchange systems (MOIIS)** — Standards for interconnecting message exchange systems.

*Job transfer and manipulation (JTM)* — Standards for transferring batch jobs from one computer to another.

*Office document architecture/office document interchange facility (ODA/ODIF)* — This allows users to transfer, edit, and return documents on systems with many users.

*File transfer, access, and management (FTAM)* — This allows access to files on another computer.

## 628. Standard Network Architecture (SNA)

As stated earlier, SNA is similar to the OSI model, but is not directly compatible with it. It has seven layers as OSI does, but some of the layers have different functions. In this lesson, we first look at how the SNA network is structured; then, as we describe each layer, we make note of the similarities and differences.

**SNA network structure.** SNA contains specifications both for the devices, or nodes, in a network and for paths between those nodes. Both nodes and paths are organized in hierarchies of several levels. The hierarchy of nodes is arranged to facilitate the distribution of functions under a central control. The hierarchy of paths is designed to provide flexibility and redundancy in routing.

**Network addressable units (NAU).** Each device in an SNA network, from an individual display terminal to a mainframe computer, controls a specific part of the network at its level of the hierarchy and operates under the control of a device at the next level. In describing both the structure and the process of an SNA network, participating devices and programs are represented as network addressable units. An NAU uniquely defines and represents a device (terminal or control unit), a program (application program in the host or cluster controller), or a portion of an SNA access method. In an actual implementation, a network addressable unit is a segment of a program code that represents a specific device or program to the network.

There are three general types of NAU: the system services control point (SSCP) and the physical unit (PU) are important in a discussion of the network's structure; the logical unit (LU) is important in a discussion of the active communications process.

**System services control point (SSCP).** The SSCP is the portion of the access method or control program that contains the network's address tables, name-to-address translation tables, routing tables, and the instructions that deal with those tables. The SSCP establishes connections between nodes in the network that wish to communicate and selects routes for communications between those nodes; it also controls the flow of information to ensure that the network operates efficiently. In other words, the SSCP controls the network.

Some SNA networks have more than one SSCP. In a network with a single mainframe host computer, the computer may have two access methods operating

in different partitions and controlling separate networks devoted to different applications. In networks with more than one host mainframe, each host may have one or more access methods controlling parts of its network. In such cases, the SSCPs associated with the various access methods interact as peers; each controls a domain in the network. A domain consists of one SSCP per access method plus the physical units and logical units that SSCP recognizes.

A major exception to the rule of peer-to-peer communications among SSCPs is the role of the gateway SSCP in the new SNA network interconnection feature. With that feature, a number of otherwise independent SNA networks communicate with one another through a single gateway. An SSCP in the gateway controls all communications between networks and, in effect, acts as a master SSCP for the internetwork. SSCPs in the participating networks have full control of communication within their respective domains and interact with one another as peers within their respective networks.

*Physical unit (PU).* The PU represents a single device to the network. In programmable devices, such as host computers and communications processors, the PU is usually implemented in software. In less intelligent devices, such as cluster controllers or terminals, the PU is usually implemented in microcode or in firmware. Notice that despite its name, the physical unit is not actually a physical device, but a portion of a control program that defines a collection of services. These are services that the node performs for itself and for any lesser intelligent devices that are attached to it. Each participating device in an SNA network has one physical unit.

In an SNA network, each PU generally operates under the control of the SSCP and serves as an entry point between the network and one or more logical units (LUs). An exception to this strict SSCP control is that some nodes can establish direct, peer-to-peer communications by implementing their own session management, which may support single or parallel, half-duplex sessions over multiple data links. The ability to do so without any mainframe intervention is a significant departure from original SNA strategy and is the key to the efficient integration of personal computers and other intelligent devices into future SNA networks. In order to achieve this new functionality, these nodes need to be supported by a special type of logical unit.

*Logical unit (LU).* The LU represents an end user to the network. Such an end user may be an operator at a terminal or an application program. The application may be a data entry task running at a terminal, a data base update running in a host, or any process that serves as an end point to an SNA communication. The logical unit comprises those portions of the application program and the communications software that pass and translate information from the network to the application. The logical unit maintains and transmits certain information about its own status, such as whether it is able to communicate and whether it is currently communicating. The number of LUs that can reside at a given PU depends on the type and function of the PU.

**Paths used to connect SNA nodes.** SNA defines a hierarchy among paths between nodes to contain both physical and logical components. Devices in an SNA network can be connected in one of two ways: over a high-speed input/output channel or over a synchronous data link control (SDLC) link. SDLC is a data link control protocol that we discuss later. In this lesson, we only discuss SDLC links.

**SDLC links.** SDLC links connect communication controllers to one another, to terminal cluster control units, to batch terminals, and to some individual display devices. Depending on the requirements of an individual configuration, two communication controllers can be connected by some number of parallel SDLC links. Here the word "parallel" refers to the parallel paths followed by the individual links and not to a parallel communications technique. Parallel SDLC links can back one another up in the event of one link's overcrowding or failure.

**Transmission groups.** Parallel SDLC links between two SNA nodes can be arranged logically into transmission groups. A transmission group comprises one or more parallel links with the same transmission characteristics, such as data rate, delay, security, and likelihood of errors. A transmission group appears to an end user as a single link. The communications controller decides which individual links within the transmission group carry specific messages. The individual links are transparent to the end user, who sees only the transmission group. With some systems, the users can establish up to eight transmission groups between any two controllers.

**Explicit routes.** An SDLC link between a communication controller and a cluster controller or terminal is called a peripheral link. Since, in SNA, only one link at a time can connect a terminal unit to a communication controller, peripheral links cannot belong to transmission groups.

The simplest SNA networks, those controlled by a single host with a single front-end processor and no remote communication controllers, use only peripheral links to communicate with their terminals. More complex networks involving more than one host or more than one communication controller use transmission groups between the communications controllers and peripheral links between the communication controllers and the terminal.

In such a network, a transmission may be routed through more than one communication controller on its way from the source to the destination. Its path consists of the channel from the host to the communication controller, one or more transmission groups between successive communication controllers, and the peripheral link between the last communication controller and the terminal. In SNA, the position of a path, not including the peripheral link, is called an explicit route. Note that the explicit route does not specify individual links within the transmission groups. The explicit route defines the physical characteristics of a specific path between two sub-area-node end points in an SNA network; the collective name for these characteristics is class of service. An explicit route, then, is a specific path between two end points that offers a specific class of service.

In a complex SNA network, where there may be more than one possible explicit route between any two end points, the last communication controller in the path connects the destination end point over a peripheral link. The explicit route thus selected remains in effect for the duration of the session between two end points. The individual transmission groups that make up an explicit route are redundant by design; each can contain a number of physical links that can serve as backups for one another. Sometimes, however, an entire transmission group can become unavailable because of link failure or controller failure.

**Virtual routes.** SNA's specification of the paths between sub-area nodes contains one final logical element. The transmissions in a given session between two end points can have one of three priorities. A transmission's priority governs its degree of access to an explicit route. Within the bounds that SNA uses to ensure that all transmissions have fair access to a given explicit route, high-priority transmissions have better (more frequent) access to the explicit route than medium-priority transmissions; medium-priority transmissions have better access than low-priority transmissions. A session's priority, plus its explicit route, is the session's virtual route. The virtual route defines the complete logical path between the sub-area-node end points on an SNA network.

**Layer architecture.** The communications process within SNA is divided into functional layers, each of which passes data to and receives data from only the layers immediately above and below it in the architecture. A message passing between two end points must pass through all layers in the sending node and again through all layers in reverse order in the receiving node. In some architectures, a message may pass through some of the lower layers each time it encounters an intermediate node. Each layer of such an architecture deals with a message that has a specific degree of intelligence and at a specific level of abstraction.

The division of the communication process into layers allows network architects a great deal of flexibility in updating, revising, or correcting the communications process. Designers, and sometimes users, can alter the process at one layer without affecting the other layers as long as the changes do not affect the way information is passed to and from the altered layer and its adjacent layers in the architecture. This is true as long as such changes are consistent throughout the network. The layer architecture also permits the outright substitution of one set of protocols for another at the lower layers. To the processes at a given layer of such architecture, all lower layers are simply part of "the network." In some implementations of the layered architecture, processes at the upper layers are programmed to select which set of low-level protocols is used for specific communications.

In descending order, SNA's seven architectural layers are listed:

- (7) Transaction services.
- (6) Presentation services.
- (5) Data flow control services.

- (4) Transmission control services.
- (3) Path control.
- (2) Data link.
- (1) Physical.

At each end point of a transmission, a message must pass through all seven layers. At each intermediate node, such as a communication controller, a message must pass through the three lower layers twice, once on receipt by the controller and once on retransmission.

**Data link layer.** The data link layer of SNA is very similar to that of OSI, so rather than restate data link functions, let us examine a protocol for this layer. Synchronous data link control (SDLC) is one of the data link protocols defined by SNA.

SDLC, as covered earlier, is a bit-oriented, serial protocol; it represents control information as the binary values of individual bits in predefined positions. In a bit-oriented protocol, the information on the function of a given series of bits need not be transmitted as data, since such information is contained in the positions of bits in the series.

The basic unit of transmission in SDLC is the frame. An SDLC frame is a string of bits with addressing and control information at the beginning, data from the higher layers of the architecture in the middle, and error control information at the end. Software at the data link layer receives the data from the higher layers and builds the frame by appending the control information at each end. Some SDLC frames (for instance, those that signal link level errors) contain only control information with no data from the higher layers.

An SDLC frame begins and ends with a flag, an invariable string of bits used to define the frame's beginning and end and to establish synchronization in the receiving node. After the beginning flag, the frame contains the following:

- a. The address of the secondary station.
- b. A field for control information that defines the purpose of the frame.
- c. Two sequence numbers used to ensure that the frame has been received in proper order among other frames in a transmission.
- d. A field containing any higher layer data that the frame might carry.
- e. A cyclic redundancy check (CRC) field for error detection.
- f. A final flag.

In SDLC, the final flag of one frame may also serve as the initial flag of the next frame.

The SDLC frame, as interpreted by the data link layer, contains no routing information. The address in the SDLC frame is that of the message's secondary station, which might be the routing node in the middle of a multi-node path. The data link header and trailer are stripped from the frame and recreated at every

node the message visits. Like most data link protocols, SDLC governs the transmission of data on a single link.

**Path control layer.** Routing is one of the two major functions of the path control layer; the other is flow control. Routing takes place at every node on the path between two LUs on the network. At the beginning of a session, the sending and receiving nodes and all nodes in between cooperate to select the best available virtual route for that session. During the session, each node along that route selects the next available link in the selected transmission groups for each message within the session.

Every LU on a network has a network name, a mnemonic known both to the end users and to applications that might communicate with that LU. The end result of this layering of addresses is that an end user need know only the name of a terminal or process on the network to communicate with it; the end user does not have to know where that terminal or process resides in the network.

The routing function also includes assignment of a class of service to each session. The SSCP passes a list of virtual routes meeting a requested class of service to the path control layer and that layer activates the first available virtual route in the list. To improve the efficiency of transmission, the path control layer at each node along a session's path can segment messages. The segmenting function divides long messages into segments for transmission by the data link layer in separate SDLC frames. The segmenting function ensures that the efficient use of the network's transmission facilities does not depend on arbitrary message lengths assigned for the convenience of individual applications.

The final flow control function of the path control layer is virtual route pacing, which ensures that traffic along a virtual route shared by a number of sessions does not overly congest that route. To communicate routing information to the path control layers in successive nodes along the session path, the path control layer appends a transmission header at the beginning of each message.

**Transmission control services layer.** This layer is responsible for the pacing of messages for individual sessions. The chief purpose of this session level pacing function is to ensure that a transmitting NAU in session with a receiving NAU does not transmit more data than the receiving NAU can handle in a given period of time. To perform session level pacing, the two NAUs at either end of a session negotiate the size of a pacing group at the beginning of the session. A pacing group is the largest number of messages that the transmitting NAU can send before it receives a pacing response from the receiving NAU telling it that it can resume sending. Sending level pacing occurs in two stages along a session's route: between the host NAU and the communication controller and between the communication controller and a peripherally-attached terminal NAU.

The transmission control services layer also performs encryption when the using application program requests it. SNA offers two types of session level encryption. Mandatory encryption causes the encryption of all messages within a session; selective encryption encrypts only those messages identified by an enciphered data

indicator embedded in the message. To perform its pacing function and to transmit certain other flow control information from the higher layers, the transmission control services layer adds a request/response header to each message before it passes the message along to the path control layer.

**Data flow control services layer.** This layer handles the order of communications within a session by establishing chains and brackets of data and by maintaining one of three send/receive modes. A chain is a group of messages associated logically for transmission in one direction on the network; a bracket is a group of messages associated logically for two-way transmission on the network. The data flow control services layer establishes chains and brackets for two purposes: error control and contention control.

When an error occurs in one message of a chain of messages, the receiving node notifies the data flow control layer of the sending node and the sending node then holds the remaining portion of the chain. If the LU detects the error, it is a protocol violation that caused the error, and the session is terminated. If the application program detects an error, user error recovery occurs.

The three send/receive modes that the data flow control services layer establishes and maintains are full-duplex, half-duplex flip-flop, and half-duplex contention. In full-duplex mode, each participating LU can transmit at any time, whether or not the other LU is transmitting. In half-duplexing flip-flop mode, the participating LUs transmit alternately; at the end of a chain, the LU currently transmitting may pass permission to transmit to the other LU. In half-duplex contention mode, one LU is designated as dominant at the beginning of a session; either LU may begin transmitting at any time, but if the dominant LU is transmitting when the other LU attempts to transmit, the other LU must withhold its transmission until the dominant LU has finished transmitting its current chain.

**Presentation services layer.** The presentation services layer defines the protocols for program-to-program communication and controls conversation-level communication between transaction programs by doing the following:

- a. Loading and invoking transaction programs.
- b. Maintaining conversation send and receive mode protocols.
- c. Enforcing correct verb parameter usage and sequencing restrictions.
- d. Processing transaction program verbs.

**Transaction services layer.** The transaction services layer is the highest architectural layer defined by SNA; it implements service transaction programs in an SNA network. Services transaction programs provide these services for end users of an SNA network:

- a. Operator control of LU-to-LU session limits.
- b. Document interchange architecture (DIA) for document distribution between office systems.

- c. SNA distribution services (SNADS) for asynchronous data distribution between distributed applications and office systems.

The transaction services layer also provides configuration, session, and management services to control the network's operation. The configuration services activate and deactivate links, load same-domain software, and assign network addresses during dynamic reconfiguration. The session services translate network names to network addresses, verify user passwords and user access authority, and select session parameters. The management services manage network problems, manage performance and accounting information, manage the network configuration, and manage changes in the network.

SNA has been designed as a framework for an evolving architecture. No single technical breakthrough will make SNA obsolete, and SNA users will be able to incorporate new technologies without major disruptions to their networks' design, operation, and upgrade plans. If you are an SNA user, you should expect the architecture to grow with the definite development of additional services throughout.

### Self-Test Questions

After you complete these questions, you may check your answers at the end of the unit.

#### 627. Open system interconnect (OSI)

1. Who developed OSI?
2. List the three modes of data link operation.
3. Match the OSI layer in column B with its description in column A. Column B answers may be used only once.

#### Column A

- \_\_\_ (1) Controls the operation of the entire network.
- \_\_\_ (2) Deals directly with the software application programs that interact throughout the network.
- \_\_\_ (3) Provides a mechanical, electrical, functional, and procedural means to activate, maintain, and deactivate an actual physical connection.
- \_\_\_ (4) Assures the end-to-end flow of complete messages.
- \_\_\_ (5) Defines the initialization and finalization procedures, decides who talks and who listens, and assures error-free data transmission.
- \_\_\_ (6) Provides data compacting, code conversions, and even data encryption.
- \_\_\_ (7) Controls the connection between machines or programs to exchange data.

#### Column B

- a. Physical.
- b. Data link.
- c. Network.
- d. Transport.
- e. Session.
- f. Presentation.
- g. Application.

## 628. Standard Network Architecture (SNA)

1. What is the segment of the program code called that represents a specific device to the SNA network?
2. Which NAU establishes connections between nodes in an SNA network that wish to communicate with each other?
3. Which NAU is part of the control program that defines a collection of services available to the SNA network?
4. Which NAU is used to pass and translate information from the network to the application?
5. What is the main purpose for an SDLC link?
6. What unit decides which individual link within a transmission group carries a specific message?
7. What two controllers are connected together by a peripheral link?
8. Why are the individual transmission groups that make up an explicit route redundant by design?
9. Why did the network architects divide the communication process into layers?
10. What are the seven architectural layers used in the communications process from start to finish?
11. Which layer handles flow control for the communications process?

12. Which layer handles encryption for the communications process?
13. What are the three send/receive modes established and maintained in the data flow control services layer?
14. Which SNA layer has the responsibility for defining the protocols for program-to-program communications?
15. Which SNA layer provides configuration, session, and management services to control network operation?

### 3-3. Digital Interface Devices

Digital communications networks have become quite elaborate in terms of technological advances. Consequently, they require a lot of equipment to make the overall network operate. Many pieces of equipment are used for interfacing. They all fit together like a jigsaw puzzle to create a system that is acceptable for our needs. We mention some of them here briefly so you can get a better understanding of how they work and why we use them.

#### 629. Systems interface devices

Basic devices are used to connect communication channels from a distant facility to local users. This lesson deals with some of these interface devices and expands your basic understanding of them. It discusses the devices from a systems control perspective.

Some material used in this lesson was developed using information from *GTE Lenkurt Demodulator* with permission from Siemens Transmission Systems, Inc. and *The Handbook of Data Communications and Computer Networks* with permission of Petrocelli Books, Inc. Permission to use these materials is gratefully acknowledged.

**The function of a modulator/demodulator.** The modulator/demodulator is commonly referred to as a "modem." Modems have been referred to as "data sets," "line adapters," or "subsets." The goal of a modem is to permit terminal-to-computer and computer-to-computer communication over a telephone line. The primary function is the modulation/demodulation of carrier signals so that digital

information can be transmitted over an analog communications link, usually a conventional voice-grade line.

If two machines (such as computers, data terminals, or facsimile machines) are communicating over a telephone line, the computer or terminal at either end must be equipped with a modem. The modem converts binary data coming from the terminal (or computer) to analog signals suitable for transmission over a voice line (and vice versa). A pair of modems are considered "transparent" since the signals into the first (the input) are identical to the demodulated signals (the output) from the second modem.

There are various types and speeds of modems that operate differently from each other. Most often, low-speed modems are asynchronous and high-speed modems are synchronous. If start and stop bits are used to "frame" each character, the transmission is asynchronous. Synchronous modems require the use of clocking devices that lock the transmitted signal of the modem and the terminal device together at a fixed transmission rate.

High-speed data generally uses a synchronous modem for its transmission. As a result of synchronous modems having identical data coding levels and transmission bit speeds, a higher data speed can be achieved. Asynchronous transmission requires the use of 2 or 3 start and stop bits for each character, depending upon the type of machine generating the digital signal. Consequently, if an 8-bit code is being used, asynchronous transmission requires 10 or 11 bits per character; synchronous requires only 8. Synchronous modems can, therefore, transmit at least 25 percent more characters than asynchronous modems at the same bit speed.

Although synchronous transmission is efficient, the clocking mechanism requires added circuitry and, therefore, makes this equipment more costly than asynchronous modems for the same speed.

Asynchronous modems have a specified maximum transmission speed, but they can be used to transmit data at any speed up to this maximum. Asynchronous modems are used for low- and medium-speed transmission up to approximately 1,800 b/s. There are also other types of modems named after their usage. Short-haul modems handle high-speed data transmission over short distances. High-speed, special-purpose modems operating at 56 k or 64 kb/s handle computer-to-computer data transfers or transfers between special-purpose terminals and CPUs. However, 64 kb/s is not the high end of modem function. Data transfers in some instances may occur at rates as high as 1.5 Mb/s.

Another way to classify data modems is according to the type of bit stream used—parallel or serial. Figure 3-11 illustrates the difference between serial and parallel bit streams. Serial bit streams are most commonly used since the digital information can be modulated as it comes from the digital machine. As long as sufficient bandwidth is available for transmission without degradation, a serial bit stream can be used.

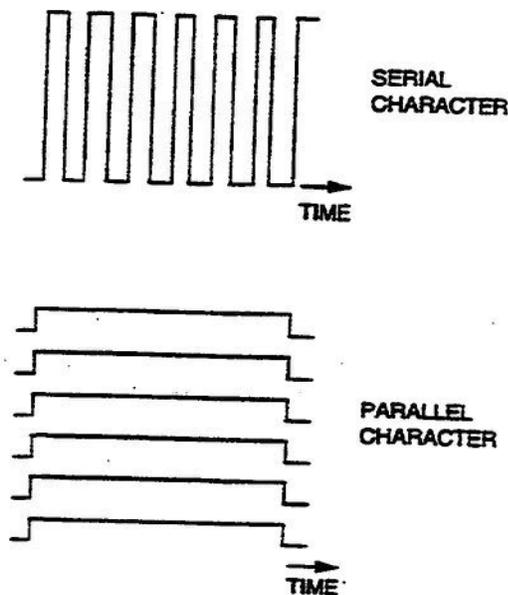


Figure 3-11. Serial vs. parallel transmission.  
(Reproduced with the permission of GTE Lenkurt Demodulator)

If, however, transmission is to take place over bandwidths that do not have uniform transmission characteristics, a serial bit stream can be converted to a parallel bit stream. At the receiver, a parallel-to-serial conversion takes place. This technique is often used to transmit data at 4,800 b/s and higher over a voice channel.

Parallel channels with their longer symbols provide better correlation of fade and phase factors and multipath delay distortion in the propagation medium (radio or cable). However, the complex circuitry for parallel transmission makes parallel modems more costly. They are also less efficient, since bandwidth is used for flanking the bandpass filters in each channel. For these reasons, serial modems have been accepted as a standard.

**How the line driver functions.** A synonymous term for line driver is "line adapter." The function of the conventional line driver or line adapter is to modify (or act as an interface for) the various electrical voltage and current levels that may exist between the communications channel and the different hardware communications devices. A line driver is a black box that takes the digital electrical signals from the DTE and amplifies and possibly reshapes the pulses so they can be transmitted farther than would be possible with RS-232-C standards. This reshaping is done by using a line driver to transform a normal data signal to a special balanced line signal. The balanced line signal can be transmitted over much greater distances than the standard interface signal. Most standard interfaces (EIA RS-232-C and MIL-STD-188) have an effective operating range of less than 100 feet. Line drivers usually can handle data speeds of up to 9,600 b/s or 19.2 kb/s, and they can transmit over distances of up to several thousand feet or even longer.

Line drivers are inexpensive compared to conventional modems, but they do require DC continuity on a pair of wires. "DC continuity" means a circuit appears to be a continuous circuit (rather than an open circuit) to a direct current. This usually means a continuous metallic circuit—a circuit made of wire that is not interrupted by devices such as amplifiers, transformers, or capacitors. Many circuits in the voice telephone network look continuous to a voice signal, but look open to a direct current because of the insertion of devices such as amplifiers or coils.

Because line drivers require DC continuity, they are usually used on circuits within a user's building or group of buildings. In some cases, however, a pair of wires with the required DC continuity can be obtained from your local dial exchange. Line drivers can usually be used in series with two or more at different points on the line so that longer distances can be covered.

**What is the purpose of the buffer?** All messages entering a telecommunications network are routed through a buffer into the main storage areas for handling, arranging, and transferring of message segments. A buffer mainly serves as an isolation device, but on some occasions, can also supply amplification with the isolation. As an isolation device, it serves to eliminate the effects of load changes on the circuit it is interfacing. A buffer and a temporary storage area make up a buffer storage unit. A buffer storage unit has two main operating areas—an area containing control information and an area containing all or part of the message. Buffer storage units must be at least 31 bytes long and may be no longer than 65,535 bytes.

During the processing in a network switching node, information can be stored in temporary memory till that processing is finished. The temporary memory can then be used by any other process. For efficient and convenient management of memory, a portion of primary memory is partitioned off as the temporary memory. This temporary memory is further divided into smaller pieces called buffer storage units. Some systems find it convenient to define all buffer storage units to be of the same size, but others prefer to define a few standard sizes with a buffer storage unit pool containing buffer storage units of each size. Buffer storage units are the basic building blocks for data before it is transmitted to another destination.

The size and number of storage units in the pool are specified by the user. For internal management purposes, 12 bytes are added to the user-specified unit size. Thus, if a user specifies a unit size of 60 bytes, the size of the unit is 72 bytes. The user does not include the extra 12 bytes when defining the buffer storage unit.

A special process called buffer management, also known as memory management, is certainly one of the most important functions of the network controller. The controller acts as a monitor to the buffer pools by managing the buffer system used to store messages during transmission to and from the network. A transmission needing a buffer sends a request to the buffer management processor, which allocates an available buffer to the requesting transmission. When finished, the requesting transmission releases the buffer so the buffer management

processor can mark it available. The buffer management process is sometimes part of an operating system.

There are a great number of different buffering strategies, each useful in a particular environment (fig. 3-12). For instance, the use of linear storage buffers is appropriate when the extent of the message is known and the buffer storage can be allocated in advance, which is often the case for output messages. The use of circular storage buffers is appropriate if several messages of undetermined length are to be stored before one of them is processed, which is often the case for input messages from typewriter terminals. Finally, a pool of chained storage buffers shared among a group of terminals is appropriate if the message sizes and arrival times vary over wide ranges and cannot be predicted in advance. Two or more storage buffers are chained together if a message exceeds the length of a single storage buffer.

In most cases, however, the length and time of arrival of a message cannot be predicted in advance. If you use linear storage buffers, allocate a storage buffer at least as long as the longest expected message before transmission can begin. Since most messages are shorter than the longest expected message, there is a

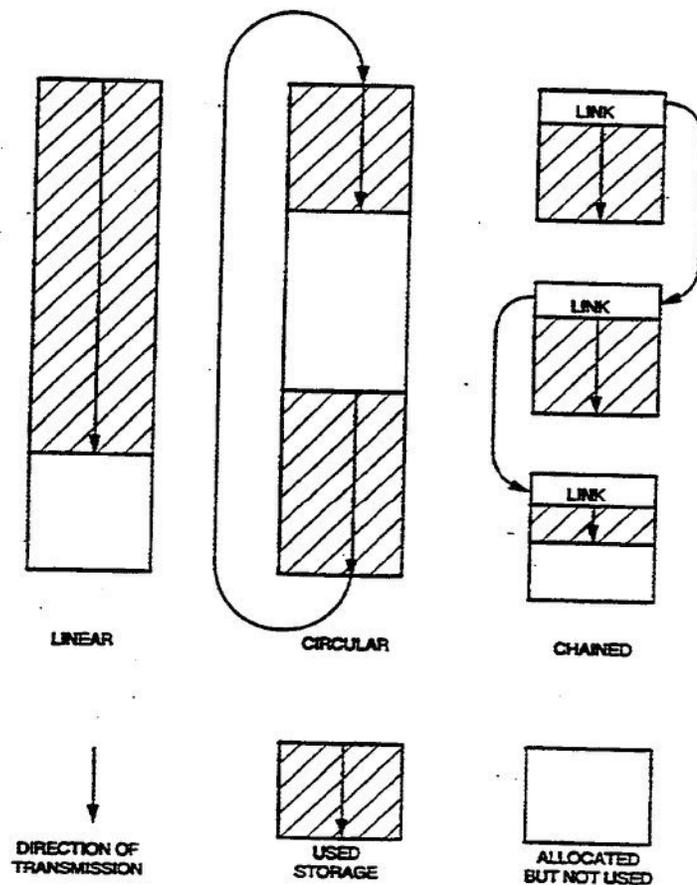


Figure 3-12. Buffer storage allocation.

considerable waste of buffer storage, especially if many terminals are involved. In such a case, a pool of short, fixed-length, chained storage buffers is appropriate. A storage buffer is removed from the pool only when storage is required as characters arrive and is returned to the pool when all characters have departed.

Although a chained storage buffer system is highly efficient in its use of storage, relative to linear and circular storage buffer systems, rather complicated input-output channel programming and critical network-control program response times are required. For these reasons, many current systems use the simpler linear and circular storage buffer systems instead.

**Purpose and characteristics of circuit switching.** One of the biggest issues in network architecture is the choice of switching technology. Let us look at the fundamentals. Circuit switching essentially provides a computer network a means of deciding how to share the communications facility. With point-to-point operations, the alternatives are multiplexing and switching. With multipoint (broadcast, multidrop) approaches, we have a choice between polling and contention. They all involve line switching. There are three basic switching disciplines available for line switching: circuit, message, and packet. We discuss only circuit switching in detail.

Circuit switching is the oldest and most widely used discipline. This classic approach provides a "total path." There is a setup time for call establishment; call termination results in the circuit being dropped. Circuit switching may have many different aspects and offer a variety of service levels.

The best example of circuit switching is a telephone network. If two users at distant locations need to talk to each other only occasionally, it is not cost effective to have a dedicated line between them. Using a circuit switching concept allows them to communicate without dedicated lines. Both users have a drop to a local switch as their only dedicated line. These short-haul dedicated lines are rather inexpensive. To connect to a distant user, the local switch uses long-haul lines to another switch that is shared with all the users serviced by that switch. These long-haul lines do not connect with any users directly, but are continually used by the users when calls are processed. By sharing the long-haul lines, the military community has serviced hundreds of users with only fifty or so long-haul (expensive) lines. By having switches located throughout the world, it is possible for very distant users to communicate effectively and cheaply. With numerous switches, it is easy to design survivability into the system by having previously planned alternate routing in cases of switch failures.

The basic characteristics of circuit switching can be outlined as follows: First, it involves the establishment of a total path at all initiations with your host switch. Because of the way in which current voice-grade networks are implemented, this results in significant inflexibility. Second, the circuit is established by a special signaling message that threads its way through different switching centers. Third, this circuit (being a total path) is subject to the speed and code limitations of the slowest link. As stated, the total path remains allocated for the transmission,

regardless of utilization. Circuit termination happens when the circuit is dropped; each line and switch is returned to a pool of available circuits.

Circuit switching has three advantages:

- (1) Circuit routing is simpler.
- (2) The discipline is the most widespread today.
- (3) There is the possibility to attach terminals with different characteristics.

There are, however, many disadvantages:

- (1) No automatic retransmission.
- (2) Switching failures fatal to the traffic present at the time of failure.
- (3) Inefficient to "bursty" traffic.
- (4) Voice-grade lines.
- (5) No record keeping.
- (6) Long setup time (with traditional circuits).
- (7) No speed- or code-matching capability.
- (8) Relatively high error probability.

Since there is no particular preoccupation with error control in circuit switching, errors filter through more often than in the other switching disciplines.

We must emphasize that many of the problems of circuit switching (connect times, bandwidth, conditioning, voice and data incompatibility) are common to current networks. With the new technologies, the entire way of handling communications traffic will change and circuit switching will be able to stage a comeback. Intelligent terminals will provide speed change, buffering, editing, etc. Circuit switching can also handle traffic without preallocating resources. With other methods, this cannot be done. These are just a few of the reasons circuit switching has gained such widespread acceptance over other network solutions.

**Concentrators vs. multiplexers.** A line concentrator is essentially a switching device that provides for connections between a large amount of subscriber lines and a small amount of talking (or interconnecting) trunks. A concentrator (or remote data concentrator) is a single-sided communications processor that concentrates data from many lines onto a smaller number of lines.

For subscribers located a considerable distance from the central office or inside plant, the cost of physical circuits for each subscriber and the time needed to install them becomes quite substantial. This is especially true when dealing with a rapid influx of new subscribers. Both concentrators and multiplexers can provide this needed new service quickly and economically.

A concentrator, quite similar at times to a multiplexer, is often referred to as "concentrator/multiplexer." Technically, concentrators differ from multiplexers in several respects:

- a. They are only needed at one end of the line, unlike multiplexers, which are needed at both.
- b. The up-line or composite line consists of a number of lines; with multiplexers, the composite line is usually a single line.
- c. They may have much more capacity than multiplexers, handling up to 3,000 or 4,000 lines.
- d. They generally deal in a single protocol, usually the native protocol of the host.

Data concentration is a logical function that combines a series of communications links into one physical line (fig. 3-13). They use a sharing and switching scheme in which some number of input channels shares a smaller number of output channels on a demand basis. This can be done with hardware (multiplexers) or with software and hardware (processors). Consequently, it is not possible to have all concentrator subscribers using their phones simultaneously. For this reason, statistics and queuing play an important role in the planning and use of concentrators in an attempt to ensure that trunks are available when needed.

Concentrators also are physical devices; they are selected according to efficiency, reconfiguration capabilities, configuration of line interfaces, and diagnostics. Consider traffic loading when you plan to use a concentrator for service to remote

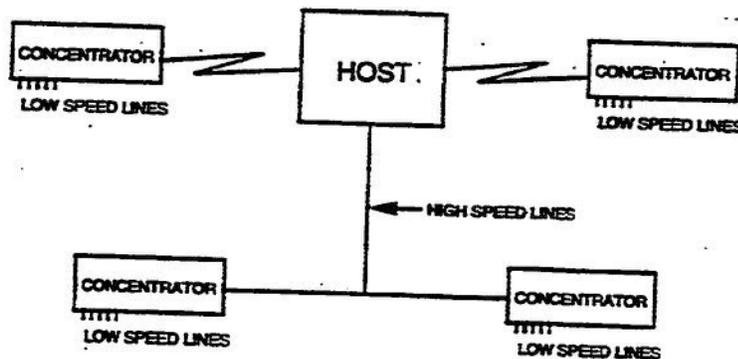


Figure 3-13. Concentrator. (Reproduced with the permission of Petrocelli Books, Inc.)

locations, along with the grade of service it requires. Grade of service that the commercial facilities normally refer to are "P(.01)," "P(.02)," "P(.03)," etc. The term "P(.01) grade of service" refers to the probability of access (or availability) that 1 call out of 100 may be blocked due to lack of a free trunk; thus the user would in turn receive a busy indication. Similarly, a P(.02) would mean there is a probability of 2 calls out of 100 being blocked due to lack of a free trunk. Therefore, you can see why the military leases the P(.01) service and also maintains its own equipment to this standard.

With a multiplexer, the grade of service of the central office is not affected and is only a function of the traffic loading; while with a concentrator, the grade of service is a function of the central office traffic loading and the traffic loading at the remote concentrator.

Both multiplexers and concentrators regroup low-speed lines into a high-speed corresponding to an equal number of terminals that are multiplexed into  $M$ -line channels. Normally, all lines coming from the terminals to the multiplexer are of the same speed. At the output side, the line uses frequency-division or time-division. The messages are of fixed format. The  $N$  lines feed into the concentrator, but the output is one channel. The lines connecting the  $N$  terminals to the concentrator may have various (mixed) speeds. The high-speed line at the output side carries formatted messages and blocks of messages. This is shown graphically in figure 3-14. The concentrator has to be an intelligent machine; it requires software. Multiplexers have long been unintelligent equipment, but this is changing.

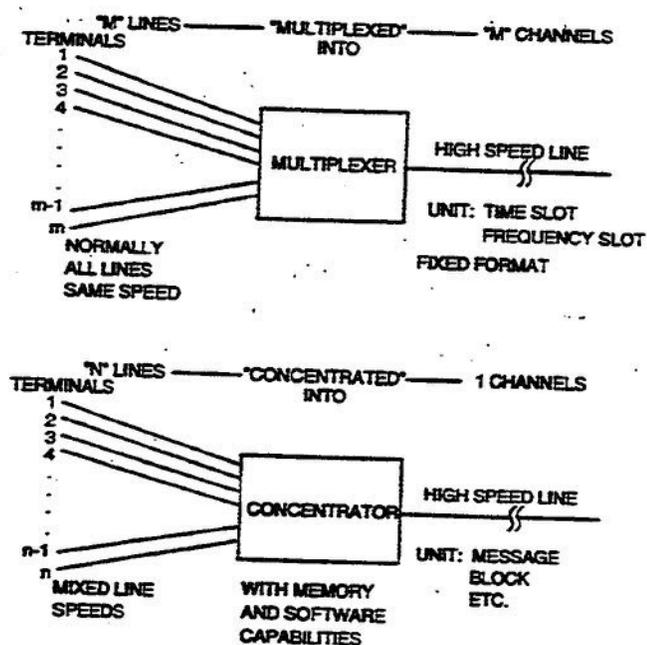


Figure 3-14. Comparison of multiplexer to concentrator.  
(Reproduced with the permission of Petrocelli Books, Inc.)

### 630. Access interface devices

Data transmission and reception requires more than the system interface devices previously discussed to service network and internetwork communication. For message traffic to be successfully processed, it must navigate through various network connections and architectural protocol layers. This is accomplished through the use of access interface devices designed to handle such tasks as message handling and routing, protocol translation, error checking, security, retransmission, and user services. Some of these access interface devices are briefly covered here and many of them are discussed in more detail later.

Portions of this lesson were developed using information from *Introduction to Local Area Networks-second edition*, ©1986, provided by The Information Factory. Permission to use these materials is gratefully acknowledged.

**Front-end processors.** Front-end processors, or front-end network processors (FNPs), are associated with host mainframe computers and perform a number of communications-related functions. They relieve large central computers of the extra effort involved in message handling and network control. This decentralized approach permits communication processors and the central processor to perform their primary functions in parallel with little interference. Data passes between the processors only when necessary and with a high degree of efficiency. The front-end processor ideally assumes all terminal, line, buffering, message handling and control functions, permitting the central computer to concentrate on processing information. The FNPs advantage is that it accomplishes tasks that the host computer would otherwise have to do. This allows the host computer system to concentrate on processing batch programs and producing useful output. FNPs routinely perform such duties as serial-parallel conversion, packetizing, multiplexing and concentration, network access signaling and supervision, protocol conversion, error control, and diagnostics.

**Hubbing devices.** These interface devices multiply digital input signals so as to make several outputs available for transmission to a number of different terminal stations. They are often used on news, weather, and party-line, order-wire circuits. Older versions were usually made up from a number of relays. The newest versions are all electronic, full-duplex, solid-state units. In operating principle, they are like the four- or six-way bridges used on analog voice circuits.

There are various models of data hub repeaters available. Generally speaking, they are up-to-date, digitized, solid-state devices, consisting of a series of plug-in printed circuit boards and an enclosure with access terminals and loop-condition indicator lamps.

We describe one model used in an Air Force satellite data collection system. The hub is an equipment unit of the ground communications network associated with this system. It is known as the Model 820 data hub repeater.

This 150-band repeater facility provides all functions required for a four-wire, half-duplex-operated, eight-circuit data hub. Each circuit leg has an input and an output port. Signals inserted into any input port are repeated at all output ports, including the output port associated with the sending circuit.

Incorporated within the repeater is an *open-loop annuller*. The annuller prevents complete interruption of the hubbing facility in the event of trouble on one of the circuit legs (open loop) or in case of failure of a portion of the repeater circuitry. After a momentary interruption (1 to 5 seconds), all circuits except the faulty one function normally. In addition, the annuller display panel gives immediate identification of the failed circuit.

Hubbing devices find application throughout the Air Force, especially with weather circuits that carry data all over the world from a central location. Another application is the critical-control, order-wire network used to coordinate with the Defense Information Systems Agency (DISA).

**Network interface unit (NIU).** The interface between the host computer and a network can occur in three different ways, depending on hardware considerations. First, the host computer can be an integral part of the communications network, possibly performing a message-switching function for the network (messages are directed to it or other nodes on the network) and thus requiring no special interface computer. Second, the host computer can be connected to the network through the message-switching computer, again requiring no additional hardware other than the transmission medium. Third, the computer can be connected to a front-end processing computer that acts as an interface to the message-switching computer and, therefore, to the network—intercepting and controlling all input and output between the network and the host computer. This last method is the one we talk about since it requires the network interface unit.

The network interface unit, also called an adapter card, is one of the most expensive parts of the local area networks. The network interface unit is located between the host computer and the network. Within the network, the NIU acts as a node and becomes the point where all the terminals link with the network. This unit can interface one terminal or many terminals to the network. Each input to the NIU is called a port, and there are usually 2 to 16 ports per unit. The unit's main purpose is to serve as a buffer storage for the terminals. Since it is the interface point between the network and the terminals, it also acts as a message router. All the messages on the network come to the NIU. It sends them to the appropriate terminal addressed on the message.

The NIU is a plug-in adapter made of printed-circuit boards. It plugs directly into the bus socket of a printed-circuit motherboard. Access to the medium takes place directly from a connector at the back of the card. Data transfers directly from the communication controller on the card into the printed circuit's memory. This is direct-memory access (DMA). Transfer of data is very fast, as it happens at bus speed (10 Mb/s). The NIU plays an important part in the overall network, but we have to keep in mind that whether or not a network requires this unit is based totally on how the network in question is designed.

**Gateways.** A gateway is the connecting link between two separate networks. The present data services are based upon providing a physical data channel with a dedicated bandwidth. This can be permanently leased as a private line or periodically accessed as a switched offering, but the crucial issue is the provision of gateways that permit different line disciplines to be attached to the network.

The gateway may be as simple as a telephone circuit linking the two networks, or it may include complex hardware and software to allow messages to move between two otherwise incompatible networks. Among the advantages of gateways are uniform access to a variety of resources, uniform access to

specialized resources, and the lower cost resulting from having a single access port to other local networks. One disadvantage of the gateway approach may be protocol incompatibility because of possible difficulties in finding matching equivalent sets of services at all levels of the standard protocol model.

Interconnection of local networks may become more complicated if the individual networks operate at different speeds or if they have different levels of security. If the networks are incompatible, the gateway may supply code, protocol, and speed conversion among other functions. Gateways may also perform electrical, character, or user-to-user translations; perform complex routing; or even initiate communication onto one of the attached networks. These characteristics are generally attributes of gateways:

- Message-passing capability between connected local networks.
- Access-control mechanisms.
- Segmentation/collection capabilities.
- Congestion and flow-control capabilities.
- Accounting mechanisms.
- Intergateway retransmission capability.

In general, the users of data communications services must call for the network to provide a wide variety of services from front ends to mainframes. This is seconded by internal gateway capabilities (functional entities) and by the other services the network must offer. Stated briefly, a gateway is a path offered between two information-processing systems with dissimilar protocols. It provides the necessary transformation from one protocol to the other and thus allows the exchange of information.

In this regard, two kinds of gateways can be identified—media-conversion and protocol-translation gateways. Media gateways receive messages from one network, perhaps remove the envelope, generate a new envelope and routing information, and forward the message to another network. The name is obtained from the fact that the media channels used by the different networks may be different. A media gateway is at the core of the internet service.

Protocol-translation gateways translate a protocol used in one network to that used in another, generally for two networks with protocols of similar code sets. It does so by mapping messages from one protocol to those of another, a task that increases in difficulty as we rise in the protocol hierarchy and simple one-for-one mappings become unlikely. One approach to this mapping is to provide gateways that map from a network-dependent protocol to a standard internetwork protocol, and the reverse.

**Network bridges.** If your network architecture is laid out into various segments of the same type, a bridge helps you to connect between the segments. This is used for homogeneous (logically similar) segments and only passes data to the

segment if that segment needs it. The bridge blocks or filters information not required for a particular segment.

A multidrop network is useful where the central processing unit (CPU) communicates with many modems scattered throughout a large geographical area. This is made possible by the use of network bridges. This approach minimizes line costs while providing each remote operator with a reasonable approximation of an on-line interactive environment. The central host site is connected in a linear fashion to many remote terminals so that the data path from the central site goes first to the remote terminals and then serially to the subsequent locations. The system operates under control from the central site, which signals or polls each remote site sequentially to receive and/or transmit information. A typical response on a polled network ranges from 0.1 to 4 seconds after a request.

Multidrop networks are available in two-wire and four-wire configurations. Two-wire networks require a two-wire bridge configuration at each drop point. This bridge must generate gain in two directions because a two-wire system uses the same pair of wires for both the transmitted and the received signal at the modem. Four-wire systems have a dedicated pair of wires for transmission in each direction.

Bridge configuration in two-wire networks is useful when the number of drops or terminals is five or less. In larger networks, line impedance problems occur, oscillations arise on the lines, and the system ceases to function effectively. As a result, four-wire systems are more common.

**Transceivers.** A transceiver is a network terminal connection device. Commonly used in baseband networks, the transceiver combines transmitter and receiver operations to connect a terminal to the baseband coaxial main cable. It is physically plugged into a hole drilled into the coaxial cable and may be remotod up to 50 meters from the terminal itself. In some applications, the transceiver provides services such as signal conditioning, impedance matching, and reaction to network traffic collisions.

**Media access unit (MAU).** A media access unit is a device that provides the signal and protocol conversion between a network terminal and the backbone of the network. It provides the proper interface between the terminal and the transmission media, usually a baseband coaxial cable system. The MAU may sometimes appear as part of a transceiver device. Some of the features may include automatic collision detection, signal loss detection, retransmission capability, and fault isolation.

**Repeaters.** The physical media employed by your network has limits in the distance it can hold a signal. A repeater is used to regenerate and retransmit the signal to extend beyond the physical limits of the media. Repeaters are among the least expensive and most common equipment in a digital transmission network. They're subjected to the manifold slings and arrows of the environment, and they're vulnerable to rough handling between manufacturer tune-up and network

installation. They receive an attenuated signal, recreate the original signal, and transmit it to the next repeater or piece of transmission equipment.

**Routers.** Routers are similar in function and have features similar to bridges. Ordinarily, on a single local area network (LAN), when a message is transmitted, it is sent to all nodes in that network. Each node must then determine from the message's destination address if it should be received and whether or not to process the message. However, when a LAN is interconnected with other networks, message routing becomes a critical issue. Normally, the function of routing messages is assigned to an intermediate node on a network that determines if the incoming messages should be sent to its network and if so, to which node they should be addressed.

Routers can develop a map of the network by exchanging information on which nodes and links within the network are active and then selecting a route for sending the messages to the appropriate nodes based on the current network map. In order for routers to be used, they must share the same network protocols and must be compatible with higher network layers. Routers offer more efficient traffic control than bridges, but are typically a bit more expensive and much more difficult to install.

**Server/processor.** At the heart of a local area network is the server. It is usually a high end computer that has a large amount of RAM and hard disk storage space. The server contains most of the programs that control the LAN. It contains the software programs needed to manage the network and does for the LAN what the operating systems do for a personal computer. It may also contain large application programs that can be called into use on an as-needed basis by the network users, instead of each of them having the application individually installed. The server contains the electronic mail programs and can also be configured to share resources, such as printers, so that all the network users have access to them.

**Disk server.** This is a common disk storage system connected to the LAN. Usually these devices are partitioned with each user having a particular storage area reserved. In some systems, the users can read or write to other users' files. Files may also be moved between users through the disk server.

**File server.** A file server is usually an intelligent disk server that can provide storage and data conversion. When accessing the file server, users generally log into a security system that determines their access level. Using the file server, files and data can be moved between users or systems and translated to match the operating systems in the various users' microcomputers. The file server can also provide electronic mail and other services.

**Application server.** Application software programs are installed in an application server to be made available to network users on an as-needed basis. This allows for the sharing of these programs and prevents each user from having to install them on their individual systems. When a user calls a program from the application server, only the program files needed to perform a specific task are

actually affected. These files are copied from the application server into the user's assigned file server storage space and remain there until no longer needed. Once the user closes the application, the program files are removed from the user's storage.

**Communication server.** Communication servers are designed to connect terminals, printers, or other non-intelligent devices into LANs. The server may be thought of as a microcomputer with floppy or hard disk storage.

Each device connected to the network must be able to communicate with required electrical and protocol standards. If a user wants to connect to the network using a "dumb" terminal, then some level of intelligence such as a communication server must be placed in front of the network connection. In this instance, the server provides services such as echoing characters, buffering data, assembling packets, establishing network channels, and matching electrical and protocol interfaces.

**Print server.** A print server connects printers to a network for resource sharing among users. They usually include a memory buffer (print spooler) that accepts files, stores them in a print queue, and prints them on a first-in/first-out or priority basis whenever the printer is available. Some software packages allow users to select, or capture, specified network addressed remote printers. In this way, the user can select the location for the print job to be accomplished. Generally, all printers in a network have an identifiable address which allows print-requested data to be directed to them.

This ends our discussion on digital interface devices and also signals the end of this volume. We began this volume by covering digital data signal characteristics, electrical interface standards, and conditioning devices. From there, we introduced you to a major source for these data signals, communications computers, where you learned about the different types and how they work. Finally, we showed you some of the protocols, architectures, and digital interface devices used in data communication processing. This also ends the first half of the 5-level course so good luck on the course examination.

---

### Self-Test Questions

After you complete these questions, you may check your answers at the end of the unit.

#### 629. Systems interface devices

1. What is the primary function of a modulator/demodulator?
2. What type of modem do you use for high-speed data?

3. What is the determining factor as to whether a serial-bit-stream modem is used over a parallel-bit-stream modem?
4. Which modem has been accepted as the standard?
5. What is the function of a line driver?
6. What is the advantage in the output of a line driver being a balanced signal?
7. Why must line drivers be located within a user's building?
8. What is the purpose of the buffer?
9. What two items make up the buffer storage unit?
10. Name the different buffering strategies that are used.
11. Why do current systems refrain from using the chained storage buffer system?
12. What is the purpose of circuit switches?
13. List the three basic characteristics of circuit switching.
14. What term do we use when a circuit is dropped and returned to a pool of available circuits?
15. What is the purpose of a concentrator?

16. Why is it more economical to use a concentrator instead of a multiplexer?
17. What must we consider as we plan the use of the concentrator for service to remote stations?
18. What is the main difference between the input signal of a concentrator and the input signal of a multiplexer?

### **630. Access interface devices**

1. What access interface device relieves the host computer from having to handle message and network control functions?
2. What conditions warrant the use of a telegraph hub?
3. In a data hub, what device prevents complete interruption of a hubbing facility when one circuit leg fails?
4. Where is the network interface unit physically located within the system?
5. What is the purpose of the network interface unit?
6. What is meant by direct-memory access?
7. What is the purpose of a gateway?
8. What are the advantages of a gateway?

9. Why is protocol incompatibility a major disadvantage of gateways?
10. What are the two types of gateways?
11. What is the purpose of a network bridge?
12. Why are network bridges received so favorably by financial managers?
13. In what types of configurations are multidrop networks available?
14. Match the functions in column A with their associated access interface devices in Column B. Column B items are used only once.

*Column A*

- \_\_\_ (1) Connects terminals to network cabling and provides impedance matching.
- \_\_\_ (2) Stores files in a queue and prints them on a first-in/first-out or priority basis.
- \_\_\_ (3) Provides local area network users the means to share software programs on an as-needed basis.
- \_\_\_ (4) Provides protocol conversion between terminals and network backbones.
- \_\_\_ (5) Provides a common storage area for local area network user data files.
- \_\_\_ (6) Recreates attenuated data signals.
- \_\_\_ (7) Selects message transmission paths using network mapping.
- \_\_\_ (8) Provides local area network users the means to exchange data and has intelligence.
- \_\_\_ (9) Connects non-intelligent devices to local area networks.

*Column B*

- a. Transceiver.
- b. Media access unit.
- c. Repeater.
- d. Router.
- e. Disk server.
- f. File server.
- g. Application server.
- h. Communication server.
- i. Print server.

## Answers to Self-Test Questions

### 624

1. An attempt to provide faultless communications.
2. (1) Vertical redundancy check (VRC), (2) longitudinal redundancy check (LRC), (3) checksum, and (4) cyclic redundancy check (CRC).
3. There is always an odd number of "1" bits in each character.
4. About 90 percent effective.
5. Vertical redundancy check (VRC) and longitudinal redundancy check (LRC).
6. Checksum.
7. The remainder of the calculation.
8. Cyclic redundancy check.

### 625

1. Character-oriented, bit-oriented, and packet-oriented.
2. Handshaking.
3. The header, text, and trailer (BCC).
4. SOH and STX.
5. A block check character used for error control.
6. It allows the receiver to recognize control characters associated with the message. Until the receiver receives DLE, it ignores the control characters.
7. NAK — data not accepted, retransmit previous block.
8. SDLC and HDLC.
9. The flag field is the first character of a frame; the receiver uses it to count down the incoming bit stream to identify the fields within the frame.
10. 01111110.
11. The commands and responses required for control of a data link.
12. They're the same.
13. It breaks the datagram down into smaller units, called datagram fragments, and reassembles the datagram at its destination.
14. Internet datagram service, virtual network service, and error reporting.
15. Virtual service.
16. Transparent data transfer between hosts within a subnetwork and error reporting.
17. The need for other transport protocols to use the internet protocol's services.
18. Error detection, positive acknowledgments, sequence numbering, and flow control.
19. An abort closing.

### 626

1. Personal computers (PCs).
2. Odd, even, or none; has block parity option.

3. This indicates a BREAK and the transmitter stops transmitting.
4. 128.
5. NAK.
6. All versions are compatible with each other and its efficiency over satellite circuits.
7. Relay Gold.

**627**

1. The International Standards Organization (ISO).
2. Synchronous unbalanced mode, asynchronous unbalanced mode, and asynchronous balanced mode.
3. (1) c, (2) g, (3) a, (4) d, (5) b, (6) f, (7) e.

**628**

1. Network addressable unit (NAU).
2. System services control point.
3. Physical unit.
4. Logical unit.
5. It is used to connect communications controllers to one another, to terminal cluster control units, to batch terminals, and to some individual display devices.
6. Communication controller.
7. Communication controller and cluster controller.
8. To serve as backups for one another.
9. To allow for greater flexibility to update, revise, and make correction to the communication process.
10. The physical, data link, path control network, transmission control services, data flow control services, presentation services, and transaction services.
11. The path control layer.
12. The transmission control services layer.
13. Full-duplex, half-duplex flip-flop, and half-duplex contention.
14. Presentation layer.
15. Transaction services layer.

**629**

1. The primary function is the modulation/demodulation of carrier signals so that digital information can be transmitted over an analog communications link.
2. Synchronous modem.
3. Bandwidth of the transmission media.
4. Serial modem.
5. To modify the various electrical voltage and current levels that may exist between the communications channel and the different hardware communications devices.
6. A balanced line signal is capable of being transmitted over much greater distances.

7. Line drivers need DC continuity.
8. To serve as an isolation device between two different circuits and, on some occasions, to supply amplification.
9. The buffer and temporary storage area.
10. Linear storage buffers, circular storage buffers, and chained storage buffers.
11. It requires a complicated input-output channel programming and critical network-control program response time.
12. To have networks that decide on how to share a communications facility.
13. It involves the establishment of a total path at all initiations; the circuit is established by a special signaling message that threads its way through different switching centers; and it is subject to the speed and code limitations of the slowest link.
14. Termination.
15. It is used as a switching device that provides for connections between a large amount of subscriber lines and a small amount of talking trunks.
16. Only one is needed at the end of the line, and it can handle more lines.
17. Its efficiency, reconfiguration capabilities, configuration of line interfaces, diagnostics, and traffic loading.
18. The incoming lines of a multiplexer are the same speed, and the incoming lines of a concentrator may be various speeds.

### 630

1. Front-end processor.
2. When there are several teletypewriter circuits being fed by one source device, or similar situations.
3. The annuller.
4. It is located between the host computer and the network.
5. It acts as a node and becomes the point where all the terminals link with the network.
6. This is when data transfers directly from the communication controller on the card into the printed-circuit's memory.
7. To serve as a connecting link between two separate networks.
8. Provides uniform access to a variety of resources, uniform access to specialized resources, and the lower cost resulting from having a single access port to other local networks.
9. Possible difficulties exist in finding matching equivalent sets of services at all levels of protocol.
10. Media conversion and protocol translation.
11. To serve as a tie point for a multidrop network where a central processing unit communicates with many modems scattered throughout a large geographical area.
12. Network bridges minimize line costs while providing each remote operator with a reasonable approximation of an on-line interactive environment.
13. Two-wire and four-wire.
14. (1) a, (2) i, (3) g, (4) b, (5) e, (6) c, (7) d, (8) f, (9) h.

## Unit Review Exercises

**Note to Student:** Consider all choices carefully, select the *best* answer to each question, and *circle* the corresponding letter. When you have completed all unit review exercises, transfer your answers to ECI Form 34, Field Scoring Answer Sheet.

**Do not return your answer sheet to ECI.**

72. (624) Vertical redundancy check (VRC) performs error control by checking each data
- a. character for even parity.
  - b. character for odd parity.
  - c. block for even parity.
  - d. block for odd parity.
73. (624) Longitudinal redundancy check (LRC) performs error control by checking each data
- a. block for odd parity.
  - b. block for even parity.
  - c. character for odd parity.
  - d. character for even parity.
74. (624) Which data error control technique is most effective for data blocks greater than 512 characters?
- a. Longitudinal redundancy check (LRC).
  - b. Vertical redundancy check (VRC).
  - c. Cyclic redundancy check (CRC).
  - d. Checksum.
75. (625) All of the following control characters are used in a character-oriented protocol *except*
- a. ETX.
  - b. SOH.
  - c. SOX.
  - d. STX.

76. (625) Which protocol type is based on bit positioning instead of control characters?
- Character-oriented.
  - Packet-oriented.
  - Block-oriented.
  - Bit-oriented.
77. (625) The information format is used in the control field of a bit-oriented protocol when
- the primary station transmits data.
  - information is transmitted between primary stations.
  - information is transmitted between secondary stations.
  - information is transmitted between primary and secondary stations.
78. (625) The internet protocol (IP) transmits blocks of data in the form of
- complete messages.
  - datagrams.
  - segments.
  - words.
79. (625) The transfer control protocol (TCP) transmits data in the form of
- words.
  - segments.
  - datagrams.
  - complete messages.
80. (626) What asynchronous modem protocol is the most widely used protocol for data transfer between microprocessors?
- BLAST.
  - KERMIT.
  - XMODEM.
  - XON/XOFF.
81. (626) Which of these modem protocols would you use on a circuit if you need two-channel capability?
- BLAST.
  - XMODEM.
  - Relay Gold.
  - TTY emulation.

- 
82. (627) Open System Interconnect (OSI) layer two operates in synchronous unbalanced mode, asynchronous unbalanced mode, and
- isochronous balanced mode.
  - asynchronous balanced mode.
  - isochronous unbalanced mode.
  - bisynchronous unbalanced mode.
83. (627) In the OSI virtual circuit service, the network layer provides
- only sequential delivery if requested.
  - no sequential delivery of packets sent.
  - virtually the same service as datagram service.
  - sequential delivery of all packets sent through the subnetwork.
84. (627) Which Open System Interconnect (OSI) layer provides data compacting, code conversion, and data encryption?
- Sessions layer.
  - Transport layer.
  - Application layer.
  - Presentation layer.
85. (628) Which component within the Standard Network Architecture (SNA) network structure contains all the tables associated with routing traffic?
- The logical unit.
  - The physical unit.
  - The system services control point.
  - The master internetwork control point.
86. (628) The Standard Network Architecture (SNA) component that represents an end user to the network is the
- logical unit.
  - physical unit.
  - network addressable unit.
  - system services control point.
87. (628) A communications controller selects the individual link that will carry a particular message on
- transmission groups.
  - explicit routes.
  - virtual routes.
  - SDLC links.

88. (628) On a complex SNA network using more than one host computer, what connects the communications controllers?
- Virtual routes.
  - Sub-area nodes.
  - Peripheral links.
  - Transmission groups.
89. (628) How many architectural layers must a message pass through at the sending and receiving node?
- One.
  - Three.
  - Five.
  - Seven.
90. (628) Which architectural layer ensures that the sending network addressable unit (NAU) does *not* transmit more data than the receiving NAU can handle?
- Data link.
  - Path control.
  - Data flow control services.
  - Transmission control services.
91. (629) Which modem is *more* adaptable for high-speed data use?
- Isochronous.
  - Synchronous.
  - Asynchronous.
  - Bisynchronous.
92. (629) Which operational requirement mandates that line drivers be used on circuits within a confined area?
- It uses the power source at the user.
  - It has a large inherent signal loss.
  - It causes distortion if overdriven.
  - It requires DC continuity.
93. (629) Within a telecommunications network node, messages are temporarily stored in areas called
- rapid-access memory units.
  - transient memory units.
  - buffer storage units.
  - virtual disk units.

- 
94. (629) The line switching method that involves allocating a total path for message transmission is
- packet switching.
  - circuit switching.
  - message switching.
  - dedicated line switching.
95. (629) Concentrators differ from multiplexers in that
- they are only needed at one end of the line.
  - they generally deal in multiple protocols.
  - their up-line consists of a single line.
  - they handle fewer lines.
96. (630) What network access device relieves mainframes by performing routine tasks such as message handling and network control?
- Front-end processors.
  - Network hubs.
  - Repeaters.
  - Gateways.
97. (630) Which interface unit is located between a host computer and the network?
- Line level.
  - Network.
  - Hybrid.
  - Code.
98. (630) Which is the connecting link between two separate computer networks?
- Network interface unit.
  - The combiner network.
  - The hubbing unit.
  - The gateway.
99. (630) The network access interface device that establishes a path between two dissimilar protocols is called a
- gateway.
  - bridge.
  - router.
  - hub.

- 
- 
100. (630) What are *two* types of gateways?
- Media conversion and protocol translation.
  - Protocol translation and format matching.
  - Media conversion and level translation.
  - Level translation and format matching.
101. (630) Multidrop network bridges are available in which configurations?
- Two-wire and six-wire.
  - Four-wire and six-wire.
  - Two-wire and four-wire.
  - Six-way wire and nine-wire.
102. (630) What type of interface device is used to recreate attenuated data signals?
- Amplifier.
  - Repeater.
  - Router.
  - Server.
103. (630) What type of server is used as a shared storage unit for network users?
- Disk.
  - Print.
  - Application.
  - Communication.
104. (630) The type of server that allows users to share software programs is
- a file server.
  - a disk server.
  - an application server.
  - a communication server.
105. (630) A device that connects printers to a network for resource sharing among network users is a
- gateway.
  - print server.
  - hubbing device.
  - application server.

**STUDENT NOTES**