

TC 9-72

**COMMUNICATIONS-ELECTRONICS
FUNDAMENTALS**

DIGITAL COMPUTERS

SEPTEMBER 2005

DISTRIBUTION RESTRICTION: Approved for public release; distribution is unlimited.

**HEADQUARTERS
DEPARTMENT OF THE ARMY**

**This publication is available at
Army Knowledge Online (www.us.army.mil) and
General Dennis J. Reimer Training and Doctrine
Digital Library at (<http://www.train.army.mil>).**

COMMUNICATIONS-ELECTRONICS FUNDAMENTALS

DIGITAL COMPUTERS

Contents

	Page
PREFACE	v
CHAPTER 1 OPERATIONAL CONCEPTS	1-1
Introduction	1-1
Operating Principles	1-1
History of Computers	1-2
Summary	1-20
CHAPTER 2 HARDWARE	2-1
Introduction	2-1
Central Processing Unit	2-2
Control Section	2-2
Arithmetic-Logic Section	2-3
Memory (Internal Storage) Section	2-3
Printers (Output)	2-15
Keyboards (Input)	2-17
Display Devices	2-18
Summary	2-23
CHAPTER 3 SOFTWARE	3-1
Introduction	3-1
Computer Programs	3-1
Operating Systems	3-1
Utility Programs	3-7
Programming Languages	3-9
Machine Languages	3-9

Distribution Restriction: Approved for public release; distribution is unlimited.

*This publication supersedes FM 11-72, 30 September 1977.

	Page
Symbolic Languages.....	3-10
Procedure-Oriented Languages	3-10
Programming.....	3-11
Overview of Programming	3-12
Flowcharting.....	3-13
Program Coding.....	3-19
Summary.....	3-28
CHAPTER 4 DATA REPRESENTATION AND COMMUNICATIONS.....	4-1
Introduction	4-1
Data.....	4-1
Computer Coding Systems	4-3
Data Storage Concepts.....	4-7
Storage Access Methods	4-12
Networks	4-14
Summary.....	4-17
APPENDIX A CHECK-ON-LEARNING ANSWERS.....	A-1
GLOSSARY	Glossary-1
REFERENCES.....	References-1
INDEX	Index-1

Figures

	Page
CHAPTER 1	
Figure 1-1. Abacus	1-2
Figure 1-2. Bulkhead-Type Mechanical Computer.....	1-4
Figure 1-3. Electromechanical Computer.....	1-5
Figure 1-4. Electronic Digital Computer	1-6
Figure 1-5. Digital Computation.....	1-9
Figure 1-6. Analog Computation	1-9
Figure 1-7. First Generation Computers Used Vacuum Tubes.....	1-11
Figure 1-8. Second Generation Computers Used Transistors.....	1-11
Figure 1-9. Third Generation Computers Used Microcircuits.....	1-12
Figure 1-10. Fourth Generation Desktop Personal Computer	1-13
Figure 1-11. Programming Flowchart Used to Build a Payroll Program	1-15
Figure 1-12. Floppy Disk	1-18
Figure 1-13. CD-ROM Disk	1-18

	Page
CHAPTER 2	
Figure 2-1. Functional Units of a Computer System	2-1
Figure 2-2. Memory Locations.....	2-4
Figure 2-3. Two-State Principle of Magnetic Storage	2-5
Figure 2-4. A Semiconductor Memory Chip (Integrated Circuit)	2-6
Figure 2-5. Bubble Memory	2-7
Figure 2-6. Location of Tracks on the Disk's Recording Surface.....	2-9
Figure 2-7. A String of Bits Written to Disk on a Single Track	2-10
Figure 2-8. Data Records as They are Written to Disk on a Single Track	2-10
Figure 2-9. Physical Organization of Data on a Disk (Cylinder Method)	2-11
Figure 2-10. Physical Organization of Data on a Disk (Sector Method)	2-12
Figure 2-11. Multiple Access Arms and Read/Write Heads Used With Disk Packs	2-14
Figure 2-12. Floppy Disk Drive Unit	2-14
Figure 2-13. Printer.....	2-15
Figure 2-14. Dot-Matrix Printing	2-16
Figure 2-15. Keyboard Combined With a CRT and Microcomputer	2-17
Figure 2-16. Keyboard Layout.....	2-18
Figure 2-17. A 7 by 9 Picture Element Character	2-20
CHAPTER 3	
Figure 3-1. Printed Report Using a RPG Program.....	3-8
Figure 3-2. Evolution of a Program	3-13
Figure 3-3. System Flowchart	3-14
Figure 3-4. Programming Flowchart.....	3-15
Figure 3-5. Fundamental Flowcharting Symbols.....	3-16
Figure 3-6. Flowchart Template	3-17
Figure 3-7. Problem Definition and Programming Flowchart	3-18
Figure 3-8. Programming Flowchart and Coded Program	3-21
Figure 3-9. Word Processing Example	3-24
Figure 3-10. Data Management Example (Prompts in Bold and Data In Italics)	3-25
Figure 3-11. Data Management Example (Sample Printed Report Sorted by Last Name) .	3-25
Figure 3-12. Data Management Example (Calculation of Inventory Value).....	3-26
Figure 3-13. Spreadsheet Example.....	3-26
Figure 3-14. Graphics Examples (Pie Chart)	3-27
Figure 3-15. Graphics Examples (Bar Chart)	3-27

	Page
CHAPTER 4	
Figure 4-1. Communications Symbols	4-2
Figure 4-2. UPC Bar Code	4-3
Figure 4-3. Format for EBCDIC and ASCII Codes.....	4-3
Figure 4-4. Eight-Bit EBCDIC Coding Chart (Including Hexadecimal Equivalents).....	4-4
Figure 4-5. DP-3 Represented Using 8-Bit EBCDIC Code	4-5
Figure 4-6. Packed Data	4-5
Figure 4-7. Eight-Bit ASCII Coding Chart (Including Hexadecimal Equivalents).....	4-6
Figure 4-8. Core Plane	4-8
Figure 4-9. Core Magnetized in One Direction.....	4-8
Figure 4-10. Core Planes Arranged Vertically to Represent Data	4-9
Figure 4-11. Fixed-Word-Length Versus Variable-Word-Length Storage, Fixed-Length Words, Containing Eight Characters Each, Occupying Two Address Locations (Word Addressable)	4-10
Figure 4-12. Fixed-Word-Length Versus Variable-Word-Length Storage, Variable-Length Words (Character Addressable)	4-10
Figure 4-13. Word Lengths Used on Flexible Byte-Addressable Computers	4-13
Figure 4-14. Data Organization	4-13
Figure 4-15. Local Area Network System	4-14
Figure 4-16. Modem	4-15
Figure 4-17. Modems Used in Network System.....	4-16

Tables

	Page
CHAPTER 4	
Table 4-1. DP-3 Characters in EBCDIC and ASCII	4-5

Preface

The objective of this training circular (TC) is to describe the basic principles, techniques, and procedures associated with digital computers. This TC covers operational concepts of digital computers along with the hardware, software, and data representation and communications used with digital computers.

Check-on-learning questions are included at the end of each chapter. Appendix A (Check-on-Learning Answers) is included to provide answers to the check-on-learning questions from each chapter.

This training circular applies to the Active Army, the Army National Guard/Army National Guard of the United States, and the United States Army Reserve.

The proponent of this publication is the United States Army Training and Doctrine Command (TRADOC). Submit changes for improving this publication on DA Form 2028 (Recommended Changes to Publications and Blank Forms) and forward it to:

Department of the Army
Training Directorate, Fixed/Arm Division
401 1st Street, Room 225
Fort Lee, VA 23801-1511

Unless this publication states otherwise, masculine noun and pronouns do not refer exclusively to men.

This page intentionally left blank.

Chapter 1

Operational Concepts

INTRODUCTION

1-1. Digital computers are used in many facets of today's Army. It is impossible for one training circular (TC) to cover in depth all the ways they are used. However, this chapter will cover a few of the ways.

OPERATING PRINCIPLES

1-2. The operating principles of personal computers (PCs) relate directly to the operating principles of mainframe computers. Since PCs are more widely used than the large mainframes, the desktop PC will be used for most of the examples. When you have satisfactorily completed these chapters, you will learn the basic terminology used in the digital computer world. You will also have a better understanding of how computers are able to perform the demanding tasks assigned to them.

1-3. The word "computer" could be defined as an instrument for performing mathematical operations at very high speeds. Some of these operations include the following:

- Addition.
- Subtraction.
- Multiplication.
- Division.
- Integration.
- Vector resolution.
- Coordinate conversion.
- Special function generation.

However, the use of computers goes well beyond the mathematical operations level.

1-4. Computers have made military, scientific, and commercial advances possible that before were considered impossible. For example, the mathematics involved in orbiting a satellite around the earth would require a lifetime of several teams of mathematicians. With the aid of electronic digital computers, the conquest of space has now become a reality.

1-5. Computers are used when repetitious calculations or the processing of large amounts of data are necessary. The most frequent applications are found in the military, scientific, and commercial fields. They are used in many varied projects. The range from mail sorting, through engineering design, to the identification and destruction of enemy targets. The following are some advantages of digital computers:

- Speed.
- Accuracy.
- Reliability.
- Manpower savings.

Computers are frequently able to take over routine jobs. This allows people to perform more important work, work that cannot be handled by a computer.

HISTORY OF COMPUTERS

1-6. The ever-increasing need for faster and more efficient computers has created amazing technological advances. Ever since man discovered that it was necessary to count objects, he has been looking for easier ways to count. Contrary to popular belief, digital computers are not a new idea. The abacus (see Figure 1-1) is a manually operated digital computer used in ancient civilizations and still currently used in the Orient. For those who consider the abacus outdated, in a contest between a person using a modern calculator and a person using an abacus, the person using the abacus won.

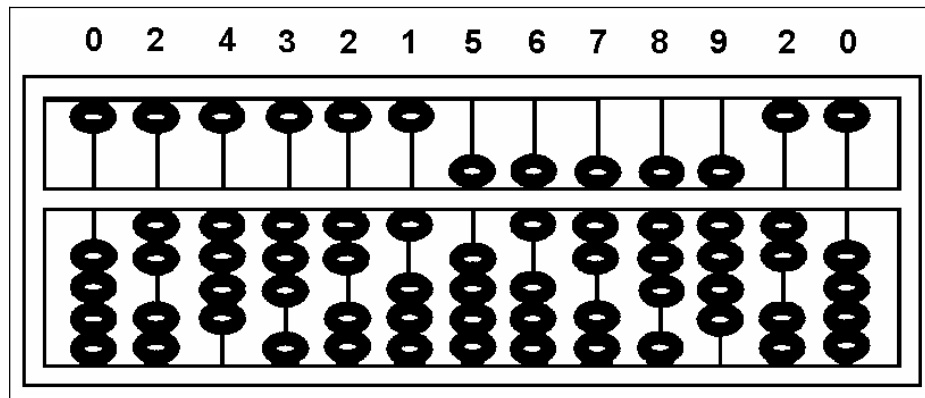


Figure 1-1. Abacus

1-7. In 1642, a Frenchman (Blaise Pascal) invented the first mechanical adding machine (calculator). Twenty years later, an Englishman (Sir Samuel Morland) developed a more compact device that could add, subtract, and multiply. In 1682, a German (Wilhelm Liebnitz) perfected a machine that could perform all the basic operations (addition, subtraction, multiplication, and division), as well as extract the square root. Liebnitz's principles are still in use today in modern electronic digital computers.

1-8. Electronics were introduced to the computer field as early as 1919. An article by W. H. Eccles and F. W. Jordan described an electronic "trigger circuit" that could be used for automatic counting. It was called the Eccles-Jordan multivibrator. This was a little ahead of its time because a trigger circuit is one of many components required to make an electronic digital computer. Modern digital computers use these circuits (known as flip-flops) to store information, perform arithmetic operations, and control the timing sequences within the computer.

1-9. Under the pressure of military needs in World War II, the science of electronic data processing made giant strides forward. In 1944, Harvard University developed a computing system known as the Automatic Sequence Controlled Calculator. After the initial design and construction, several improved models were built.

1-10. Meanwhile, at the University of Pennsylvania, a second system was being developed. This system, completed in 1946, was named "ENIAC". ENIAC used 18,000 vacuum tubes in its circuitry. In spite of these bulky, hot tubes, it worked quite successfully. The first problem assigned to ENIAC was a calculation in nuclear physics that would have taken 100 years to solve by conventional methods. The ENIAC solved the problem in two weeks, only two hours of which were actually spent on the calculation. The remainder of the time was spent checking the results and operational details. All modern computers are based on these two early developments conducted at Harvard University and the University of Pennsylvania.

1-11. The UNIVAC I was developed in 1950. This machine was usually regarded as the most successful electronic data processor of its day. An outstanding feature of the UNIVAC I was that it checked its own results in each step of a problem. This eliminated the need to run the problems more than once to ensure accuracy.

1-12. During the first outbreak of publicity about computers (especially when the UNIVAC predicted the outcome of the 1952 presidential election), the term "giant brain" caused much confusion and uneasiness. Many people assumed that science had created a thinking device superior to the human mind. Most people now know better and realized that this "giant brain" was wholly dependent upon human instructions to perform even the simplest jobs. A computer is only a machine and definitely cannot think for itself. However, the field of artificial intelligence is developing computer systems that can "think." That is, the systems can mimic human thought in a specific area and exhibit improved performance with experience and operation. The field of digital computers is still in the growing stages. New types of circuitry and new ways of accomplishing things are continuing to be developed at a rapid rate.

1-13. In the military field, the accomplishments of digital computers are many and varied. One outstanding example is in weapons systems. Most of the controlling is done by digital computers.

CLASSIFICATIONS OF COMPUTERS

1-14. Computers can be classified in the following ways:

- By the type of technology they use (mechanical, electromechanical, or electronic).
- The purpose for which they were designed (general purpose or special purpose).
- By the type of data they can handle (digital or analog).
- By the amount they cost (from hundreds of dollars and up).
- By their physical size (handheld to room size).

The following types of computers will be briefly explained:

- Mechanical, electromechanical, and electronic.
- Special purpose and general purpose.
- Analog and digital.

MECHANICAL COMPUTERS

1-15. Mechanical computers are devices used for the computation of mathematical problems. They are made up of components such as integrators, sliding racks, cams, gears, springs, and driveshafts. Figure 1-2 shows a typical mechanical computer used by the Army. These computers are analog in nature. Their physical size depends on the number of functions the computer has to perform. In an analog computer, a continuing input will give a constantly updated output. Since the analog computer is perfect for target information, the Army uses this type of computer primarily for fire control. As weapon systems become more and more complex, the need for different computers becomes apparent. Functions that now have to be performed have increased the size of the computer to an unreasonable scale.

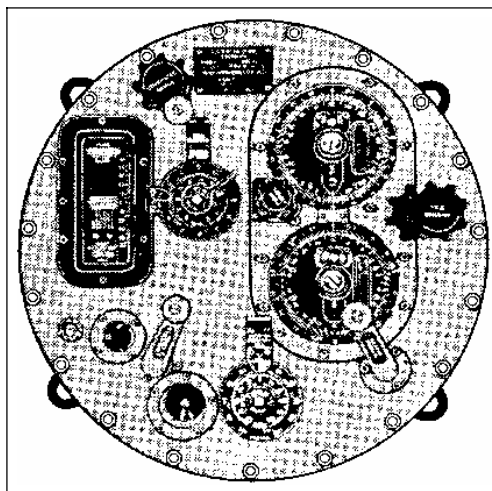


Figure 1-2. Bulkhead-Type Mechanical Computer

ELECTROMECHANICAL COMPUTERS

1-16. Electromechanical computers came next. These computers were different from mechanical computers in that they use electrical components to perform and to increase the accuracy of some of the calculations. Since electrical components are smaller than their mechanical counterparts, the size of the computer was reduced, even though it performs more functions. The components used to perform the calculations are such devices as:

- Synchros.
- Servos.
- Resolvers.
- Amplifiers.
- Servo amplifiers.
- Summing networks.
- Potentiometers.
- Linear potentiometers.

Figure 1-3 shows one of the Army's electromechanical computers. These computers are used in gun fire control and missile fire control. Even though they are better than the mechanical computer, they still have their drawbacks. The prime importance of these computers is that they are special purpose computers. This means they can only be used for one job, dependent on their design characteristics. By today's Army standards, they are still too large and the maintenance time on them is excessive. The need for a more accurate, reliable, versatile, and smaller computer was recognized.

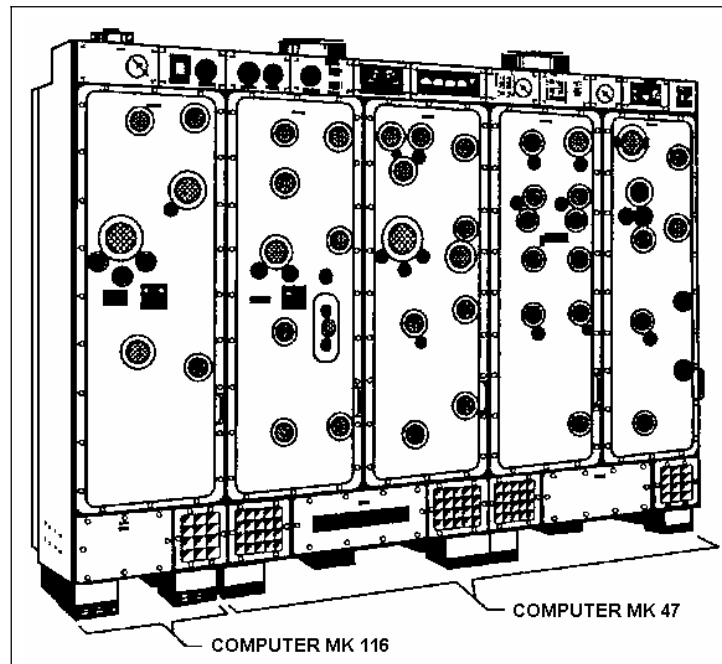


Figure 1-3. Electromechanical Computer

ELECTRONIC COMPUTERS

1-17. Electronic computers came next. The early electronic computers mathematical processes were solved by only using electrical voltages applied to elements such as amplifiers, summing networks, differentiating, and integrating circuits. The weak link in this type of electrical computation was the vacuum tube. To correct this, transistors that consume less power and last longer than vacuum tubes were used in the amplifiers. Through technological research and development, electronic computers have progressed from tubes, to transistors, to miniaturized circuits, to integrated circuitry. These advances have made it possible to reduce the size and weight of computers. Figure 1-4 is an example of one of many modern electronic digital computers.

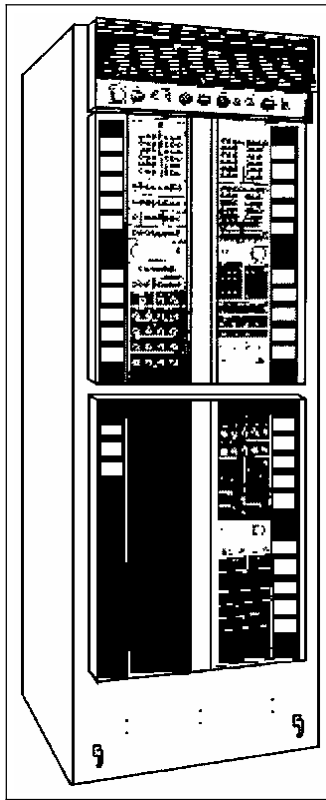


Figure 1-4. Electronic Digital Computer

SPECIAL PURPOSE COMPUTERS

1-18. A special purpose computer, as the name implies, is designed to perform a specific operation and usually satisfies the needs of a particular type of problem. Such a computer system would be useful in weather predictions, satellite tracking, or oil exploration. While a special purpose computer may have many of the same features found in a general purpose computer, its applicability to a particular problem is a function of its design rather than a stored program. The instructions that control it are built directly into the computer. This makes for a more efficient and effective operation. However, a drawback of this specialization is the computer's lack of versatility. It cannot be used to perform other operations.

GENERAL PURPOSE COMPUTERS

1-19. Most people will probably use this type of computer. General purpose computers are designed to perform a wide variety of functions and operations. The reason they can perform a wide variety of operations is because they can store and execute different programs in its internal storage. Unfortunately, having this ability is often achieved at the expense of speed and efficiency. However, in most situations, the flexibility of these types of computers makes this compromise a most acceptable one.

ANALOG COMPUTERS

1-20. All analog computers are special purpose computers. They are designed to measure continuous electrical or physical conditions (such as current, voltage, flow, temperature, length, or pressure). They then convert these measurements into related mechanical or electrical quantities. The early analog computers were strictly mechanical or electromechanical devices. They did not operate on digits (in binary notation, either of the characters, 0 and 1). If digits were involved at all, they were obtained indirectly. A wristwatch (if nondigital) as well as a car's speedometer, oil pressure, temperature, and fuel gauges are also considered analog computers. The output of an analog computer is often an adjustment to the control of a machine (such as an adjustment to a valve that controls the flow of steam to a turbine generator or a temperature setting to control the ovens in a ship's galley for baking). Analog computers are also used for controlling processes. To do so, they must convert analog data to digital form, process it, and then convert the digital results back to analog form.

1-21. A digital computer can process data with greater accuracy than an analog computer. However, in some systems, an analog computer can process data faster than a digital computer. Some computers combine the functions of both analog and digital computers. They are called hybrid computers.

DIGITAL COMPUTERS

1-22. Digital computers perform arithmetic and logic functions on separate discrete data (like numbers) or combinations of discrete data (such as names, rates, and divisions). This makes them different from analog computers that operate on continuous data, like measuring temperature changes. Digital computers may be either special purpose or general purpose. Word processing is among the most common applications for personal computers. Digital computers are generally used for business and scientific data processing. The following are some examples when digital computers are used:

1-23. **Accounting.** Computers are ideal for keeping payroll records, printing paychecks, billing customers, preparing tax returns, and taking care of many other accounting tasks.

1-24. **Recordkeeping.** Computers can record information like inventories and personnel files. They can also keep track of books checked out of a library. Airline ticket counters today are much more efficient than they used to be, thanks to centralized reservation computers that can be reached over the telephone lines.

1-25. **Industrial Uses.** Industrial computers save considerable time and reduce waste by efficiently performing hundreds of industrial tasks. Some of these tasks range from filling sales orders, routing parts to various locations on an assembly line, designing earthquake-resistant structures, and controlling an entire oil refinery.

1-26. **Science.** Research and development applications are the most numerous. Digital computers are being used to do lengthy and complicated mathematical calculations millions of times faster than people. They are also used in the following ways:

- Collect, store, and evaluate data from experiments.
- Analyze weather patterns.
- Forecast crop statistics.
- Design other computers.

COMPUTER PROGRAMS

1-27. A computer must first have instructions loaded before it can perform any work. This is done by means of a list of instructions called a program. The instructions in the program must be written in one of the languages the computer understands. The most popular generic term for computer programs is “software” (this is covered in Chapter 3). Hardware (covered in Chapter 2) refers to the computer and related equipment. It is easy to say that both computer hardware and software are interdependent because neither can perform without the other.

ACCURACY OF COMPUTERS

1-28. The fundamental difference between analog and digital computers is that digital computers deal with discrete quantities (such as beads on an abacus, notches on a toothed wheel, or electrical pulses). However, analog computers deal with continuous physical variables (such as electrical voltages or mechanical shaft rotations). Computation with analog computers depends on the relation of information to a measurement of some physical quantity. For example, the number of boards in a picket fence can be determined by either a digital or an analog system. In the digital method (see Figure 1-5), an adding machine is used when the boards are counted one by one. In the analog method (see Figure 1-6), a string is drawn (marked off in inches for the width of each board including the gap) over the length of the fence, and then measure the length of the string. The number of boards may then be determined by dividing the length of string by the number of inches per board.

1-29. The accuracy of an analog computer is restricted to the accuracy with which physical quantities can be sensed and displayed. This, in turn, is related to the quality of the components used in constructing the computer (for example, the tolerance of electrical resistors or mechanical shafts and the quality of the output equipment). In an analog computer, for example, if the constant is represented by a voltage, it probably could be read only to the third decimal place.

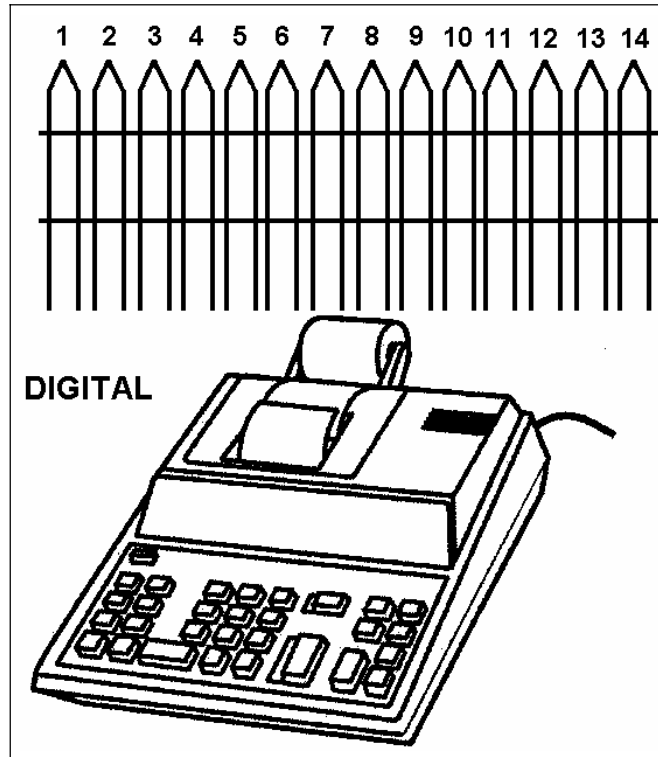


Figure 1-5. Digital Computation

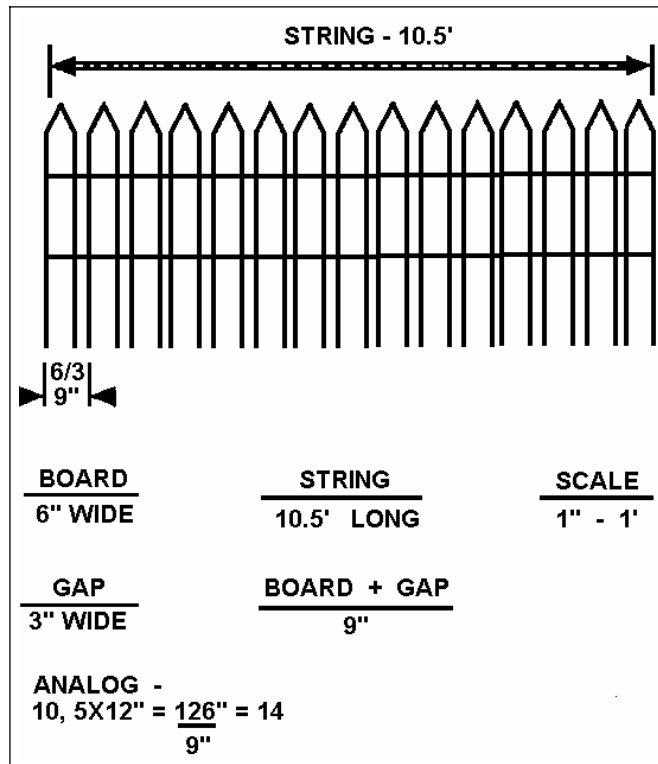


Figure 1-6. Analog Computation

1-30. However, the number of significant figures carried in the computations governs the accuracy of a digital computer. This, in turn, is determined by the computer's design. In a digital computer, the number of decimal places in the constant could be many, depending on the design of the computer-processing unit. Therefore, the digital computer is capable of higher precision and accuracy. However, a computer, regardless of its accuracy, would be of no use if the wrong one was chosen for a given task.

1-31. Most of the computer systems used will be general purpose digital computers. The remainder of this chapter will be about general purpose digital computers.

DIGITAL COMPUTER GENERATIONS

1-32. In the electronic computer world, technological advancement is measured by generations. A specific system is said to belong to a specific "generation." Each generation indicates a significant change in computer design. The UNIVAC I represents the first generation. Presently, we are in the fourth generation of computers.

FIRST GENERATION

1-33. The computers of the first generation (1951-1958) were physically very large machines characterized by the vacuum tube (see Figure 1-7). Since they used vacuum tubes, they were very unreliable, required much power to run, and produced so much heat that adequate air conditioning was critical to protect the computer parts. Compared to today's computers, they had slow input and output devices, were slow in processing, and had small storage capacities. Many of the internal processing functions were measured in thousandths of a second (millisecond). The software (computer program) used on first generation computers was unsophisticated and machine oriented. This meant that the programmers had to code all computer instructions and data in actual machine language. They also had to keep track of where instructions and data were stored in memory. Using such a machine language (see chapter 3) was efficient for the computer but difficult for the programmer.

SECOND GENERATION

1-34. The computers of the second generation (1959-1963) were characterized by transistors (see Figure 1-8) instead of vacuum tubes. Transistors were smaller, less expensive, generated almost no heat, and required very little power. Therefore, second generation computers were smaller, required less power, and produced a lot less heat. The use of small, long-lasting transistors also increased processing speeds and reliability. Cost performance also improved. Storage capacity greatly increased with the introduction of magnetic disk storage and the use of magnetic cores for main storage. High-speed card readers, printers, and magnetic tape units were also introduced. Internal processing speeds also increased. Functions were measured in millionths of a second (microseconds). Like the first generation, computers of the second generation were designed to process either scientific or business oriented problems but not both. The software was also improved. Symbolic machine languages or "assembly languages" were used instead of actual machine languages. This allowed the programmer to use mnemonic

operation codes for instruction operations and symbolic names for storage locations or stored variables. Compiler languages were also developed for second generation computers (see chapter 3).

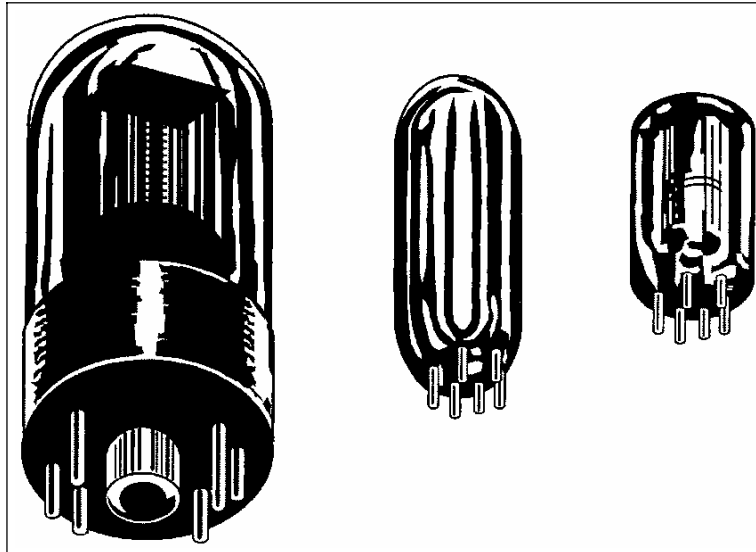


Figure 1-7. First Generation Computers Used Vacuum Tubes

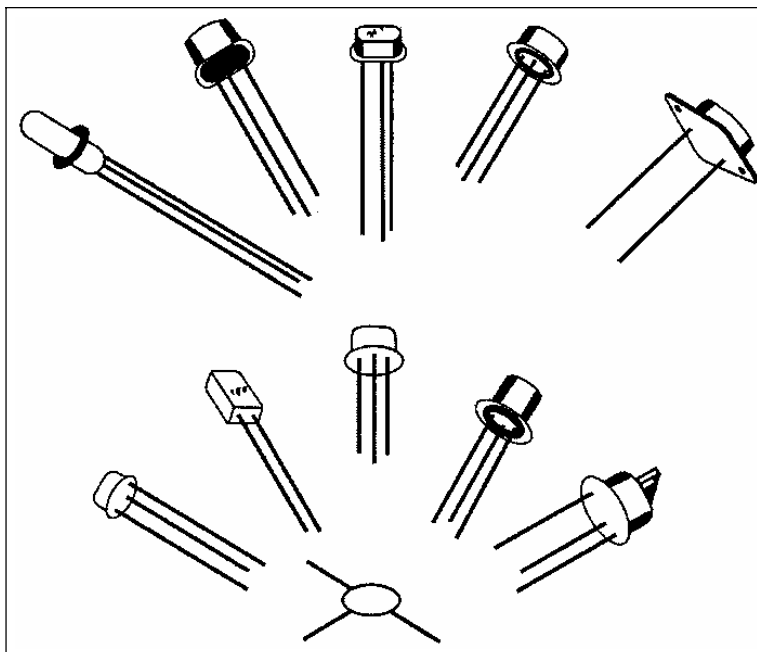


Figure 1-8. Second Generation Computers Used Transistors

THIRD GENERATION

1-35. The computers of this generation (1964-1970), many of which are still in use, were characterized by miniaturized circuits. This reduced the physical size of computers even more and increased their durability and internal processing speeds. One design used solid-state logic microcircuits (see Figure 1-9) for which conductors, resistors, diodes, and transistors had been

miniaturized and combined on half-inch ceramic squares. Another smaller design, used silicon wafers on which the circuit and its components were etched. The smaller circuits allowed for faster internal processing speeds, resulting in faster execution of instructions. Internal processing speeds were measured in billionths of a second (nanoseconds). The faster computers made it possible to run jobs that were considered impractical or impossible on first or second generation computers. Since the miniature components were more reliable, maintenance was reduced. New mass storage, such as the data cell, was introduced during this generation. This type of storage gave a storage capacity of over 100 million characters. Drum and disk capacities and speed were increased. The portable disk pack was also developed. Faster and higher density magnetic tapes also came into use. Considerable improvements were made to card readers and printers, while the overall cost was greatly reduced. Applications (covered in later chapters), using online processing, real-time processing, time-sharing, multi-programming, multi-processing, and teleprocessing became widely accepted.

1-36. Manufacturers of third generation computers produced a series of similar and compatible computers. This allowed programs written for one computer model to run on larger models of the same series. Most third generation systems were designed to handle scientific and business data processing applications. Improved program and operating software had been designed to provide better control, resulting in faster processing. These enhancements were of significant importance to the computer operator. They simplified system initialization (booting) and reduced the need for inputs to the program from a keyboard (console intervention) by the operator.

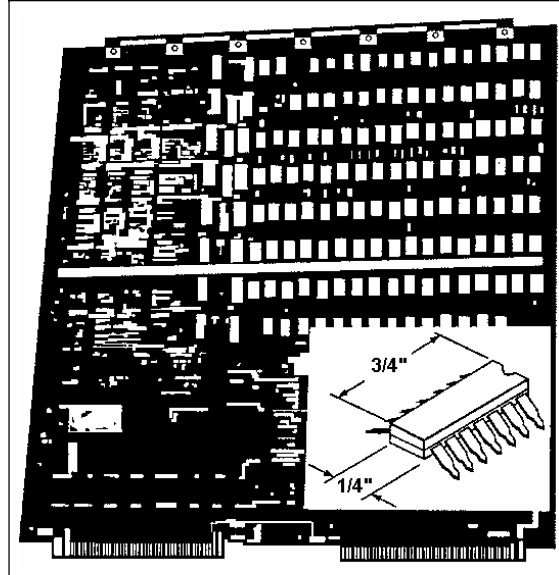


Figure 1-9. Third Generation Computers Used Microcircuits

FOURTH GENERATION AND BEYOND

1-37. The computers of the fourth generation (1971 to the present) are not easily distinguished from earlier generations, yet there are some striking and important differences. The manufacture of integrated circuits (ICs) has advanced to the point where thousands of circuits (active components) can be

placed on a silicon wafer only a fraction of an inch in size (the computer on a chip). This has led to what is called large-scale integration and very large-scale integration. As a result of this technology, computers are significantly smaller in physical size and lower in cost. Yet they have retained large memory capacities and are ultra fast. Large mainframe computers are increasingly complex. Medium-sized computers can perform the same tasks as large third generation computers. A new breed of computers (called microcomputers [personal computers]) are small and inexpensive, but yet they provide a large amount of computing power (see Figure 1-10).

1-38. Even though the computer industry still has a long way to go in the field of miniaturization, there are some things in store for the future (such as the power of a large mainframe computer on a single super chip). The future challenges will not be in increasing the storage or increasing the computer's power, but rather in properly and effectively using the computing power available. This is where software (programs such as assemblers, report generators, subroutine libraries, compilers, operating systems, and applications programs) will come into play (see chapter 3). Some believe that new development of software and in learning how to use these extraordinary, powerful machines will be far more important than further developments in hardware over the next 10 to 20 years. As a result, the next 20 years may be even more interesting and surprising than the last 20 years.

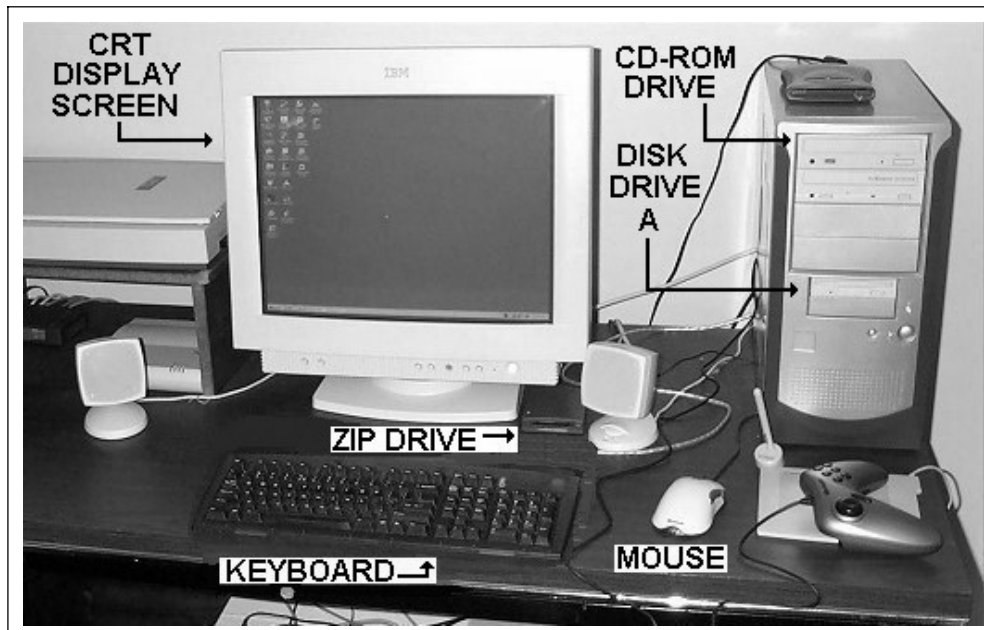


Figure 1-10. Fourth Generation Desktop Personal Computer

USES OF A DIGITAL COMPUTER

1-39. In today's modern computer world, uses of the digital computer are almost as limitless as a person's imagination. New and better programs are being written every day for easier and greater uses. Consider how many mathematicians it would take, when it takes only one computer, to put an astronaut in orbit around the moon. Think back to the days without word processing when a document had to be retyped entirely when any changes were needed. Think back to the days of using an adding machine to prepare and revise budgets and accounting reports. The primary uses of general purpose digital computers in the Army are word processing and accounting and recordkeeping.

WORD PROCESSING

1-40. The word processor can be considered a typewriter with a display screen. To the hundreds of thousands of word processor users, the computer is nothing more than a typewriter. Both have keyboards and a mechanism for making the image of the character, which is selected on the keyboard; appear on some type of visual medium. When using an electric typewriter, the process is strictly mechanical. When a key is pressed, it causes the typeface to strike the paper, and in so doing, it leaves an impression. In the computer, the process is more indirect. A program stored in the computer's memory causes a visual representation to appear on a cathode-ray tube (CRT) and then outputted to a printer. However, from the viewpoint of the user, the result is the same (a printed document).

1-41. A further breakthrough came with the development of word processing application programs for microcomputers. These programs cost a fraction of their office machine counterparts and could be run on general purpose microcomputers. This was unique because general purpose microcomputers could be used for functions such as spreadsheets, data base management systems, and programming in common computer languages.

1-42. The Army saw obvious uses for microcomputers using word processing programs. These uses include manuscript writing, memorandum writing, filing of identification card applications, and recordkeeping.

ACCOUNTING AND RECORDKEEPING

1-43. There are virtually unlimited applications for the computer in today's modern business world. These applications range from basic accounting functions to controlling the manufacture of products, and of course, keeping records of these actions. Six standard systems dealing with accounting applications are widely accepted. These systems are:

- Order entry.
- Inventory control.
- Accounts receivable.
- Accounts payable.

- General ledger.
- Payroll.

Figure 1-11 shows a simplified flowchart of payroll. The area of recordkeeping has two requirements, legal and audit. The Army has included similar functions in its Shipboard Non-Tactical ADP Program (called SNAP) for work center use.

USING A DESKTOP PERSONAL COMPUTER

1-44. Knowing about the hardware (the equipment) and the software (the programs) will help a person to effectively use a desktop PC. Also, knowing how to handle floppy and compact disk-read only memory (CD-ROM) disks and how to back up programs and data files will be effective in using a desktop PC. A desktop PC (see Figure 1-10) usually consists of a CRT display screen, a keyboard, a mouse, a central processing unit (CPU) (with a floppy disk drive and CD-ROM drive), a zip drive, and a printer (not shown). Software (computer programs) is needed to make the computer operate. The first program needed is the operating system. The operating system manages the computer and allows the application programs to run (like word processing or recordkeeping programs).

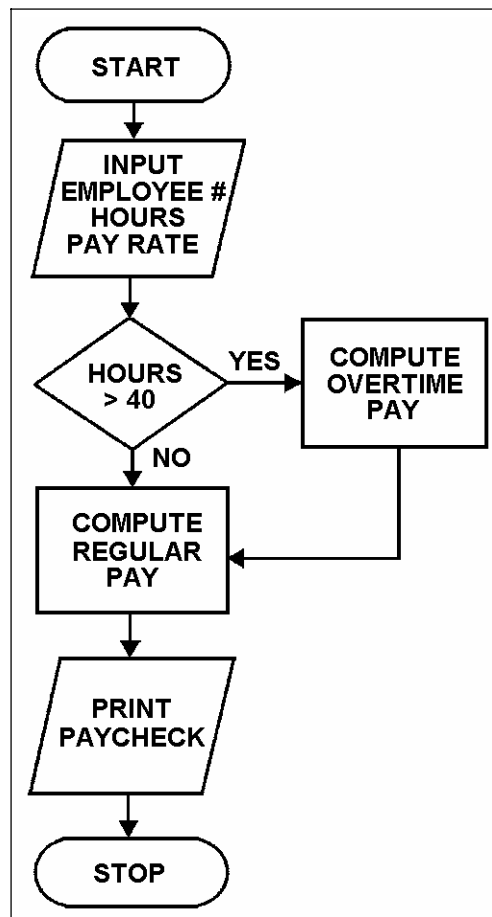


Figure 1-11. Programming Flowchart Used to Build a Payroll Program

OPERATING SYSTEM

1-45. An operating system is simply a set of programs and routines that lets a person use the computer. A digital computer uses one central set of programs called the operating system to manage execution of other programs and to perform common functions like read, write, or print. Most computers today come with a preloaded operating system. If not, an operating system, along with other programs, can be ordered to perform these common functions. These orders are called system calls when other programs use them or simply a command when put them through the keyboard.

1-46. The operating system must first be loaded into the computer so that the operator could use the computer programs. The operating system can be loaded using the floppy disk drive (disk drive A) or CD-ROM drive on the desktop PC (see Figure 1-10).

BOOTING THE SYSTEM

1-47. Each desktop PC has a built-in program called "bootstrap loader." When a computer is turned on for the first time, this program tries to load, or "boot," an operating system from disk into the computer's internal memory. The term "boot" comes from the idea of "pulling yourself up by your bootstraps". The computer loads a small program that then tells it how to load a second, bigger program (the operating system). The operating system then tells it how to load another program (an applications program or utility program) to perform a specific job or function. The first thing to know about using a computer is that computers and their programs are very particular. They require complete accuracy and attention to detail on the operator's part. They are not good at guessing what an operator meant. The operator will quickly learn there are a few things that can go wrong at this point, in which case the computer will display an error message on the display screen similar to this:

DEVICE ERROR

This means the computer is not reading anything in disk A drive. Check for the following:

- No floppy disk in drive A.
- Floppy disk inserted incorrectly in drive.

Another error message that might be displayed is:

NO SYSTEM

This means the computer is properly reading the inserted floppy disk, but there is no operating system on the disk. Replace the disk with one that does contain the operating system.

Proper Booting

1-48. Once the operating system is properly booted (loaded), the following will be displayed:

```
A>
```

The computer is displaying what is called a prompt. At this point the operator can tell the computer what to do next, such as run an application program.

Running an Application Program

1-49. To load an application program into the computer from drive A, a disk with the application program must be inserted into disk drive A. Next, the name of the program following the operating system prompt (A>) must be typed in, as follows.

```
A>WORDPROC
```

This tells the system what program to load and run (in this case, word processing). The computer then does what the application program tells it to do. If the application is word processing, the system is ready for the operator to type a new document, correct an existing document, print a document, and so on. More about the operating system and application programs are covered in chapter 3.

1-50. Each application program will have its own set of instructions to follow. In addition to printed documentation, many programs will include online HELP screens that can be displayed while working. These explain to the operator how to perform a given function or operation.

1-51. Two more areas that needs attention are the following:

- Correctly handling of floppy and CD-ROM disks.
- Making backup copies to ensure work is not lost.

STORAGE MEDIA HANDLING AND BACKUP

1-52. A 3 1/2-inch floppy disk (see Figure 1-12) and a CD-ROM disk (see Figure 1-13) are the most common ways in which to store data either directly or by backing up the data stored on hard (or fixed) disk. Since floppy and CD-ROM disks are extremely fragile, certain guidelines should be followed to ensure their proper care and handling. This includes properly labeling and backing up disks.

Handling

1-53. Never touch the exposed surface of a floppy disk or read/write side of a CD-ROM disk. A sliding bar protects the read/write surface of a 3 1/2-inch disk. Also, never slide this bar to expose the read/write disk. Touching the surface of the read/write disk could destroy the read/write capability or destroy the data that is already stored on the disk. Care must also be taken when handling CD-ROM disks. Try to avoid touching the read/write side of the disk. Handle a CD-ROM disk by its outer edges.

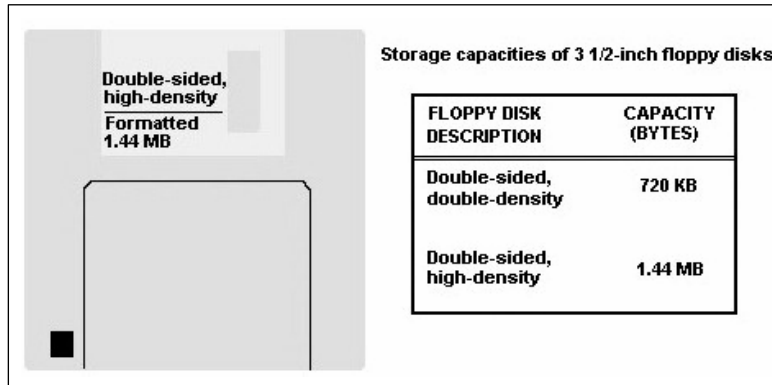


Figure 1-12. Floppy Disk

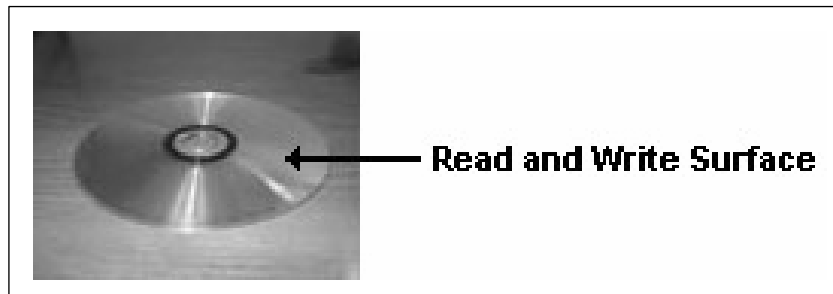


Figure 1-13. CD-ROM Disk

Storage

1-54. Never try to distort the shape of a disk. Never place heavy objects (such as books) on top of disks. Store floppy disks in the box they came in or in filing containers that are specifically designed for storing disks. Store CD-ROM disks in either a paper or plastic protective jacket. Try to store disks vertically. If disks are stored horizontally, do not stack more than ten disks.

Exposure

1-55. Floppy and CD-ROM disks are subject to exposure from magnetic fields, smoke, heat, and sunlight. X-rays may also have a negative effect. Care must be taken from exposure.

Magnetic Fields

1-56. Floppy and CD-ROM disks should never be exposed to anything that could be the source of a magnetic field. Exposure of a disk to a magnetic field could cause the destruction of some or all of the data contained on that disk. Some common sources of magnetic energy are CRTs, disk drives, and perhaps the most common, the telephone.

Smoke

1-57. Smoke can cause buildup on disks and on disk drives. DO NOT SMOKE while working at a terminal or computer.

Heat and Sunlight

1-58. Never expose floppy or CD-ROM disks to excessive heat or direct sunlight. These can cause the disks to become warped or distorted so they cannot be used. Disks will typically operate only between 10 and 50 degrees Celsius (50 to 120 degrees Fahrenheit). They will accept a relative humidity of 10 percent to 80 percent.

X-Rays

1-59. There is some question about the effect that airport X-ray machines have on disks. Walk-through X-ray machines at airports usually have no effect on floppy or CD-ROM disks. However, this is not to say there will be no effect. It is up to the person carrying the disks to take appropriate measures to protect the disks. Do not take chances if there is a possibility that the X-ray machines can affect the disks.

Labeling

1-60. When labeling the outside of a 3 1/2-inch floppy disk, write the label before attaching it to the disk. Never use a pencil or ballpoint pen to write on a label once that label has been attached to a disk. Using an instrument with a sharp point to write on the label can actually etch into the surface of the disk underneath the protective sheath, destroying that disk. Use a felt-tip marker if a label has already been attached to a disk. When labeling a CD-ROM disk, use a fine tip permanent marker. Never mark on the read/write side of the disk.

Data Backup

1-61. In almost all computer systems, the possibility exists for errors to occur that accidentally alter or destroy the data stored in the databases or files. This may occur because of natural disasters (such as fire, flood, or power outages). It may also occur through operator error or equipment malfunction. Therefore, it is essential to provide a means to ensure that any data lost can be recovered. The most common method is backup files. A backup file is merely a copy of a file. If for some reason the file or database is destroyed or becomes unusable, the backup file can be used to recreate the file or database. Floppy and CD-ROM disks are the most commonly used media for backup.

1-62. **Floppy Disk.** The most common method of creating a backup for a microcomputer is to use a floppy disk and the disk copy procedure. This is done by using the original database or file and copying the information onto a blank floppy disk. The instructions for this procedure will be provided with the particular computer and program in use.

1-63. **CD-ROM Disk.** Another method of creating a backup is to use CD-ROM disks. The information contained on a floppy disk or hard drive, whether it is a database or file, can be copied onto a CD-ROM disk.

SUMMARY

1-64. This chapter has presented information on the history and classification of computers. It introduced electronic digital computers and their uses and operation. The information that follows summarizes the important points of this chapter.

1-65. Early computers were MECHANICAL or ELECTROMECHANICAL. ELECTRONIC COMPUTERS came into use in the 1940s.

1-66. ANALOG COMPUTERS are special purpose computers designed to measure continuous electrical or physical conditions.

1-67. DIGITAL COMPUTERS are special or general purpose computers designed to perform arithmetic and logic functions on separate discrete data. They are generally used for business and scientific data processing.

1-68. Digital computers have evolved through four generations: vacuum tubes, transistors, miniaturized circuits, and ICs.

1-69. WORD PROCESSING is one of the most widespread uses of desktop PCs.

1-70. ACCOUNTING AND RECORDKEEPING are also major uses of computers. Included are order entry, inventory control, accounts receivable, accounts payable, general ledger, and payroll.

1-71. The Army's SHIPBOARD NON-TACTICAL ADP PROGRAM consists of computers used by work center supervisors for logistic and administrative support. This system expedites the storage and retrieval of information the Army has about its ships.

1-72. A DESKTOP (PERSONAL) COMPUTER is a microcomputer with at least a display screen, keyboard, mouse, and floppy disk drive. It may also have additional devices such as a CD-ROM drive and a zip drive.

1-73. An OPERATING SYSTEM is loaded into the computer to let the operator and other programs use the computer. It also provides common functions like read, write, and print.

1-74. A computer can be directed to run an APPLICATION PROGRAM by telling the operating system the name of program to run. Common application programs are word processing, accounting, and recordkeeping.

1-75. FLOPPY DISKS and CD-ROM disks are used for data storage and backup. To ensure that disks are not damaged, use care in handling, labeling, and storing the disks.

Check-on-Learning Questions

1. What are some of the ways that computers are classified?
2. Are mechanical computers classified as digital or analog?
3. The Army uses analog computers primarily for what purpose?
4. How do electromechanical computers differ from mechanical computers?
5. In electronic computers, transistors replaced vacuum tubes. What device subsequently replaced the transistors?
6. A computer that is designed to perform a specific operation and usually satisfies the needs of a particular type of problem, is said to be what type of computer?
7. Rather than using a stored program, a special purpose computer's applicability to a particular problem is a function of what?
8. What is a drawback to the special purpose computer?
9. How does a general purpose computer most differ from a special purpose computer?
10. How is a general purpose computer able to perform different operations?
11. In a general purpose computer, the ability to perform a wide variety of operations is achieved at the expense of what capabilities?
12. Are analog computers special purpose or general purpose computers?
13. What sorts of conditions are analog computers designed to measure?
14. Early analog computers were strictly mechanical or electromechanical devices. True or false?
15. What are computers called that combine the functions of both analog and digital computers?
16. Digital computers are generally used for what purposes?
17. What is the fundamental difference between analog and digital computers?
18. How is the accuracy of an analog computer restricted?
19. In an analog computer, a constant represented by a voltage can be read to what decimal place?
20. The accuracy of a digital computer is governed by what factor?
21. In a digital computer, what determines the number of decimal places in the constant?
22. You will most likely be working with what type of computer?
23. How many generations currently characterize the electronic computer world?
24. What does each generation of computer systems indicate?
25. Size-wise, how would computers of the first generation be characterized?
26. In first generation computers, internal processing functions were measured by what division of time?
27. What characterized the software used in first generation computers?

28. How were processing speed and reliability increased in second generation computers?
29. How was the storage capacity greatly increased in second generation computers?
30. With improvements in software, what kind of computer languages could be used on second generation computers?
31. What major benefit resulted from third generation computers?
32. In third generation computers, internal processing speeds were measured by what division of time?
33. The data cell had a storage capacity of how many characters?
34. What type of applications were most third generation computer systems designed to handle?
35. What type of computers are small and inexpensive yet provide a lot of computing power?
36. What will be one of the future challenges involving computer power?
37. What is one of the more widespread uses of the computer?
38. How many systems dealing with accounting applications have been widely accepted?
39. What is a central set of programs called that manages the execution of other programs and performs common functions like read, write, and print?
40. What is the function of a built-in program called a bootstrap loader?
41. What does it mean when the error message "NO SYSTEM" is displayed?
42. How should CD-ROM disks be handled?
43. If disks are stored horizontally, how many can be stacked?
44. What can exposure to a magnetic field do to the data on a disk?
45. What is the temperature range within which a disk will operate?
46. What is the purpose of creating backup files?
47. What is the most common method of creating backup files for a microcomputer?
48. Name another method used to create backup files.

Chapter 2

Hardware

INTRODUCTION

2-1. Components or tools of a computer system are grouped into one of two categories, hardware or software. Software will be covered in Chapter 3. Machines that make up the computer system are known as hardware. This hardware includes all the mechanical, electrical, electronic, and magnetic devices within the computer itself (the CPU). It also includes all related peripheral devices (printers, CD-ROM drives, zip drives, and so on). These devices will be covered in this chapter to show how they function and how they relate to one another. Figure 2-1 shows the functional units of a computer system (the inputs, the CPU, and the outputs). The inputs can be located on any storage medium from floppy disks, CD-ROM disks, zip disks, and so on. Inputs can also be entries from a console keyboard or a CRT terminal. The CPU will process the data from one or more of these inputs to produce output. The output may be contained in floppy disk, CD-ROM disk, zip disk; or it may be located in printed reports or information displayed on a console typewriter or CRT terminal. Figure 2-1 also shows the data flow, instruction flow, and flow of control. The discussion of hardware begins with the CPU, moves into storage media (magnetic disk, floppy disk, and CD-ROM), and ends with a discussion of input/output devices and how they work.

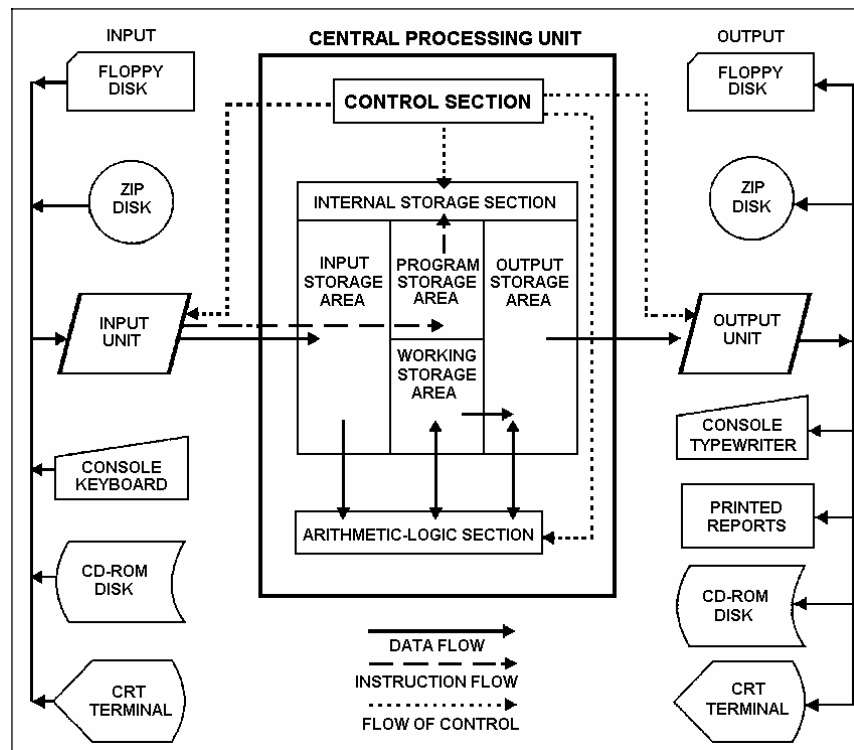


Figure 2-1. Functional Units of a Computer System

CENTRAL PROCESSING UNIT

2-2. The brain of a computer system (which is generally referred to as the mainframe) is the CPU. The CPU IS THE COMPUTER. The CPU processes the data transferred to it from one of the various input devices. The CPU then transfers either the intermediate or final results of the processing to one of many output devices. A central control section and work areas are required to perform calculations or manipulate data. The CPU is the computing center of the system. It consists of a control section, internal storage section (main or primary memory), and arithmetic-logic section (see Figure 2-1). Each of the sections within the CPU serves a specific function and has a particular relationship to the other sections within the CPU.

CONTROL SECTION

2-3. The control section may be compared to a telephone exchange because it uses the instructions contained in the program in much the same manner as the telephone exchange uses telephone numbers. When a telephone number is dialed, it causes the telephone exchange to energize certain switches and control lines to connect the dialing telephone with the telephone having the number dialed. In a similar manner, each programmed instruction, when executed, causes the control section to energize certain control lines, enabling the computer to perform the function or operation indicated by the instruction.

2-4. The program may be stored in the internal circuits of the computer (computer memory) or it may be read instruction-by-instruction from external media. The internally stored program type of computer, generally referred to only as a stored-program computer, is the most practical type to use when speed and fully automatic operation are desired.

2-5. Computer programs may be so complex that the number of instructions plus the parameters necessary for program execution will exceed the memory capacity of a stored-program computer. When this occurs, the program may be sectionalized (or broken down into modules). One or more modules are then stored in computer memory and the rest in an easily accessible auxiliary memory. As each module is executed producing the desired results, it is then swapped out of internal memory and the next succeeding module reads in.

2-6. In addition to the commands that tell the computer what to do, the control unit also controls how and when each specific operation is performed. It is also active in initiating circuits that locate any information stored within the computer or in an auxiliary storage device and in moving this information to the point where the actual manipulation or modification is accomplished.

- 2-7. The four major types of instructions are:
- **Transfer.** The basic function of transfer instructions is to transfer (move) data from one location to another.
 - **Arithmetic.** Arithmetic instructions combine two pieces of data to form a single piece of data using one of the arithmetic operations.
 - **Logic.** Logic instructions transform the digital computer into a system that is more than just a high-speed adding machine. Using logic instructions, the programmer may construct a program with any number of alternate sequences. For example, through the use of logic instructions, a computer being used for maintenance inventory will have one sequence to follow if the number of a given item on hand is greater than the order amount and another sequence if it is smaller. The choice of which sequence to use will be made by the control section under the influence of the logic instruction. Logic instructions provide the computer with the ability to make decisions based on the results of previously generated data. That is, the logic instructions permit the computer to select the proper program sequence to be executed from among the alternatives provided by the programmer.
 - **Control.** Control instructions send commands to devices not under direct command of the control section, such as input/output units or devices.

ARITHMETIC-LOGIC SECTION

2-8. The arithmetic-logic section performs all arithmetic operations (adding, subtracting, multiplying, and dividing). Through its logic capability, it tests various conditions encountered during processing and takes action based on the result. As indicated by the solid arrows in Figure 2-1, data flows between the arithmetic-logic section and the internal storage section during processing. Specifically, data is transferred as needed from the internal storage section to the arithmetic-logic section, processed, and returned to the internal storage section. At no time does processing take place in the storage section. Data may be transferred back and forth between these two sections several times before processing is completed. The results are then transferred from internal storage to an output unit as indicated by the solid arrow (see Figure 2-1).

MEMORY (INTERNAL STORAGE) SECTION

2-9. All memory (internal storage) sections must contain facilities to store computer data or instructions (that are intelligible to the computer) until these instructions or data are needed in the performance of the computer calculations. Before the stored computer program can begin to process input data, it is first necessary to store in its memory a sequence of instructions, tables of constants, and other data it will use in its computations. The process by which these instructions and data are read into the computer is called loading.

2-10. The first step in loading instructions and data into a computer is to manually place enough instructions into memory. Load the instructions by using the keyboard or electronically using an operating system (discussed in chapter 1). The instructions can be used to bring in more instructions as desired. In this manner, a few instructions are used to bootstrap more instructions. Some computers make use of an auxiliary (wired) memory that permanently stores the bootstrap program, thereby making manual loading unnecessary.

2-11. The memory (internal storage) section of a computer is essentially an electronically operated file cabinet. It has a large number (usually several hundred thousand) of storage locations, each referred to as a storage address or register. Every item of data and program instruction read into the computer during the loading process is stored or filed in a specific storage address and is almost instantly accessible.

TYPES OF INTERNAL STORAGE

2-12. Remember that the internal storage section is the holding area where instructions and data are kept. For the control section to control and coordinate all processing activity, it must be able to locate each instruction and data item in storage. To understand how the control section is able to find these instructions and data items, look at storage as nothing more than a collection of mailboxes. Each mailbox has a unique address and represents a location in memory (see Figure 2-2). Like the mail in a mailbox, the contents of a storage location can change, but the number on a mailbox or memory address always remains the same. In this manner, a particular program instruction or data item that is held in storage can be located by knowing its address. Some computers can address each character of data in memory directly. Others address computer words that each contains a group of characters at a single address. Some of the more common types of internal storage media used in today's computers are magnetic core and semiconductor.

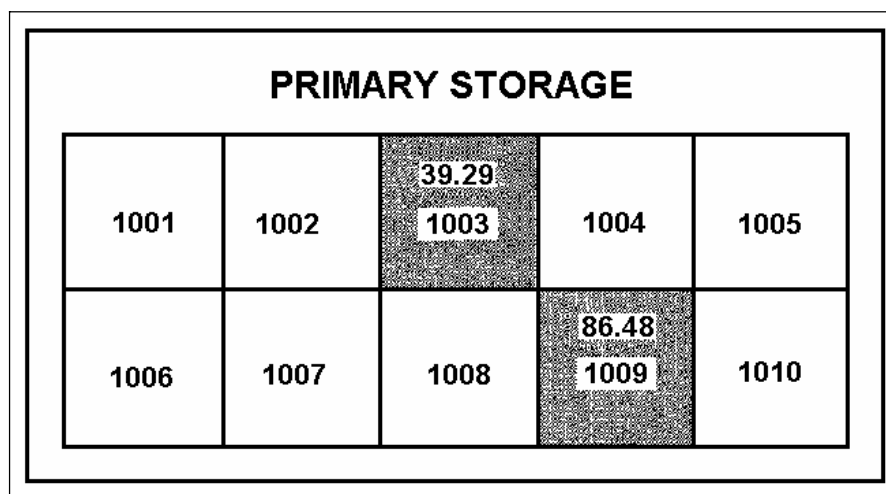


Figure 2-2. Memory Locations

MAGNETIC CORE STORAGE

2-13. Although magnetic core storage is no longer as popular as it once was, it will be covered in some detail because its concepts are easily understood and apply generally to the more integrated semiconductor and bubble-type memories. Magnetic core storage is made up of tiny doughnut-shaped rings made of ferrite (iron) that are strung on a grid of very thin wires (see Figure 2-3). Since data in computers is stored in binary form, a two-state device is needed to represent the two binary digits (bits), 0 for OFF and 1 for ON. In core storage, each ferrite ring (depending on its magnetic state) can represent a 0 or 1 bit. If magnetized in one direction, it represents a 1 bit, and if magnetized in the opposite direction, it represents a 0 bit. Sending an electric current through the wires on which the core is strung magnetizes these cores. It is this direction of current that determines the state of each core.

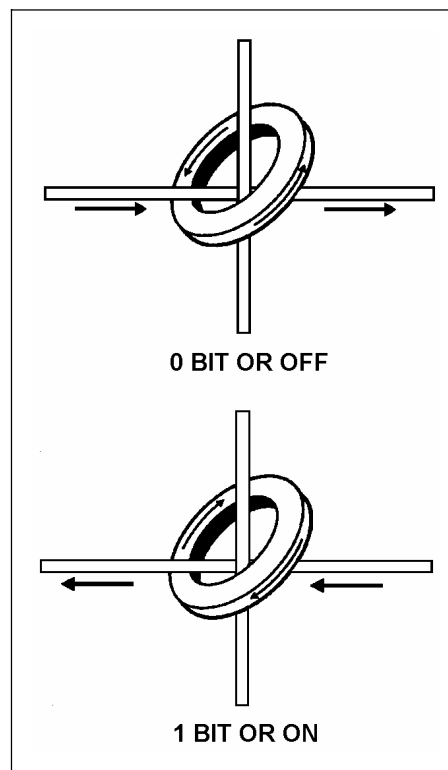


Figure 2-3. Two-State Principle of Magnetic Storage

SEMICONDUCTOR STORAGE (THE SILICON CHIP)

2-14. Semiconductor memory consists of hundreds of thousands of tiny electronic circuits etched on a silicon chip (see Figure 2-4). Each of these electronic circuits is called a bit cell and can be in either an OFF or ON state to represent a 0 or 1 bit, depending on whether or not current is flowing through that cell. Another name used for semiconductor memory chips is the ICs. Developments in technology have led to large-scale integration (LSI), which means that more and more circuits can be squeezed onto the same silicon chip. Companies are even manufacturing very large-scale integration (VLSI), which means even further miniaturization.

2-15. Some of the advantages of semiconductor storage are fast internal processing speeds, high reliability, low power consumption, high density (many circuits), and low cost. However, there is a drawback to this type of storage. It is volatile, which means all data in memory is lost when the power supply is removed. Should the power on a computer fail and there is no backup power supply, all the stored data is lost. This is not the case with magnetic core storage. Core storage is nonvolatile. This means the data is retained even if there is a power failure or breakdown, since the cores store data in the form of magnetic charges rather than electric current.

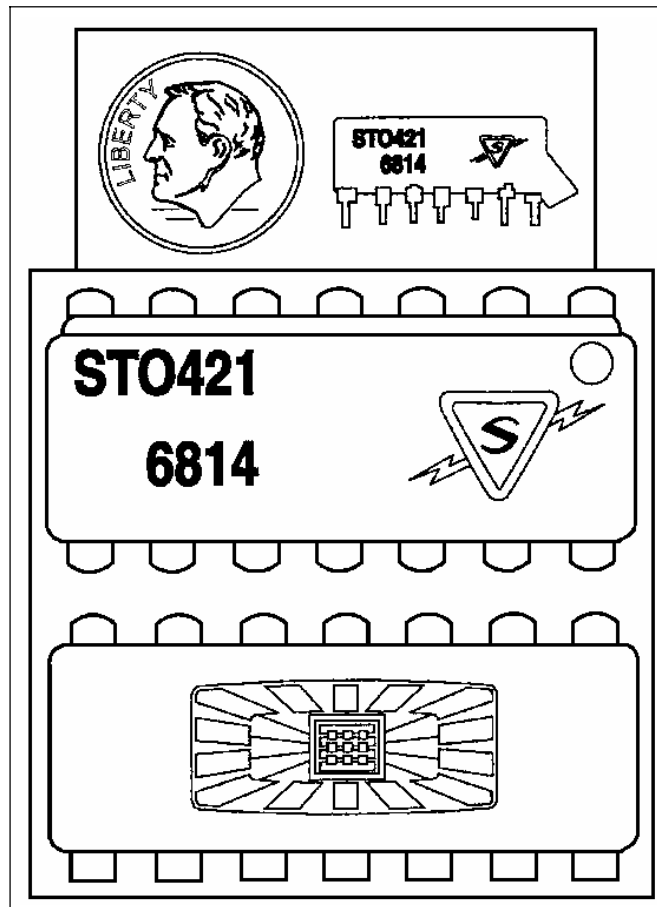


Figure 2-4. A Semiconductor Memory Chip (Integrated Circuit)

BUBBLE STORAGE

2-16. Another technological development, in 1981, in storage media was the introduction of bubble memory (see Figure 2-5). Bubble memory consisted of a very thin crystal made of semiconductor material. The molecules of this special crystal acted as tiny magnets. The polarity of these molecules or "magnetic domains" could be switched in an opposite direction by passing a current through a control circuit imprinted on top of the crystal. In this manner, data could then be stored by changing the polarity of the magnetic domains. The name bubble memory comes from viewing the magnetic domains (they look like tiny bubbles) under a microscope.

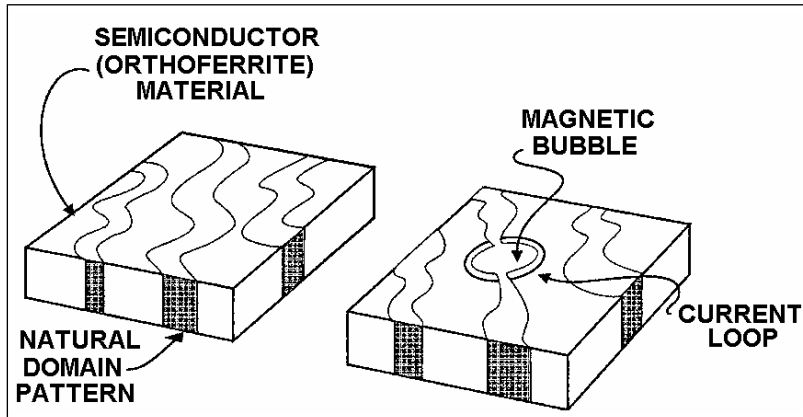


Figure 2-5. Bubble Memory

2-17. Since the principle is the same as for magnetic core storage, bubble memory is considered nonvolatile. The data is retained even if there is a power failure. The process of reading from bubble memory is nondestructive, meaning that the data is still present after being read. This is not the case with core storage, which must be regenerated after being read. Bubble memory became obsolete within five years when battery backup Complementary Metal Oxide Semiconductor-Random Access Memory (CMOS-RAM) became affordable.

CLASSIFICATIONS OF INTERNAL STORAGE

2-18. There is another way to classify internal (primary or main) storage. This method of classification is based on the following different kinds of memories used within the CPU:

- Read-only memory.
- Random-access memory.
- Programmable read-only memory.
- Erasable programmable read-only memory.

READ-ONLY MEMORY

2-19. In most computers, it is useful to have often used instructions, such as those used to bootstrap (initial system load) the computer or other specialized programs, permanently stored inside the computer. Memory that enables us to do this without the programs and data being lost (even when the computer is powered down) is called read-only memory. The computer manufacturer provides these programs in read-only memory (ROM). However, they can be updated to meet hardware requirements. Many complex functions such as routines to extract square roots, translators for programming languages, and operating systems can be placed in ROM. Since these instructions are hardwired (permanent), they can be performed quickly and accurately. Another advantage of ROM is that a computer facility can order programs tailored for its needs and have them permanently installed in

ROM by the manufacturer. Such programs are called microprograms or firmware.

RANDOM-ACCESS MEMORY

2-20. Another kind of memory used inside computers is called random-access memory or read/write memory. Random-access memory (RAM) is like a blackboard on which notes can be scribbled, read, and then erased when finished reading them. In the computer, RAM is the working memory. Data can be read (retrieved) from or written (stored) into RAM just by giving the computer the address of the location where the data is stored or will be stored. When the data is no longer needed, it is simply written over. This allows the storage to be used again for something else. Core, semiconductor, and bubble storage all have random access capabilities.

PROGRAMMABLE READ-ONLY MEMORY

2-21. An alternative to ROM is programmable read-only memory (PROM). PROM can be purchased already programmed by the manufacturer or in a blank state. By using a blank PROM, any program can be entered into the memory. However, once the PROM has been written into, it can never be altered or changed. Therefore, along with the advantage of ROM, there is the additional flexibility to program the memory to meet a unique need. The main disadvantage of PROM is that if a mistake is made and entered into PROM, it cannot be corrected or erased. A special device is needed to "burn" the program into PROM.

ERASABLE PROGRAMMABLE READ-ONLY MEMORY

2-22. The erasable programmable read-only memory (EPROM) was developed to overcome the drawback of PROM. EPROMs can also be purchased blank from the manufacturer or programmed locally. However, this requires special equipment. The big difference with EPROM is that it can be erased if and when the need arises. Data and programs can be retrieved over and over again without destroying the contents of the EPROM. They will safely stay there until the EPROM is reprogrammed by first erasing it with a burst of ultra-violet light. This is an advantage because if a mistake is made while programming the EPROM, it is not considered fatal. The EPROM can be erased and corrected. It also allows the flexibility to change programs to include improvements or modifications in the future.

SECONDARY STORAGE

2-23. The last kind of memory (which will be briefly introduced) is called secondary storage or auxiliary storage. This is memory outside the main body of the computer (CPU) where programs and data are stored for future use. When the computer is ready to use these programs and data, they are read into internal storage. Secondary (auxiliary) storage media extends the storage capabilities of the computer system. Secondary storage is needed for the following two reasons:

- Since the computer's internal storage is limited in size, it cannot always hold all the data that is needed.

- Data and programs do not disappear when power is turned off. Secondary storage is nonvolatile. This means information is lost only if it is intentionally erased by the user.

The three types of secondary storage most commonly used are 3 1/2-inch floppy disks, CD-ROM disks, and zip disks.

MAGNETIC DISK STORAGE

2-24. Disk storage devices are popular largely because of their direct-access capabilities. Most every system (micro, mini, and mainframe) will have disk capability. Magnetic disks resemble round platters that are coated with a magnetizable recording material (iron oxide). However, their similarities end there. Magnetic disks come in many different sizes and storage capacities. They range from 3 inches to 4 feet in diameter and can store from 1.44 megabytes (MBs) on a 3 1/2-inch floppy disk to 250 MB on a zip disk. They can be portable in that they are removable or they can be permanently mounted in the storage devices called disk drive units or disk drives. They can be made of rigid metal (hard disks) or flexible plastic (floppy disks or diskettes).

2-25. Data is stored on all disks in a number of invisible concentric circles called tracks. Each track has a designated number beginning with track 000 at the outer edge of the disk. The numbering continues sequentially toward the center to track 199, 800, or whatever is the highest track number. No track ever touches another (see Figure 2-6). The number of tracks can vary from 35 to 77 on a floppy disk surface and from 200 to over 800 on hard disk surfaces.

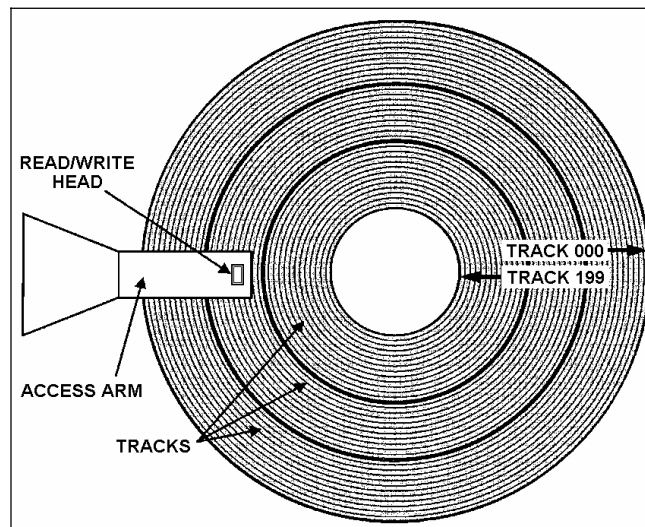


Figure 2-6. Location of Tracks on the Disk's Recording Surface

2-26. Data is written as tiny magnetic bits (or spots) on the disk surface. Eight-bit codes are generally used to represent data. Each code represents a different number, letter, or special character. Chapter 4 will explain how the codes are formed. When data is read from the disk, the data on the disk remains unchanged. When data is written on the disk, it replaces any data previously stored on the same area of the disk.

2-27. Characters are stored on a single track as strings of magnetized bits (0s and 1s) (see Figure 2-7). The 1 bits indicate magnetized spots or ON bits. The 0 bits represent unmagnetized portions of the track or OFF bits. Although the tracks get smaller as they get closer to the center of the disk platter, each track can hold the same amount of data because the data density is greater on tracks near the center.

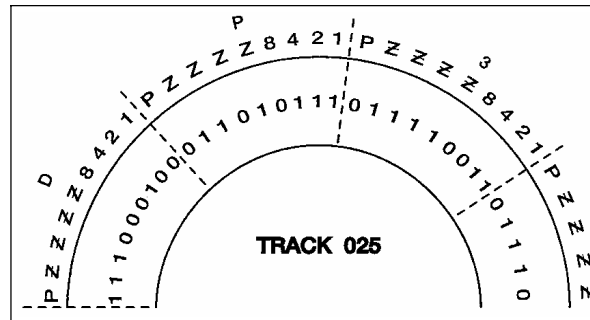


Figure 2-7. A String of Bits Written to Disk on a Single Track

2-28. A track can hold one or more records. A record is a set of related data treated as a unit. The records on a track are separated by gaps in which no data is recorded, and each of the records is preceded by a disk address. This address indicates the unique position of the record on the track and is used to directly access the record. Figure 2-8 shows a track on which five records have been recorded. Because of the gaps and addresses, the amount of data that can be stored on a track is reduced as the number of records per track is increased. Records on disk can be blocked (grouped together). Only one disk address is needed per block, and as a result, fewer gaps occur. The blocking technique can be used to increase the amount of data that can be stored on one track.

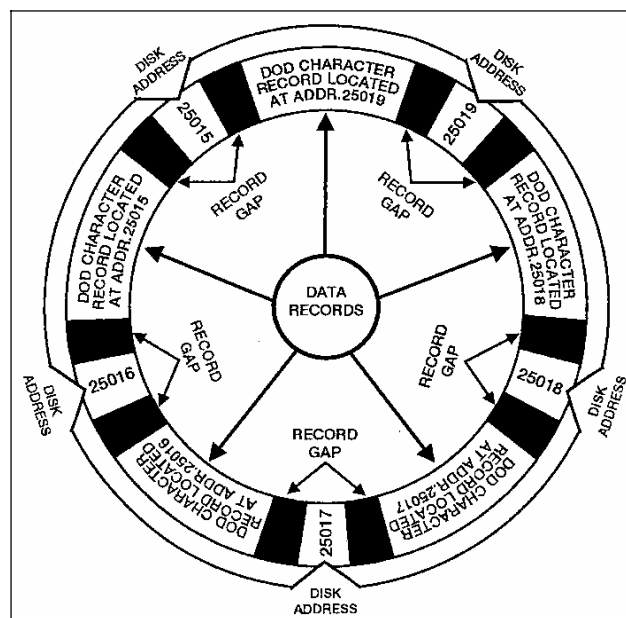


Figure 2-8. Data Records as They are Written to Disk on a Single Track

2-29. The storage capacity of a disk depends on the bits per inch of track and the tracks per inch of surface. Using Winchester technology, the designers of disk drive units were able to increase the data density of a disk by increasing the number of tracks. Winchester was the code name used by International Business Machines Corporation (IBM) during the development of this technology. The designers originally planned to use dual disk drives to introduce the new concept. Each drive was to have a storage capacity of 30 million characters, and therefore was expected to be a "30-30." Since that was the caliber of a famous rifle, the new product was nicknamed "Winchester." The designers found that data density could be improved and storage capacity increased by reducing the flying height. Flying height is the distance of the read/write heads over the disk surfaces when reading and writing. By doing this, smaller magnetized spots could be precisely written and then read. The read/write heads were moved so close to the disk that a human hair looked like a mountain in the path of the flying head. Winchester technology reduced this potential problem by sealing the disks in a contamination-free container. This eliminated foreign objects from coming in contact with the read/write heads.

2-30. Depending on the manufacturer and the model of disk drive in use, data can be physically organized in one of two ways on a disk pack. One way uses the cylinder method and the other uses the sector method. On diskettes, data is organized using the sector method.

Cylinder Method

2-31. This method uses a cylinder as the basic reference point. Figure 2-9 shows a disk pack containing six disk platters with 10 recording surfaces. Looking down through the disk pack from the top will show that all the tracks with the same number line up vertically. Together they are called a cylinder. These 10 tracks, one on each recording surface, can be referenced by the 10 read/write heads on the five access arms at each discrete location where the access arms can be positioned. To physically reference a record stored using the cylinder method, a computer program must specify the cylinder number, the recording surface number, and the record number. In Figure 2-9 the record is stored in cylinder 25 of recording surface 6 and is the first record on that track. Special data stored on each track specifies the beginning of the track so that the first record, second record, third record, and so on, can be identified.

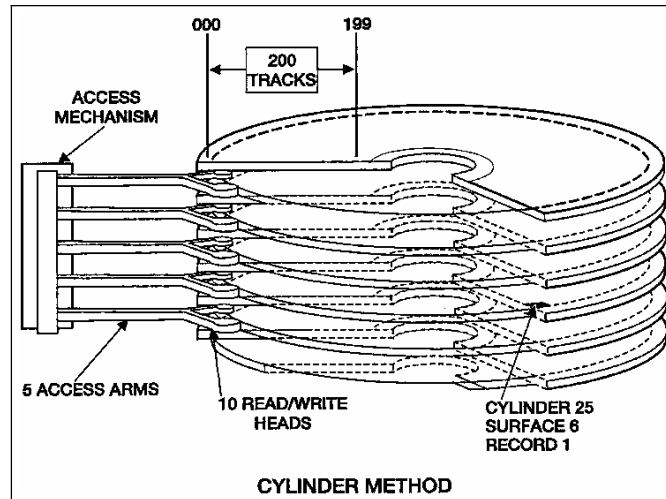


Figure 2-9. Physical Organization of Data on a Disk (Cylinder Method)

Sector Method

2-32. Another way to physically organize data on the disk pack (and on diskettes) is to use the sector method. This requires that each of the tracks be divided into individual storage areas called sectors (see Figure 2-10). The number of sectors varies with the disk system used; however, there are usually eight or more. Each sector holds a specific number of characters. Before a record can be accessed, a computer program must again give the disk drive the record's address specifying the track number, the surface number, and the sector number of the record. One or more read/write heads are then moved to the proper track, the head over the specified surface is activated, and the data is read from or written to the designated sector as it spins under the head.

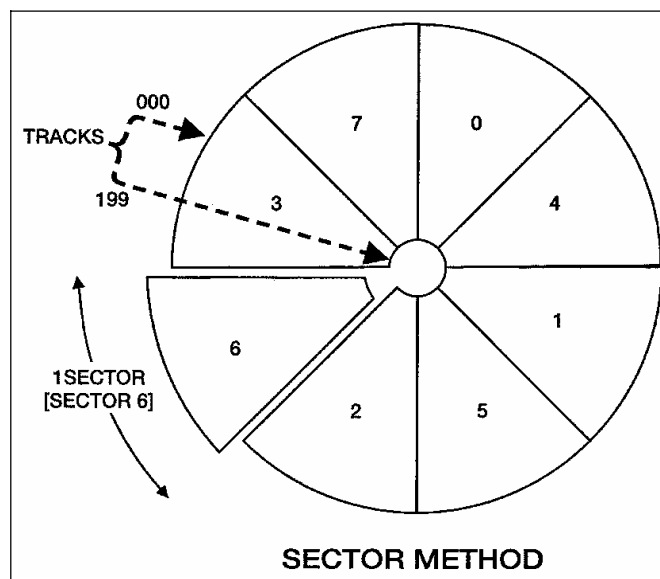


Figure 2-10. Physical Organization of Data on a Disk (Sector Method)

INPUT/OUTPUT DEVICES (EXTERNAL)

2-33. Input and output devices are similar in operation but perform opposite functions. It is through the use of these devices that the computer is able to communicate with the outside world. Input data may exist in any one of the following three forms:

- Manual inputs from a keyboard, mouse, light pen, and touchpad.
- Analog inputs from instruments or sensors.
- Inputs from a source on or in which data has previously been stored in a form intelligible to the computer.

2-34. Computers can process hundreds of thousands of computer words or characters per second. A study of the first method (manual input) reflects the inability of human-operated keyboards or keypunches to supply data at a speed that matches the speed of digital computers. A high average speed for keyboard operation is two or three characters per second. When coded to form computer words, this would reduce the data input rate to the computer to less than a computer word per second. Since mainframe computers are capable of reading several thousand times this amount of information per second, it is clear that manual inputs should be reduced to make more efficient use of computer time. However, as a rule, the keyboard is the normal input media for microcomputers.

2-35. Input data that has previously been recorded on magnetic disks, floppy disks, CD-ROM disks, or zip disks in a form understood by the program may also be entered into the computer. These are much faster methods than entering data manually from a keyboard. The most commonly used input devices in this category are a magnetic disk drive unit, a floppy disk drive unit, CD-ROM drive, and zip drive.

2-36. Output information is also made available in three forms:

- Displayed information (codes, numbers, words, or symbols presented on a display device like a cathode-ray screen).
- Control signals (information that operates a control device; such as a lever, aileron, or actuator).
- Recordings (information that is stored in a machine language or human language on disks [floppy, CD-ROM, and zip] or printed media).

Devices that display, store, or read information includes magnetic tape units, magnetic disk drive units, floppy disk drive units, printers, and display devices.

MAGNETIC DISK DRIVE UNITS (INPUT/OUTPUT)

2-37. Magnetic disk drive units are storage devices that read and write information on the magnetized surfaces of rotating disks. The disks are made of thin metal, coated on each side so that data can be recorded in the form of magnetized spots. As the disks spin, characters can be stored on them or retrieved in a direct manner. This direct accessing of data has a big advantage over the sequential accessing of data. It gives us fast, immediate

access to specific data without having to examine each and every record from the beginning. The disk drive can also be directed to begin reading at any point.

2-38. Located within each disk drive unit is a drive motor that rotates the disk at a constant speed, normally 3,600 revolutions per minute (rpm) (or 60 revolutions per second [rps]). The rotational speed for a floppy disk (because of its plastic base) is usually between 300 and 400 rpm. Data is written on the tracks of a spinning disk surface and read from the surface by one or more (multiple) read/write heads. When reading from and writing to hard disks (rigid disks), the read/write heads float on a cushion of air and do not actually touch the surface of the disk. The distance between the head and the surface varies from a millionth of an inch to one-half millionth of an inch. This distance is called the flying height. When multiple disks (platters) are packaged together as a unit in a disk pack, a number of access arms and read/write heads are used to access both surfaces of each platter (see Figure 2-11). The disk pack shown consists of six metal disks mounted on a central spindle. Data can be recorded on all surfaces except the top surface of the top disk and the bottom surface of the bottom disk. These two surfaces are intentionally left blank for protection.

FLOPPY DISK DRIVE UNITS (INPUT/OUTPUT)

2-39. Floppy disk drive units are physically smaller than magnetic disk drive units and are typically used with desktop PCs (see Figure 2-12). The unit consists of a disk drive in which the disk rotates and a controller containing the electronic circuitry that feeds signals onto and from the disk. The 3 1/2-inch floppy is the most common size used. The disk (diskette) is made of a thin, hard plastic shell. The platter (floppy disk) located inside the shell, is coated with magnetic material so characters can be recorded on the surface in the form of magnetized spots.

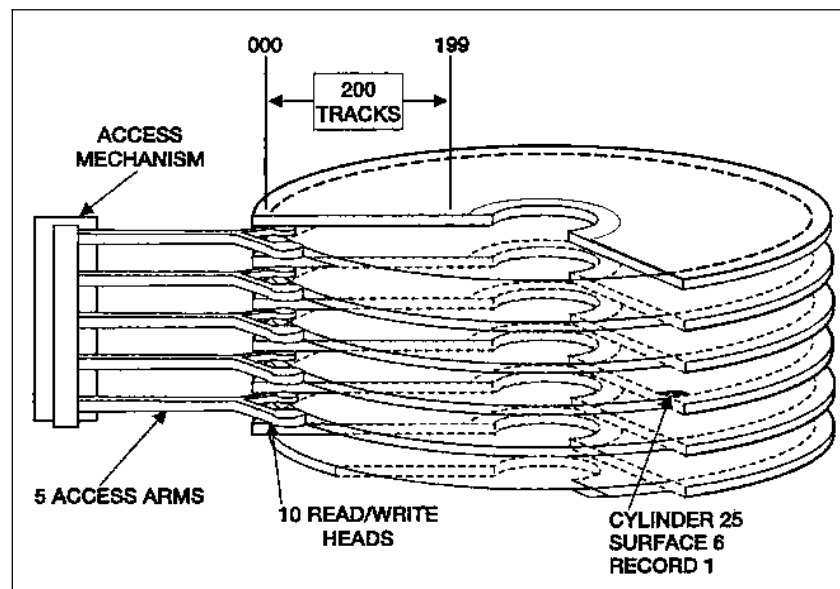


Figure 2-11. Multiple Access Arms and Read/Write Heads Used With Disk Packs

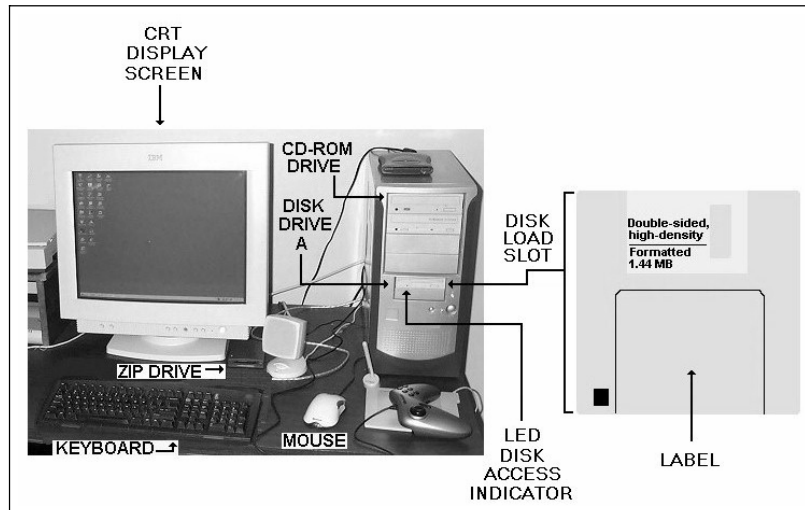


Figure 2-12. Floppy Disk Drive Unit

PRINTERS (OUTPUT)

2-40. Printers (see Figure 2-13) are widely used output devices that express coded characters on hard (paper document) copy. They print out computer program results as numbers, letters, words, symbols, graphics, or drawings. Printers range from electronic typewriters to high-speed printers. High-speed printers are usually used on mainframes and minis to prepare supply requisitions, paychecks, inventory, or financial reports at ten lines per second and faster. The types of printers that will be discussed are the daisy-wheel, dot matrix, ink jet, and laser. These are the ones commonly used with PCs.

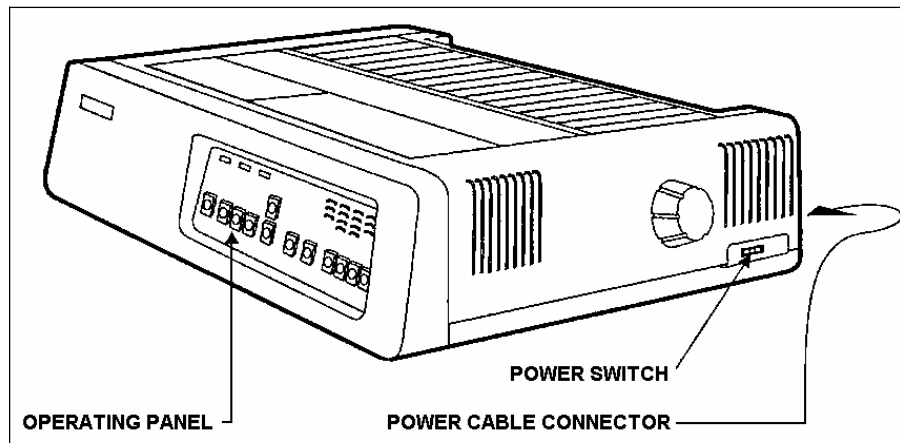


Figure 2-13. Printer

DAISY-WHEEL PRINTERS

2-41. Daisy-wheel printers have the most professional-looking, pleasing-to-the-eye print of all the printers in the character-at-a-time impact printer class. Daisy-wheel printers were often used in an office or word processing environment. However, they have been replaced by ink jet and laser printers. The daisy-wheel printer uses a round disk, with embossed characters located

at the end of each petal-like projection (one character per petal), similar to the petals of a daisy. A drive motor spins the wheel at a high rate of speed. When the desired character spins to the correct position, the print hammer strikes that character, causing it to be printed on the paper. Once the character is printed, the daisy wheel continues to move, searching out the next character to be printed, until the line is completed. The speeds of daisy-wheel printers range from 30 to 60 characters per second (cps).

DOT-MATRIX PRINTERS

2-42. Dot-matrix printers create characters in much the same way numbers on the scoreboard at a football game are seen. In contrast to daisy-wheel printers, dot-matrix printers use arrangements of tiny pins or hammers, called a dot matrix, to generate characters a dot at a time. A dot-matrix print head builds characters out of the dots created by the pins in the matrix. Figure 2-14, view A, shows what dot-matrix characters look like printed.

2-43. The dot matrix is defined in terms of rows and columns of dots. A 5 by 7 matrix uses up to five vertical columns of seven dots to create a character. Figure 2-14, view D, shows an example of a 5 by 7 matrix printing the letter H. The size of dot matrixes varies from a 5 by 7 matrix to as large as a 58 by 18 matrix. A number of dot-matrix printers use a single vertical column of pins to print characters (see Figure 2-14, view B). The characters are printed by moving (stepping) the print head a small amount and printing the vertical columns one at a time until the character is printed (see Figure 2-14, views C and D).

2-44. The size of the matrix determines the quality of the printed character. In other words, the more dots used to print a character, the better the character is filled in and the higher its print quality. Dot-matrix printers are faster than the daisy-wheel printers with speeds ranging from 60 to 350 cps, but their print quality is not as good.

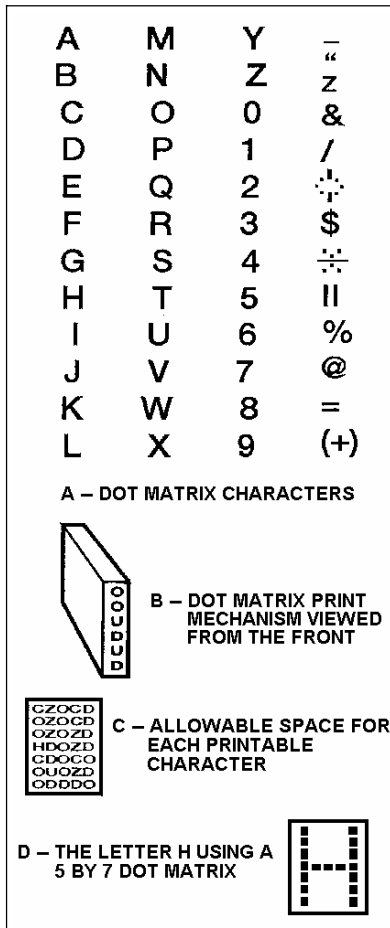


Figure 2-14. Dot-Matrix Printing

INK JET PRINTERS

2-45. Ink jet printers use a technique very similar to the way a can of spray paint and a stencil are used. A spray of electrically charged ink is shot (under pressure) toward the paper. Before reaching the paper, the ink is passed through an electrical field that creates the letters in a matrix form. The print resulting from this process consists of easy-to-read, high-quality characters. Some manufacturers use large droplets of ink for faster printing, while others use small droplets for better clarity but with slightly reduced printing speeds. These types of printers can print up to 300 cps.

LASER PRINTERS

2-46. Laser printers direct a beam of light through a rotating disk containing the full range of print characters. The appropriate character image is directed onto photographic paper, which is then put through a toner, developed, and used to make additional copies. The print resulting from this process consists of sharp, clean images that are easy on the eyes. These types of printers can print up to 20,000-plus lines per minute, or 26,666 cps.

KEYBOARDS (INPUT)

2-47. A keyboard is nothing more than an array of switches called keyswitches. Keyboards are designed to input a code to the computer when a keyswitch is depressed. Each key on the keyboard is assigned a particular code value and is usually imprinted with a legend to identify its function. Figure 2-15 shows a keyboard combined with a CRT on a microcomputer.

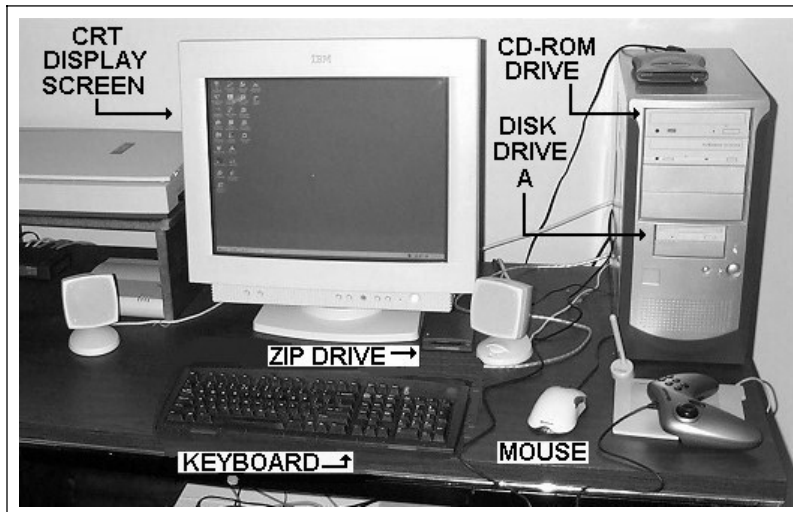


Figure 2-15. Keyboard Combined With a CRT and Microcomputer

2-48. The primary purpose of a keyboard is to enter or input alphanumeric (numbers, letters, and special characters) character codes. The major grouping of keyswitches on a keyboard will be in one of the two styles of a typewriter keyboard arrangement (QWERTY or DVORAK). The typewriter keyswitches are arranged in four rows of 10 or more switches. The keyboard arrangement shown in Figure 2-16 is called QWERTY. The rows are usually offset to the row above to make it easier to reach all the keys when typing. The tops of the individual keyswitches are sculptured to conform to the shape of the human finger.

2-49. Other groupings of keyswitches are used for special purposes, such as number entry (calculator) keypads, special function switches (F1-F12), and cursor control keys (to move to different locations on the screen use the cursor control key). The special function switches allow an operator to use the special functions designed in the software. For example, in a word processing program, special function switches to do the following:

- Check the spelling of a document.
- Search for a particular portion of text.
- Move text from one place to another.
- Print hard copies of a document.

These are just a few of the functions allowed. The more familiar an operator gets with a computer, the more functions they will learn.

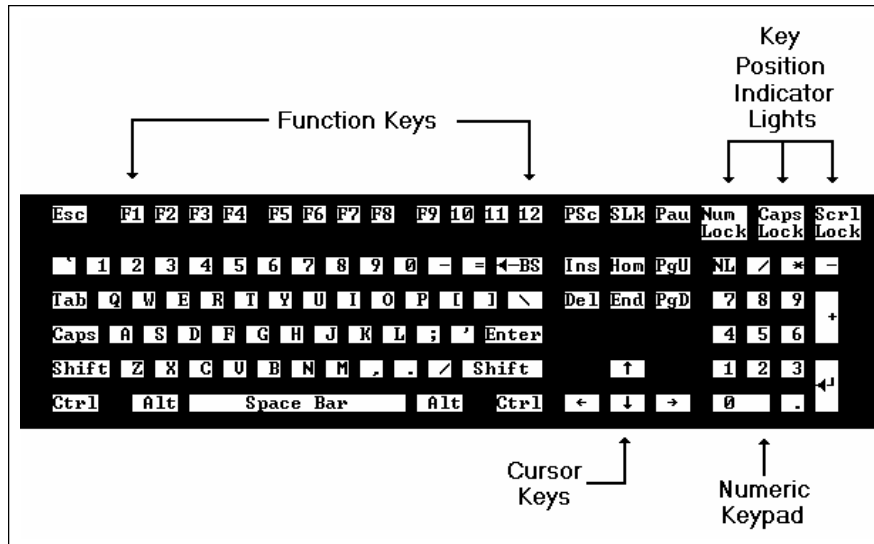


Figure 2-16. Keyboard Layout

2-50. Keyboard designs vary from device to device. Keyboards are dependent on the requirements of the system on which they are installed. Keyboards are generally used with nontactical computer systems. However, the newer tactical display system consoles have optional keyboards for data entry. A keyboard may be built into the display device or it may be a separate component connected only by a communication cable.

DISPLAY DEVICES

2-51. Display devices are the CRTs and other displays that are part of computer terminals, computer consoles, and microcomputers. They are designed to project, show, exhibit, or display soft copy information (alphanumeric or graphic symbology).

2-52. The information displayed on a display device screen is not permanent. That is where the term “soft copy” comes from. The information is available for viewing only as long as it is on the display screen. Two types of display devices used with PCs are the raster scan CRTs and the flat panel displays.

RASTER SCAN CATHODE-RAY TUBE

2-53. Raster scan CRTs (television [TV] scan video monitors or display monitors) are used extensively in the display of alphanumeric data and graphics. They are used primarily in nontactical display applications such as user terminals and desktop PCs.

2-54. The raster is a series of horizontal lines crossing the face of the CRT screen. Each horizontal line is made up of one trace of the electron beam from left to right. The raster starts at the top left corner of the CRT screen. As each horizontal line is completed, the blanked electron beam is rapidly returned or retraced to the left of the screen.

2-55. Vertical deflection moves the beam down and the horizontal sweep repeats. When the vertical sweep reaches the bottom line of the raster, a vertical blanked retrace returns the sweep to the starting position of the raster, and the process is repeated.

2-56. Each completed raster scan is referred to as a field (two fields make up a frame). The display rate of fields and frames determines the amount of flicker in the display that is perceived by the human eye. Each field is made up of approximately 525 horizontal lines. The actual number of horizontal lines varies from device to device. A frame consists of the interlaced lines of two fields. The horizontal lines of the two fields are interlaced to smooth out the display. A display rate of 30 frames per second produces a smooth, flicker-free raster and corresponding display on the screen.

PICTURE ELEMENTS

2-57. The actual display of data results from the use of picture elements. A picture element is a variable dot of light derived from video signals input to the display monitor. The picture elements, often called pixels or pels, are contained in the horizontal scan lines crossing the face of the CRT screen. The horizontal and vertical sweeps are continuous and repetitive in nature.

2-58. Pictures with alphanumeric characters and graphics can be created and displayed by varying the intensity or brightness of the picture element dots. This is done in conjunction with the phosphor coating on the face of the CRT.

2-59. The number of picture elements in each horizontal line varies from device to device. The actual number of picture elements is dependent on the following:

- Frequency bandwidth of the video monitor.
- The number of characters to be displayed on a line.
- The physical size of the screen.

2-60. Each picture element is addressable by a row and column address. Picture elements are numbered from left to right on each horizontal line (column number). Each horizontal line has a row number. Picture elements, at a minimum, will have OFF (blanked) or ON (full intensity) states. Many display devices have the capability to display picture elements at varying degrees of intensity for the display of graphics.

2-61. Characters are assembled on the screen in much the same way as a dot-matrix print head prints a character. It takes several horizontal lines and picture elements on each line to create a character. Figure 2-17 shows the generation of the character "A," seven picture elements wide and nine horizontal lines high. The character is built using what is, in effect, a 7 by 9 dot matrix. The picture elements used to build the character would be at full intensity; the remaining picture elements in the matrix would be blanked. If light characters on a darkened screen were desired, then the character picture elements would be blanked and the remainder displayed at full intensity.

2-62. Approximately 640 picture elements per horizontal line are required for the display of an 80-character line. Therefore, there will be 140,000 picture elements on a raster scan display screen (80 alphanumeric characters per line and 25 lines).

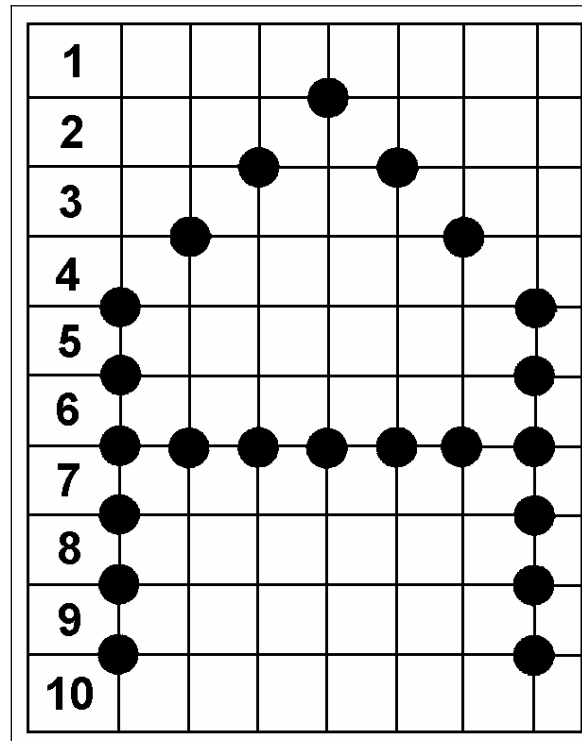


Figure 2-17. A 7 by 9 Picture Element Character

HORIZONTAL AND VERTICAL RESOLUTION

2-63. Horizontal resolution is defined in terms of the number of picture elements that can be displayed on the horizontal line without overlapping or running into each other. It is often stated in terms of lines of resolution. In other words, a monitor with a horizontal resolution of 1,000 lines can display 1,000 vertical lines using 1,000 picture elements per line.

2-64. Vertical resolution depends on the number of horizontal scan lines used by the particular display raster. Generally, the greater the number of scanned lines, the easier it is to resolve a horizontal line of display. This characteristic remains true up to a point, called the merge point, where the human eye cannot detect the variation between the lines.

DISPLAYING DATA ON RASTER SCAN SCREENS

2-65. Raster scan displays are repetitive in nature. The raster frame is displayed approximately 30 times a second.

2-66. The basic video monitor does nothing more than display the video signals it receives. If no video signals are received, then all the picture elements remain blanked, and the screen is blank in each frame. For data to

be displayed accurately, each and every frame must blank and unblank the same picture elements.

2-67. The digital logic that drives video monitors is designed to take advantage of the repetitive nature of frames. There can only be a fixed number of picture elements on the screen of a display. Therefore, the contents of the display screen are organized into a data unit called a page.

2-68. The page contains the status of every picture element on the display screen. The page is usually stored in some form of random-access memory, RAM chips being the most common. The contents of page memory (or as it is sometimes called, video memory) are continually scanned by the video generation logic and used to develop the video signals for the picture element display. The picture element locations in page memory are read in time to develop the video signals for the picture element display on the horizontal lines.

2-69. If the display is to be changed, the contents of page memory must be changed. The display on the screen changes as new data is stored in page memory.

UNFORMATTED DISPLAYS

2-70. Displays that reference page memory by picture element address, are called unformatted or fully populated displays. These displays are more commonly used for graphics rather than alphanumeric characters.

FORMATTED DISPLAYS

2-71. Displays are often organized by character position and line number. These displays are known as formatted displays. This display method is used with devices displaying alphanumeric characters only or those with an alternate graphic capability.

2-72. The video generation logic of these types of displays scans the entire page memory, as before, to generate the display picture elements. The difference is in the way the new data is written into the page memory. Individual picture element addresses are not used. Character addresses are used to reference page memory.

2-73. The screen is organized into character lines. Each line is made up of a fixed number of character positions or columns. A fixed number of character lines can be displayed. A common arrangement found on display screens is twenty-five 80-character lines or 2,000 characters.

2-74. The character set that can be displayed on a device's formatted screen is stored in ROMs or PROMs. That is, the dot-matrix (picture element) patterns for each individual character to be displayed are stored. Different character sets may be displayed by simply replacing the appropriate ROM or PROM chips with new chips containing different character patterns.

2-75. Upon receipt of a character code and a row and column address, the device logic reads the picture element pattern (dot matrix) from the ROM and writes the pattern into the appropriate character position in the page memory. The desired character is then displayed at the correct position. Other display devices store the codes in page memory and convert the codes to picture element dots when scanning memory to refresh or redisplay the characters on the screen. The use of formatted displays greatly simplifies the programming requirements for the display of alphanumeric data.

FLAT PANEL DISPLAYS

2-76. A number of display methods are in use and are designed to reduce the depth of the CRT display caused by the length of the tube. These devices are collectively known as flat panel displays. Three types of flat panel displays commonly in use with computer systems are liquid crystal displays (LCDs), gas plasma displays (GPDs), and electroluminescent displays (ELDs). The screens of these flat panel displays are made up of pairs of electrodes. Each pair of electrodes is used to generate one picture element.

2-77. The LCD differs from the GPDs and ELDs in that it does not generate its own light for the picture elements. The LCD requires an external light source, often called a backlight, for computer applications. The liquid crystal material between the charged electrodes becomes translucent when voltage is applied and allows the backlight to shine through as a picture element.

2-78. In the GPDs and ELDs, the picture element light is generated by ionizing a gas (neon or neon argon) between the charged electrodes (GPD) or by stimulating a luminescent material in the same manner (ELD). In either case, the picture element only emits light when voltage is applied to the electrodes.

2-79. One of the advantages of flat panel displays is that smaller voltages are required for their operation than for a CRT. GPDs use approximately 200 volts to charge the electrodes and ELDs require only 20 volts.

2-80. The picture elements in these displays are addressed by the row and column method. Displays with as many as 737,280 picture elements (960 rows by 768 columns) have been developed.

2-81. The picture elements on flat panel displays are not lighted continually. This would require a large amount of power and generate excessive heat. A sequential scan similar to a CRT raster is used. Once again, a page memory is required. The picture element electrodes turn on and off as the scan sequentially addresses page memory.

2-82. Picture elements that display a dot are momentarily turned on and off starting with the first picture element in the top row, or line, and ending with the last picture element on the bottom row. The picture elements are turned on and off at a high enough frequency that the human eye cannot detect the flicker of the off-on-off cycle.

2-83. The sequential scan used to light the picture elements is continuous and repetitive. Once again, the page memory must be changed to change the display. Flat panel displays may be formatted or unformatted in the same manner as CRT displays.

SUMMARY

2-84. This chapter has presented information on digital computer hardware. It also presented terminology needed to understand digital computers. Chapter 2 also discussed the input and output devices associated with digital computers. The information that follows summarizes the important points of this chapter.

2-85. The CENTRAL PROCESSING UNIT is the brain of the computer. This is generally referred to as the CPU or mainframe.

2-86. The CONTROL SECTION directs the flow of traffic (operations) and data and maintains order within the computer.

2-87. The ARITHMETIC-LOGIC SECTION performs all arithmetic operations (adding, subtracting, multiplying, and dividing). It also tests various conditions during processing and takes action based on the result.

2-88. INTERNAL STORAGE is sometimes referred to as primary storage, main storage, or main memory (because its functions are similar to human memory). It stores the programs and data.

2-89. MAGNETIC CORE STORAGE is made up of tiny doughnut-shaped rings made of ferrite (iron) that are strung on a grid of very thin wires.

2-90. SEMICONDUCTOR STORAGE consists of hundreds of thousands of tiny electronic circuits etched on a silicon chip.

2-91. BUBBLE STORAGE is made up of semiconductor material in the form of a very thin crystal.

2-92. READ-ONLY MEMORY allows us to permanently store programs that will not be lost even when the computer is powered down.

2-93. RANDOM-ACCESS MEMORY is read/write memory. It is the working memory. It is like a blackboard, where notes can be scribbled, then read, and then erased when finished with them.

2-94. SECONDARY STORAGE is the memory outside the main body of the computer (CPU) where programs and data are stored for future use.

2-95. MAGNETIC DISK is a direct access storage device.

2-96. INPUT/OUTPUT DEVICES are the means by which the computer communicates with the outside world. These include magnetic tape units, magnetic disk drive units, floppy disk drive units, printers (daisy-wheel, dot-matrix, ink jet, and laser), and display devices (raster scan CRT and flat panel).

Check-on-Learning Questions

1. What is the brain of a computer system?
2. How many sections make up the central processing unit?
3. What are the names of the sections that make up the CPU?
4. The control section can be compared to what?
5. What are the four major types of instructions in the control section?
6. What capability allows the arithmetic/logic section to test various conditions encountered during processing and takes action based on the result?
7. In the arithmetic/logic section, data is returned to what section after processing?
8. What is the process by which instructions and data are read into a computer?
9. Magnetic core storage is made up of what?
10. A semiconductor memory consists of what?
11. What is another name for semiconductor memory chips?
12. In computer storage, what does volatile mean?
13. What type of storage can retain its data even if there is a power failure or breakdown?
14. Bubble memory consists of what type of material?
15. How are the magnetic domains of a bubble memory switched?
16. What does it mean when it says that reading from bubble memory is nondestructive?
17. In what types of memory are often-used instructions and programs permanently stored inside the computer?
18. Who provides the programs stored in ROM?
19. Can programs in ROM be updated?
20. What is another name for random-access memory?
21. How is data read from or written into RAM?
22. In what two states, or modes, can programmable read-only memory be purchased?
23. What is the main disadvantage of PROM?
24. What is the biggest difference between PROM and EPROM?
25. How is EPROM erased?
26. Why are disk storage devices popular?
27. How is data stored on all disks?
28. What precedes each record on a disk?
29. How is the storage capacity of a disk determined?
30. What two ways can data be physically organized on a disk pack?
31. What devices are used that enables the computer to communicate with the outside world?

32. What is the normal input media for microcomputers?
33. Why is direct accessing of data a big advantage over the sequential accessing of data?
34. What is the most commonly used size floppy disk?
35. What output device expresses coded characters as hard copy (paper documents)?
36. What four types of printers are commonly used with personal computers?
37. What is the primary purpose of a keyboard?
38. Raster scan or TV scan video monitors are used extensively for what purpose?
39. How many fields make up a frame?
40. A field is approximately how many horizontal lines?
41. What are picture elements often called?
42. Vertical resolution depends on what?
43. What are used to reference page memory?
44. Flat panel displays are designed to reduce what problem of a CRT display?
45. What does the liquid crystal display require for computer applications?

Chapter 3

Software

INTRODUCTION

3-1. Software plays a major role in computer data processing. For example, without software, the computer could not perform simple addition. It is the software that makes everything happen. Putting it another way, software brings the computer to life.

COMPUTER PROGRAMS

3-2. Remember, it takes a program to make the computer function. An operating system must be loaded into the computer to manage the computer's resources and operations. Also, job information must be inputted to the operating system to tell the computer what operation to perform. For example, the following are some things a computer can do:

- Assemble or compile a Common Business Oriented Language (COBOL) program.
- Run the payroll or print inventory reports.
- Copy a tape using a utility program.
- Print the data from a disk file, also using a utility program.
- Test a program.

This job information may be entered into the computer from floppy disk or CD-ROM disk. The programmer or user may also enter the information from a remote computer terminal. The operating system receives and processes the job information and executes the programs according to that job information.

3-3. Software can be defined as all the stored programs and routines (operating aids) needed to fully use the capabilities of a computer. Generally speaking, the saying goes, "If it is not hardware, then it must be software."

OPERATING SYSTEMS

3-4. The operating system is the heart of any computer system. Through it, everything else is accomplished. An operating system is a software program that enables the computer hardware to communicate and operate with the computer software. Without an operating system, a computer would be useless. Operating systems are designed basically to provide the operator with the most efficient way of executing many user programs. An operating system is a collection of many programs used by the computer to manage its own resources and operations. These programs control the execution of other programs. They schedule, assign resources, monitor, and control the work of the computer. There are several types.

TYPES OF OPERATING SYSTEMS

3-5. As computers have progressed and developed so have the types of operating systems. Below is a basic list of the different types of operating systems and a few examples of operating systems that fall into each of the categories. Many operating systems will fall into more than one of the below categories.

Graphical User Interface

3-6. Graphical User Interface (GUI) operating systems contain graphics and icons and are commonly navigated using by using a computer mouse. See the GUI dictionary for a complete definition. The following are some examples of GUI operating systems:

- System 7.x.
- Windows 98.
- Windows CE.

Multi-user

3-7. Multi-user operating systems allow for multiple users to use the same computer at the same time and/or different times. See the multi-user dictionary for a complete definition. The following are some examples of multi-user operating systems:

- LINUX.
- UNIX.
- Windows 2000.

Multi-processing

3-8. Multi-processing operating systems are capable of supporting and using more than one computer processor. The following are some examples of multi-processing operating systems:

- LINUX.
- UNIX.
- Windows 2000.

Multi-tasking

3-9. Multi-tasking operating systems are capable of allowing multiple software processes to be run at the same time. The following are some examples of multi-tasking operating systems:

- UNIX.
- Windows 2000.

Multi-reading

3-10. Multi-reading operating systems allow different parts of a software program to run concurrently. The following are some examples of multi-reading operating systems:

- LINUX.
- UNIX.
- Windows 2000.

OPERATING SYSTEM DESIGN

3-11. Operating systems are designed to provide various operating modes. Some systems can only do one task at a time, while others can perform several at a time. Some systems allow only one person to use the system, and others allow multiple users. Single user/single tasking operating systems (such as CP/M®-80¹, CP/M-86®¹, and MS-DOS®(1, 2)²) are the simplest and most common on microcomputers. Single user/multi-tasking operating systems allow an operator to do more than one task as long as the tasks do not use the same type of resources. For example, while printing one job, another job can be running, as long as the second job does not require the printer. Examples are Concurrent CP/M-86³, Concurrent DOS³, and MS-DOS® (3.0 and above)². Multi-user and multi-tasking operating systems let more than one user access the same resources at the same time. This is especially useful for sharing common data. These are only feasible on processors (the functional unit in a computer that interprets and executes instructions) of 16 bits or more and with large memories. UNIX⁴ is an example. There are also multi-processor systems, shared resource systems. This means each user (or operator) has a dedicated microprocessor (CPU) that shares common resources (disks, printers, and so on).

NOTE:

1. CP/M and CP/M-86 are registered trademarks of Digital Research Inc.
 2. MS-DOS is a registered trademark of Microsoft Corporation.
 3. Concurrent CP/M and Concurrent DOS are trademarks of Digital Research Inc.
 4. UNIX is a trademark of AT&T.
-

COMPATIBILITY WITH APPLICATIONS SOFTWARE

3-12. To be usable, an applications program must be compatible with the operating system. Therefore, the availability of application software for a particular operating system is critical. Because of this, several operating systems have become the most popular. For 8-bit microcomputers, Control Program for Microprocessors is widely used because it has been adopted by many hardware manufacturers. MS-DOS, designed from CP/M, dominates in

lower performance 16-bit systems. UNIX, an operating system for larger computers, is being used on the more powerful 16-bit and 32-bit microcomputers. Microcomputer manufacturers also offer other operating systems.

3-13. To overcome the applications software compatibility problem, some software comes in several versions so it can be run under several different operating systems. One thing to remember is that not all applications software will run on all systems. Check to ensure compatibility exists, if not, the right version is needed.

OPERATING SYSTEM FUNCTIONS

3-14. A PC cannot do anything useful unless it is running an operating system. An operating system is a basic type of software that acts as a supervisor in the use of all applications, games, or other programs. The operating system sets the rules for using memory, drives, and other parts of the computer. However, before a PC can run an operating system, it needs some way to load the operating system from disk to RAM. The way to do this is with the bootstrap or simply to boot a small amount of code that is a permanent part of the PC. The bootstrap is aptly named because it lets the PC do something entirely on its own, without any outside operating system.

3-15. The boot operation does not do much. In fact, it has only two functions. One is to run a POST, or power-on-self-test, and the other is to search drives for an operating system.

3-16. After conducting a POST check of all the hardware components of a PC, the boot program contained on the computer's ROM Basic Input-Out System (BIOS) chips checks drive A to see whether it contains a formatted floppy disk. If a disk is mounted in the drive, the program searches specific locations on the disk for the files that make up the first two parts of the operating system. If no floppy disk is in the drive, the boot program checks for system files on the main hard drive (usually C drive). If that also fails, the program searches any disc in the CD-ROM drive. Ordinarily, these system files will not be seen because each is marked with a special file attribute that usually hides it from any file listing. For a Windows system, the files are named, IO.SYS and MSDOS.SYS. If the floppy drive is empty, the boot program checks hard drive C for the system files, and on some system, as a last resort, checks the CD-ROM drive. If a boot disk does not contain the files, the boot program generates an error message.

3-17. After locating a disk with the system files, either a floppy or within the hard-drive, the boot program reads the data stored on the disk's first sector and copies that data to specific locations in RAM. This information constitutes the boot record, commonly known as the Master Boot Record in Windows 2000.

3-18. The boot record takes control of the PC and loads the IO.SYS into RAM. The IO.SYS file contains extensions to the ROM BIOS and includes a routine called SYSINIT that manages the rest of the boot up. After loading the IO.SYS, the boot record no longer is needed and is replaced in RAM by another code.

3-19. SYSINIT assumes control of the start-up process and loads MSDOS.SYS into RAM. The MSDOS.SYS file works with the BIOS to manage files, execute programs, and respond to signals from hardware.

3-20. SYSINIT searches the root directory of the boot disk for a file named CONFIG.SYS. If it exists, it tells MSDOS.SYS to execute the commands in the file. CONFIG.SYS is a file created by the user. Its commands tell the operating system how to handle certain operations, such as how many files can be opened at one time. CONFIG.SYS can also contain code that extends the capabilities of the BIOS to control memory or hardware devices. In Windows, drivers are loaded through records in a file called the Registry.

3-21. SYSINIT tells MSDOS.SYS to load the file COMMAND.COM. This operating system file consists of three parts. One is a further extension to the input/output functions. This part is loaded in memory with the BIOS and becomes a part of the operating system.

3-22. The second part of COMMAND.COM contains the internal DOS commands, such as COPY and TYPE. It is loaded at the high end of conventional RAM, where any application program can overwrite it if it needs the memory.

3-23. The third part of COMMAND.COM is used only once and then discarded. This part searches the root directory for a file named AUTOEXEC.BAT. This file is created by the computer's user and contains a series of DOS batch file commands and/or the names of program that the user wants to run each time the computer is turned on.

3-24. When these functions are complete, the boot operation is complete. If nothing is wrong with the operating system, the computer is ready for use.

3-25. Operating systems are now found as a permanent part of some computing devices such as palm-sized PCs (which handles limited amount of information). However, in most cases, the operating system is loaded from hard disk for the following two reasons.

3-26. Upgrading the operating system is simpler when loading from a disk. When a company such as Microsoft (which makes MS-DOS and Windows - the most commonly used PC operating system) wants to add new features or fix serious bugs, it can simply issue a new set of disks or service packs. Sometimes, all that is necessary is a single file that patches a flaw in the operating system. It is cheaper for Microsoft to distribute an operating system on disk than to design a microchip that contains the operating system. It is also easier for computer users to install a new operating system from disk than it is to swap chips.

3-27. The other reason for loading an operating system from disk is that it gives users a choice of operating systems. Although most PCs based on microprocessors built by Intel uses Windows or MS-DOS, there are alternative operating systems, such as LINUX, OS/2, and UNIX.

3-28. There are several actions that must be performed that are beyond basic functions. These actions include checking the OS version, the Device Manger, and the Control Panel.

CHECKING OPERATING SYSTEM (OS) VERSION

3-29. In order to check the version of the OS in use, simply open the System Properties Window. This can be done in several ways:

- Click Start, then Settings, and then Control Panels. Scroll down until the system icon is visible, and Double-Click that icon
- Right-Click My Computer, and click on Properties.
- Or simply click on Start – Run – and type WINVER and press ENTER.

The first two choices may display a window with several more pieces of information besides just the operating system version. It also tells us the Registered Key used to install the OS and the name of the registered user, which can help during re-installations and when contacting technical support. It also displays the amount of RAM and the processor type installed on the computer, which can help with diagnosing problems.

DEVICE MANAGER

3-30. The most important piece of troubleshooting information that can be found is in a little application known as the Device Manager. In Windows 2000, it is found under the Hardware tab, but in Windows '98, it has its own tab. The Device Manager gives a quick glimpse at drivers and equipment that are installed on a computer. It also displays how they are functioning.

3-31. By clicking the (+) sign beside an item, the exact drivers that are being used by that equipment are displayed. Yellow circles over items mean the item is not functioning properly and has its driver's temporarily suspended until the problem is diagnosed. A red circle over an item is a driver that has been manually turned off by a user. The most important aspect of this is the fact that it shows devices that are working and those which are not.

CONTROL PANEL

3-32. The other major features of Windows 9x/ME/XP/2000 are its control panels. These are applications that reside within Windows that allow the operator to perform very distinct functions. The Device Manager resides within here, as well as the Printer Control Panel, Sound Control Panel, Modem Control Panel, Networking Control Panel, and more. These control panels are used to provide specific functions that are not part of outside software.

3-33. For example, a modem is required to connect to the Internet. Use the Modem Control Panel to detect, set up, and configure the modem. Use the Networking Control Panel to add and configure Transmission Control Protocol/Internet Protocol (TCP/IP) and client software. Finally, use the Internet Control Panel to set up the dialing properties and connection specifications.

3-34. It is impossible to go into detail on each control panel. However, the name of the control panel should explain what it does. Select each control panel to learn the functions of each one. The most used ones are the

Accessibility Options, Printers, Networking, Modem, and System control panels.

UTILITY PROGRAMS

3-35. Another type of program is called utilities. In addition to the utility commands (like disk copy and rename), which are built into the operating system, there are some other independent utility programs. These are standard programs that run under control of the operating system just like the applications programs. They are called utilities because they perform general types of functions that have little relationship to the content of the data. Utility programs eliminate the need for programmers to write new programs when all they want to do is copy, print, or sort a data file. Although a new program is not needed, the operator will have to tell the program what they want it to do. This is done by providing information about files, data fields, and the process to be used. For example, a sort program arranges data records in a specified order. The operator will have to tell the sort program what fields to sort on and whether to sort in ascending or descending sequence.

3-36. Let us examine two types of utility programs to get some idea of how a utility program works. The first will be sort-merge and the second, the report program generators.

SORT-MERGE PROGRAMS

3-37. Sorting is the term given to arranging data records in a predefined sequence or order. Merging is the combining of two or more ordered files into one file. An example of this is by putting a list of people's names in alphabetical order and then arranging them in sequence by last name.

3-38. By knowing the alphabetical sequence (B comes after A, C after B, and so on); it is easy to arrange a list manually, even though it is time consuming. However, on a computer; the sequence of characters is also defined. It is called the collating sequence. Every coding system has a collating sequence. The capability of a computer to compare two values and determine which is greater (B is greater than A, C is greater than B, and so on) makes sorting possible. Numbers and special characters are also part of the collating sequence. In Extended Binary Coded Decimal Interchange Code (EBCDIC), which is discussed in detail in chapter 4, special characters such as #, \$, &, and * are located in front of alphabetic characters, and numbers follow. When records are sorted in the defined sequence, they are in ascending sequence. Most sort programs also allow sorting in reverse order. This is called descending sequence. In EBCDIC, it is 9-0, Z-A, then special characters.

3-39. To sort a data file, the operator must tell the sort program what data field or fields to sort on. These fields are called sort (or sorting) keys. In the example, the last name is the major sort key and the first name is the minor sort key.

3-40. Sorting is needed in many applications. For example, addresses will be in zip code order for mailings; personnel records may be kept in service number order; and inventory records may be kept in stock number order, and so on. Since many of the files are large, sorting is very time-consuming and it is one of the processes most used on computer systems. The more a process is used, the more familiar an operator will become of the process.

3-41. Sort-merge programs usually have phases. The sort-merge program initializes the program by reading the parameters, producing the program code for the sort, allocating the memory space, and setting up other functions. The sort-merge program then reads in as many input data records as the memory space allocated can hold, arranges (sorts) them in sequence, and writes them out to an intermediate sort-work file. It continues reading input, sorting and writing intermediate sort-work files, until all the input is processed. It then merges (combines) the ordered intermediate sort-work files to produce one output file in the sequence specified. The merging process can be accomplished with less memory than the sort process since the intermediate sort-work files are all in the same sequence. Records from each work file can be read, the sort keys compared based on the collating sequence and sort parameters, and records written to the output file maintaining the specified sequence.

REPORT PROGRAM GENERATORS

3-42. Report Program Generators (RPGs) are used to generate programs to print detail and summary reports of data files. Figure 3-1 is an example of a printed report. RPGs were designed to save programming time. Rather than writing procedural steps in a language like Beginner's All-Purpose Symbolic Instruction Code (BASIC) or COBOL, the RPG programmer writes the printed report requirements on specially designed forms.

SUMMARY OF REQUISITIONS BY UNIT IDENTIFICATION CODE		
<u>UIC</u>	<u>NUMBER OF REQUISITIONS</u>	<u>TOTAL COST</u>
WA4MAA	15	\$ 25,275.00
WAA9AA	27	\$100,150.00
WABHAA	10	\$ 80,000.00
<u>TOTALS</u>	<u>52</u>	<u>\$205,425.00</u>

Figure 3-1. Printed Report Using a RPG Program

- 3-43. The following are included in the report requirements:
- An input file description.
 - The report heading information lines.
 - The input data record fields.

- The calculations to be performed.
- The data fields to be printed and summarized.

The RPG program takes this information and generates a program for the specific problem. That program will then be run with the specified input data file to produce the printed report. The input data file must be in the sequence in which the report, to be generated, is to summarize the data.

3-44. In Figure 3-1, requisitions were summarized based on unit identification codes (UICs). First, the input data file was sorted on the field that contained the UIC. Next, specifications were provided to the RPG program to tell it to accumulate totals from the detail (individual) data records until the UIC changed. Lastly, the total number of requisitions and total cost for that UIC were printed. Also (even though not shown), each detail record could have been printed. The UIC is called the control field. Each time the control field changes, there is a control break. Each time there is a control break the program prints the summary information. After all records are read and processed, it prints a summary line (TOTALS) for all UICs. RPGs can also be used to generate a program to update data files.

PROGRAMMING LANGUAGES

3-45. Programmers must use a language that can be understood by the computer. Several methods can achieve human-computer communication. For example, let us assume the computer only understands French and the programmer speaks English. One approach to communicate with the computer is for the programmer to code the instructions with the help of a translating dictionary before giving them to the processor. This would be fine so far as the computer is concerned; however, it would be very awkward for the programmer.

3-46. Another approach is a compromise between the programmer and computer. The programmer first writes instructions in a code that is easier to relate to English. This code is not the computer's language; therefore, the computer does not understand the orders. The programmer solves this problem by giving the computer another program, one that enables it to translate the instruction codes into its own language. This translation program would be equivalent to an English-to-French dictionary, leaving the translating job to be done by the computer.

3-47. The third and most desirable approach from an individual's standpoint is for the computer to accept and interpret instructions written in everyday English terms. Each of these approaches has its place in the evolution of programming languages and is used today in computers.

MACHINE LANGUAGES

3-48. With early computers, the programmer had to translate instructions into the machine language form that the computers understood. This language was a string of numbers that represented the instruction code and operand address(es).

3-49. In addition to remembering dozens of code numbers for the instructions in the computer's instruction set, the programmer also had to keep track of the storage locations of data and instructions. This process was very time-consuming, quite expensive, and often resulted in errors. Correcting errors or making modifications to these programs was a very tedious process.

SYMBOLIC LANGUAGES

3-50. In the early 1950s, mnemonic instruction codes and symbolic addresses were developed. This improved the program preparation process by substituting letter symbols (mnemonic codes) for basic machine language instruction codes. Each computer has mnemonic codes, although the symbols vary among the different makes and models of computers. The computer still uses machine language in actual processing, but it translates the symbolic language into machine language equivalent. The following are some advantages of using symbolic languages over machine language coding:

- Less time is required to write a program.
- Detail is reduced.
- Fewer errors are made (errors which are made are easier to find).
- Programs are easier to modify.

PROCEDURE-ORIENTED LANGUAGES

3-51. The development of mnemonic techniques and macroinstructions led to the development of procedure-oriented languages. Macroinstructions allow the programmer to write a single instruction that is equivalent to a specified sequence of machine instructions. These procedure-oriented languages are oriented toward a specific class of processing problems. A class of similar problems is isolated and a language is developed to process these types of applications. Several languages have been designed to process problems of a scientific-mathematical nature and others that emphasize file processing.

3-52. Procedure-oriented languages were developed to allow a programmer to work in a language that is close to English or mathematical notation. This improves overall efficiency and simplifies the communications process between the programmer and the computer. These languages have allowed us to be more concerned with the problems to be solved rather than with the details of computer operation. For example:

- COBOL was developed for business applications. It uses statements of everyday English and is good for handling large data files.
- FORMula TRANslator (FORTRAN) was developed for mathematical work. Engineers, scientists, statisticians, and others for whom mathematical operations are most important often use this language.
- BASIC, although this form of programming is no longer practiced as widely, was designed as a teaching language to help beginning programmers write programs. Therefore, it is a general-purpose,

introductory language that is fairly easy to learn and to use. With the increase in the use of microcomputers, BASIC has regained popularity and is available on most microcomputer systems.

3-53. Other languages gaining in popularity are PASCAL and Ada. PASCAL is being used by many colleges and universities to teach programming because it is fairly easy to learn, yet is a more powerful language than BASIC. Although PASCAL is not yet a standardized language, it is still used rather extensively on microcomputers. It has greater programming capabilities on small computers than are possible with BASIC.

3-54. Ada's development was initiated by the United States Department of Defense. Ada is a modern general-purpose language designed with the professional programmer in mind and has many unique features to aid in the implementation of large-scale applications and real-time systems. Since Ada is so strongly supported by the DOD and other advocates, it will become an important language like those previously mentioned. Its primary disadvantage relates to its size and complexity, which will require considerable adjustment on the part of most programmers.

3-55. The most familiar of the procedure-oriented languages are BASIC and FORTRAN for scientific or mathematical problems and COBOL for file processing.

3-56. Programs written in procedure-oriented languages, unlike those in symbolic languages, may be used with a number of different computer makes and models. This feature greatly reduces reprogramming expenses when changing from one computer system to another. The following are some other advantages to procedure-oriented languages:

- They are easier to learn than symbolic languages.
- They require less time to write.
- They provide better documentation.
- They are easier to maintain.

However, there are some disadvantages of procedure-oriented languages. They require more space in memory and they process data at a slower rate than symbolic languages.

PROGRAMMING

3-57. Programming is simply the process of planning the computer solution to a problem. A generalized process or program for finding the total resistance of a parallel resistance circuit can be derived by computing the following steps:

- **Step 1.** Take the reciprocal of the resistance of all resistors (expressed in ohms).
- **Step 2.** Sum the values obtained in step 1.
- **Step 3.** Take the reciprocal of the sum derived in step 2.

To progress from this example to preparing a program for a computer is not difficult. However, one basic characteristic to remember about the computer is that it cannot think. It can only follow certain commands, and these commands must be correctly expressed and must cover all possibilities. If a program is to be useful in a computer, it must be broken down into specifically defined operations or steps. Then the instructions, along with other data necessary for performing these operations or steps, must be communicated to the computer in the form of a language or code that is acceptable to the machine. In broader terms, the computer follows certain steps in executing a program. It must first read the instructions (sequentially unless otherwise programmed) and then in accordance with these instructions, it executes the following procedures:

- Locates the parameters (constants) and such other data as may be necessary for problem solution.
- Transfers the parameters and data to the point of manipulation.
- Manipulates the parameters and data in accordance with certain rules of logic.
- Stores the results of such manipulations in a specific location.
- Provides the operator (user) with a useful output.

3-58. Even a program of elementary characters (such as the one above) would involve breaking each of the steps down into a series of machine operations. Then these instructions, parameters, and the data necessary for problem solution must be translated into a language or code that the computer can accept. Next, an introduction will be provided to the problem solving concepts and flowcharting necessary to develop a program.

OVERVIEW OF PROGRAMMING

3-59. Before learning to program in any language, it is helpful to establish some context for the productive part of the entire programming effort. This context includes the understanding and agreement that there are four fundamental and discrete steps involved in solving a problem on a computer. The four steps are as follows:

- **Step 1.** State, analyze, and define the problem.
- **Step 2.** Develop the program logic and prepare a program flowchart or decision table.
- **Step 3.** Code the program, prepare the code in machine-readable form, prepare test data, and perform debug and test runs.
- **Step 4.** Complete the documentation and prepare operator procedures for implementation and production.

3-60. Figure 3-2 depicts the evolution of a program. Programming can be complicated. Before actually starting to write or code a program, advance preparation is required. The first two steps, problem understanding/definition and flowcharting, fall into the advance planning phase of programming. It is important at this point to develop correct habits and procedures, since this will prevent later difficulties in program preparation.

3-61. Whether working with a systems analyst, a customer, or solving a problem on your own, it is extremely important that there is a thorough understanding of the problem. Every aspect of the problem must be defined:

- What is the problem?
- What information (or data) is needed?
- Where and how will the information be obtained?
- What is the desired output?

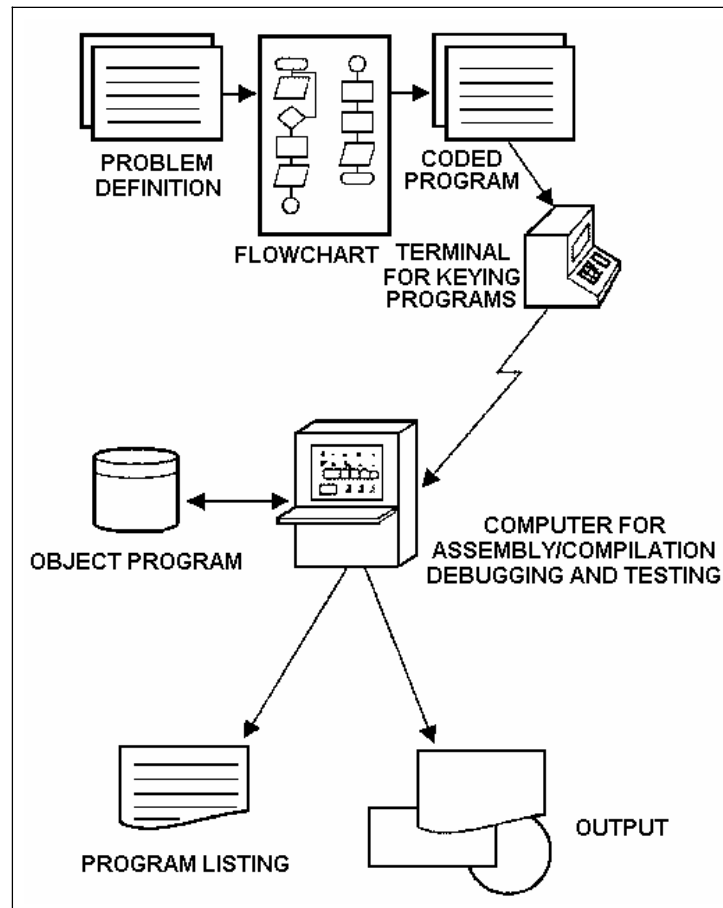


Figure 3-2. Evolution of a Program

3-62. Starting with only a portion of the information, or an incomplete definition, will result in having to constantly alter what has been done to accommodate the additional facts as they become available. It is easier and more efficient to begin programming after all of the necessary information is understood. Once there is a thorough understanding of the problem, the next step is flowcharting.

FLOWCHARTING

3-63. Flowcharting is one method of pictorially representing a procedural (step-by-step) solution to a problem. Prepare a flowchart before starting to write the computer instructions required to produce the desired results.

Flowcharts use different shaped symbols connected by one-way arrows to represent operations, data flow, equipment, and so forth.

3-64. The two types of flowcharts are system (data) flowcharts and programming flowcharts. A system (data) flowchart (see Figure 3-3) defines the major phases of the processing, as well as the various data media used. It shows the relationship of many jobs that make up an entire system. In the system (data) flowchart, an entire program run or phase is always represented by a single processing symbol, together with the input/output symbols showing the path of data through a problem solution.

3-65. The second type of flowchart and the one covered in this TC is the programming flowchart (see Figure 3-4). It is constructed by the programmer to represent the sequence of operations the computer will perform to solve a specific problem. It graphically describes what will take place in the program. It displays specific operations and decisions and their sequence within the program.

TOOLS OF FLOWCHARTING

3-66. It is important to know the tools used in flowcharting. These tools are the fundamental symbols, graphic symbols, flowcharting template, and flowcharting worksheet.

3-67. **FUNDAMENTAL SYMBOLS.** Know the symbols and their related meanings before starting to construct a flowchart. These symbols are standard for the military (as directed by Department of the Army Automated Data Systems Documentation Standards).

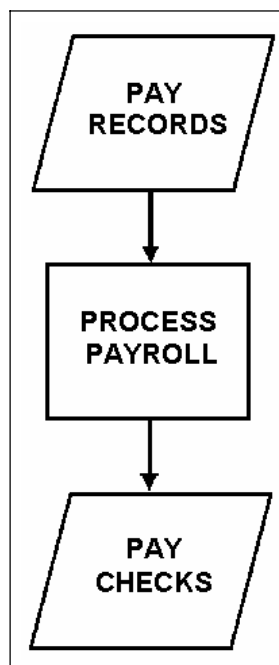


Figure 3-3. System Flowchart

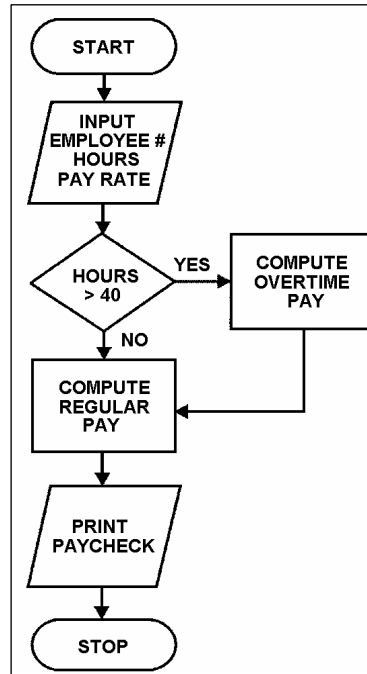


Figure 3-4. Programming Flowchart

3-68. Symbols are used to represent functions. These fundamental functions are processing, decision, input/output, terminal, flow lines, and connector symbols. All flowcharts may be initially constructed using only these fundamental symbols as a rough outline to work from. Each symbol corresponds to one of the functions of a computer and specifies the instruction(s) to be performed by the computer. The contents of these symbols are called statements. Figure 3-5 shows the shape, definition, example, and explanation of the fundamental flowcharting symbols.

3-69. **GRAPHIC SYMBOLS.** Within a flowchart, graphic symbols are used to specify arithmetic operations and relational conditions. The following are commonly used arithmetic and relational symbols:

+	plus, add
-	minus, subtract
*	multiply
/	divide
±	plus or minus
=	equal to
>	greater than
<	less than
≥	greater than or equal to
≤	less than or equal to
≠	not equal
YES or Y	(Yes)
NO or N	(No)
TRUE or T	(True)
FALSE or F	(False)

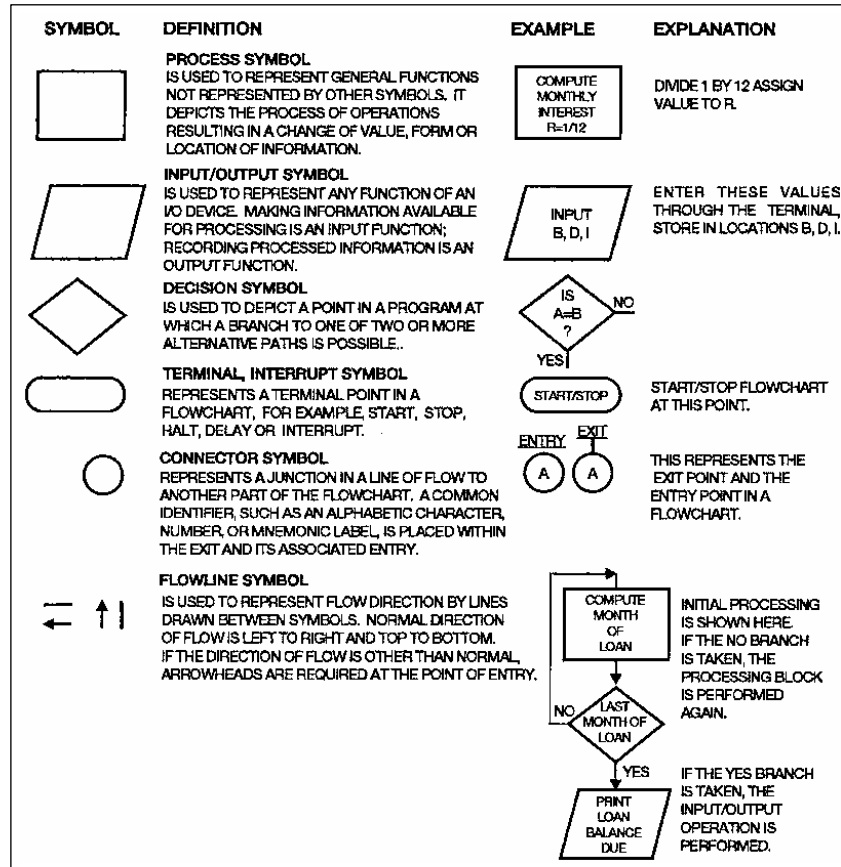


Figure 3-5. Fundamental Flowcharting Symbols

3-70. **FLOWCHARTING TEMPLATE.** Use a flowcharting template in order to aid in the drawing of flowcharting symbols. Figure 3-6 shows a template containing the standard symbol cutouts. A template is usually made of plastic with the symbols cut out to allow tracing the outline.

3-71. **FLOWCHARTING SOFTWARE.** A more simple form and easier way of drawing flowcharts is to use software developed for this task. Microsoft developed VISIO®, a complete and very easy to use program that allows the programmer to lay out the flow of program with a simple drag-and-drop function. This program can be found on the web at (<http://www.microsoft.com/office/visio>). However, the most popular would have to be SmartDraw 6.0. This can be found on the web at (<http://www.smartdraw.com>). Developed by SmartDraw, it is a relatively smaller program that offers all the vital tools for flowcharting and its interface is easier to understand and use.

3-72. **FLOWCHART WORKSHEET.** The flowchart worksheet is a means of standardizing documentation. It provides space for drawing programming flowcharts and contains an area for identification of the job (including application, procedure, date, and page numbers). The flowchart worksheet can be helpful in developing flowcharts. If this form is not available, a plain piece of paper will do.

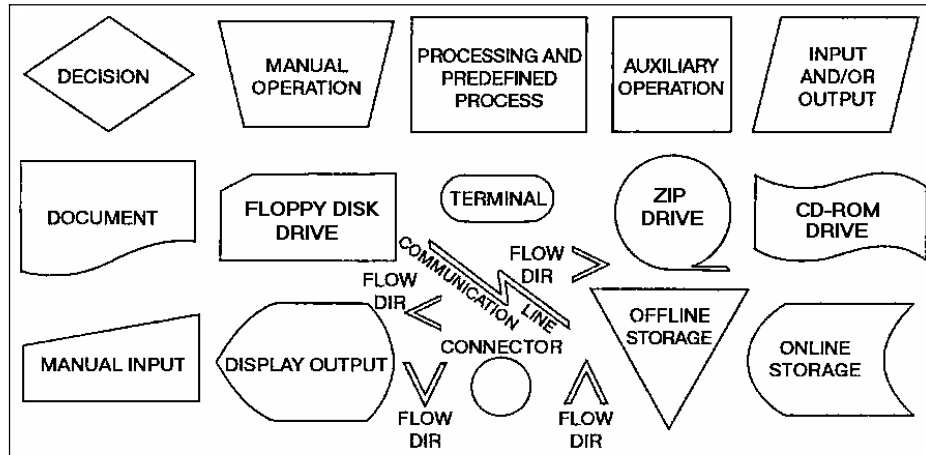


Figure 3-6. Flowchart Template

CONSTRUCTING A FLOWCHART

3-73. There is no "best way" to construct a flowchart. There is no way to standardize problem resolution. Flowcharting and programming techniques are often unique and conform to the individual's own methods or direction of problem resolution.

3-74. The following will show an example of developing a programming flowchart. It is not the intent to say this is the best way; rather, it is just one of the ways. By following this text example, an operator should grasp the idea of solving problems through flowchart construction. As the operator gains experience and familiarity with a computer system, these ideas will serve as a foundation.

3-75. The first thing that needs to be known before developing a flowchart is what problem needs to be solved. After that, study the problem definition and develop a flowchart to show the logic, steps, and sequence of steps the computer must execute to solve the problem.

3-76. As an example, suppose you have taken out a short-term second mortgage on a new home and you want to determine the following:

- Real costs.
- Amount of interest.
- Amount to be applied to principal.
- Final payment at the end of the three-year loan period.

The first step in completely understanding the problem is to know what are the inputs and the outputs and what steps are needed to answer the questions. Even when specifying a problem of your own, you will normally not think in small, detailed, sequential steps. However, that is exactly how a computer operates—one step after another in a specified order. Therefore, it is necessary to think the problem solution through step by step. The problem might be clarified by using the Problem Definition (see Figure 3-7).

3-77. After you have attained this level of narrative problem definition, you are ready to develop a flowchart showing the logic and sequence of steps you want the computer to execute to solve the problem. Figure 3-7 also shows a programming flowchart of this problem. Study both the problem definition and the flowchart to see their relationship and content.

3-78. You now have a plan of what you want the computer to do. The next step is to code a program that can be translated by a computer into a set of instructions it can execute. This step is called program coding.

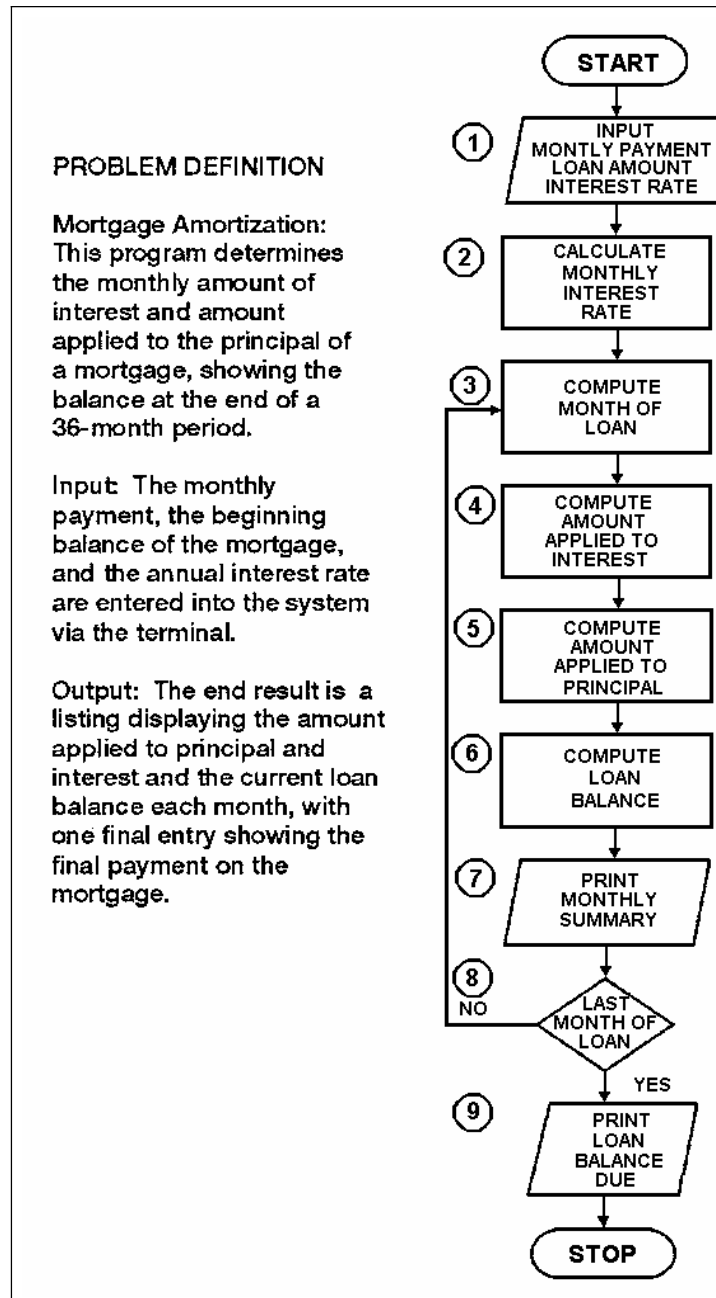


Figure 3-7. Problem Definition and Programming Flowchart

PROGRAM CODING

3-79. It is important to remember that program coding is not the first step of programming. Too often the tendency is to start coding too soon. Remember, a great deal of planning and preparation must be done before sitting down to code the computer instructions to solve a problem. For the example amortization problem (see Figure 3-7), the specifications have been analyzed in terms of the following:

- The output desired.
- The operations and procedures required in producing the output.
- The input data needed.

In conjunction with this analysis, a programming flowchart has been developed that outlines the procedures for taking the input data and processing it into usable output. The operator is now ready to code the instructions that will control the computer during processing. This requires that the operator knows a programming language. All programming languages (FORTRAN, COBOL, BASIC and so on) are composed of instructions that enable the computer to process a particular application or perform a particular function.

INSTRUCTIONS

3-80. The instruction is the fundamental element in program preparation. Like a sentence, an instruction consists of a subject and a predicate. However, the subject is usually not specifically mentioned; rather, it is some implied part of the computer system directed to execute a command. For example, the chief tells a sailor to "dump the trash." The sailor will interpret this instruction correctly even though the subject "you" is omitted. Similarly, if the computer is told to "ADD 1234," the control section may interpret this to mean that the arithmetic-logic section should add the contents of address 1234 to the contents of the accumulator (a register in which the result of an operation is formed).

3-81. In addition to an implied subject, every computer instruction has an explicit predicate consisting of at least two parts. The first part is referred to as the command or operation; it answers the question "what?" It tells the computer what operation to perform (such as read, print, and input). An operation code is used to communicate the programmer's intent to the computer. The second specific part of the predicate, known as the operand, names the object of the operation. In general, the operand answers the question "where?" Operands may indicate the following:

- The location of data to be processed.
- The location where the result of processing will be stored.
- The location of the next instruction to be executed. When this type of operand is not specified, the instructions are executed in sequence.

3-82. The number of operands and the structure or format of the instructions varies from one computer to another. However, the operation

always comes first in the instruction and is followed by the operand(s). The programmer must prepare instructions according to the format required by the language and the computer to be used.

INSTRUCTION SET

3-83. The number of instructions in a computer's instruction set may range from less than 30 to more than 100. These instructions may be classified into categories by the action they perform (such as input/output, data movement, arithmetic, logic, and transfer of control). Each of these categories are described below:

- **Input/output.** These instructions are used to communicate between I/O devices and the central processor.
- **Data movement.** These instructions are used for copying data from one storage location to another and for rearranging and changing data elements in some prescribed manner.
- **Arithmetic.** These instructions permit addition, subtraction, multiplication, and division. They are common in all digital computers.
- **Logic.** These instructions allow comparison between variables, or between variables and constants.
- **Transfer of control.** These instructions are classified into two types—conditional and unconditional.
 - Conditional transfer of control instructions is used to branch or change the sequence of program control, depending on the outcome of the comparison. If the outcome of a comparison is true, control is transferred to a specific statement number. If it proves false, processing continues sequentially through the program.
 - Unconditional transfer of control instructions is used to change the sequence of program control to a specified program statement regardless of any condition.

CODING A PROGRAM

3-84. Regardless of the language used, the programmer must adhere to strict rules with regard to punctuation and statement structure when coding any program. Using the programming flowchart shown in Figure 3-4, a program coded in BASIC has been added to show the relationship of the flowchart to the actual coded instructions (see Figure 3-8). Do not worry about complete understanding, just look at the instructions with the flowchart to get an idea of what coded instructions look like.

3-85. Specific information about the computer is needed in order to code a program. For example, how the language is implemented on that particular computer. The computer manufacturer or software designer will provide these specifics in their user's manual. Get a copy of the user's manual and study it before starting to code. The differences may seem minor, but they may prevent a program from running. Once coding is completed, the program must be debugged and tested before implementation.

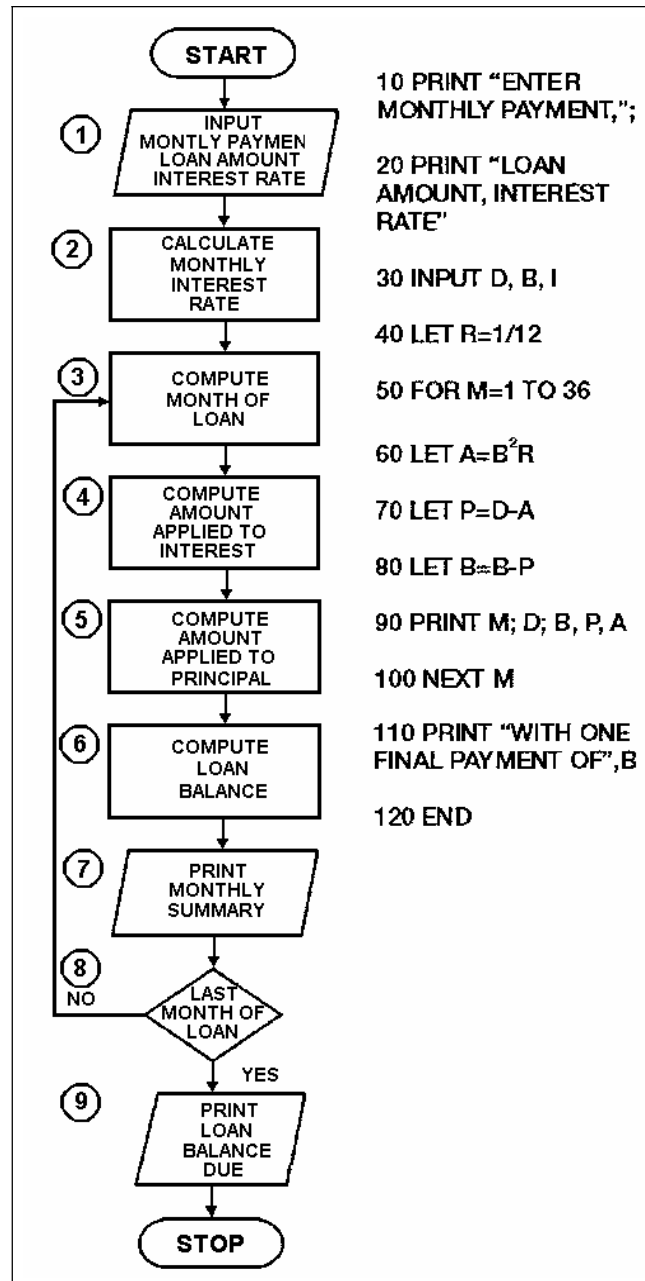


Figure 3-8. Programming Flowchart and Coded Program

DEBUGGING

3-86. Errors caused by faulty logic and coding mistakes are referred to as "bugs." Finding and correcting these mistakes and errors that prevent the program from running and producing correct output is called "debugging."

3-87. Rarely do complex programs run to completion on the first attempt. Time spent debugging and testing often equals or exceeds the time spent in program coding. This is particularly true if insufficient time was spent on

program definition and logic development. The following are some common mistakes that cause program bugs:

- Mistakes in coding punctuation.
- Incorrect operation codes.
- Transposed characters.
- Keying errors.
- Failure to provide a sequence of instructions (a program path) needed to process certain conditions.

3-88. To reduce the number of errors, carefully check the coding sheets before they are keyed into the computer. This process is known as "desk-checking" and should include an examination for program completeness. Typical input data should be manually traced through the program processing paths to identify possible errors. After desk-checking the program for accuracy, the program is ready to be assembled or compiled. Assembly and compiler programs prepare the program (source program) to be executed by the computer. These programs will also have error diagnostic features that detect certain types of mistakes in the program. These mistakes must be corrected. Even if the program passes error-free through the assembly or compiler program, this does not mean the program is perfected. However, it usually means the program is ready for testing.

TESTING

3-89. Once a program reaches the testing stage, it usually has proven it will run and produce output. The purpose of testing is to determine that all data can be processed correctly and that the output is correct. The testing process involves processing input test data that will produce known results. The test data should include the following:

- Typical data, which will test the commonly used program paths.
- Unusual but valid data, which will test the program paths used to process exceptions.
- Incorrect, incomplete, or inappropriate data, which will test the program's error routines.

If the program does not pass these tests, more testing is required. Examine the errors and review the coding to make the needed coding corrections. When the program passes these tests, it is ready for computer implementation. Before computer implementation takes place, documentation must be completed.

DOCUMENTATION

3-90. Documentation is a continuous process. Documentation begins with the problem definition. Documentation involves collecting, organizing, storing, and otherwise maintaining a complete record of the programs and other documents associated with the data processing system.

3-91. The Army has established documentation standards to ensure completeness and uniformity for the use of computer system information

between commands and between civilian and Army organizations. A documentation package should include the following:

- **A definition of the problem.** Why was the program written? What were the objectives? Who requested and approved the program? These are the types of questions that should be answered.
- **A description of the system.** The system environment (hardware, software, and organization) in which the program functions should be described (including systems flowcharts). General systems specifications should be clearly defined outlining the scope of the problem, the form and type of input data to be used, and the form and type of output required.
- **A description of the program.** Programming flowcharts, program listings, program controls, test data, test results, and storage dumps. These and other documents should be included that describe the program and give a historical record of problems and/or changes.
- **Operator instructions.** Items that should be included are computer switch settings, loading and unloading procedures, and starting, running, and termination procedures.

IMPLEMENTATION

3-92. The program is ready for use after the documentation is complete and the test output is correct. If a program is designed to replace a program in an existing system, it is generally wise to have a period of parallel processing. This means that the job application is processed both by the old program and by the new program. The purpose of this period is to verify processing accuracy and completeness.

PACKAGED SOFTWARE

3-93. Fortunately, a program does not have to be written for every problem to be solved. Instead, packaged or off-the-shelf programs can be used that are designed for specific classes of applications. More and more packaged software (software written by the manufacturer, a software house, or CDA) becomes available every day for general use. It may be up to the operator to set up and process a job within the specifications of a packaged program. Four classes of packaged software (word processing, data management, spreadsheets, and graphics) that are used in everyday work are described below.

WORD PROCESSING

3-94. Word processing software is used for any function that involves text (letters, memos, forms, reports, and so on). At a minimum, it includes routines for creating, editing, storing, retrieving, and printing text. Under the word processing software control, text is generally entered from the keyboard (see Figure 3-9) and it is printed on a display screen. After inputting of text is completed, it can be stored on a floppy disk or CD-ROM disk, sent to print, or continue to make changes. Changes to text (adding or deleting words, characters, lines, sentences, or paragraphs) can be made by

using the edit functions. Text can also be rearranged. For example, move a paragraph or block of information to another place in the same document or even move it to a different document. Word processing is particularly useful for text documents that are repetitive or that require a lot of revisions. It saves a lot of re-keying.

Word-processing software saves keystrokes. It helps you create, store, retrieve, edit, and print text. You can insert, delete, and rearrange text. You can also use special features like spelling checkers, mailing lists, document compilation, and communications.

Figure 3-9. Word Processing Example

3-95. Other features and software often available with a word processing software package include the following:

- Spell and grammar checks.
- Mailing list programs.
- Document compilation programs.
- Communications programs.

3-96. The spell and grammar checks help find misspelled words (but not misused words) and misuse of sentence structure and punctuation. During spell check, the software scans the text matching each word against a built-in dictionary of words. If the word is not found in the dictionary, the system flags the word. If known, correct the spelling or check the spelling by using a dictionary. The grammar check flags sentence structure and usually recommends a better wording. Even with these checks, the document must still be proofread to see that everything was keyed and that the words are used correctly.

3-97. Mailing list programs are designed for maintaining name and address files. They often include a capability to individualize letters and reports by inserting names, words, or phrases.

3-98. Document compilation programs are useful when there are standard paragraphs of information that needs to be combined in different ways for various purposes. For example, if answering inquiries or putting together contracts or proposals; once the standard paragraphs that are needed are selected, variable information can then be added. This saves both keying and proofreading time.

3-99. Communications software and hardware enables an operator to transmit and receive text on a microcomputer. Many organizations use this capability for electronic mail. In a matter of minutes an operator can enter and transmit a memo to other commands or to personnel in other locations. Monthly reports, notices, or any documents prepared on the microcomputer can also be transmitted.

DATA MANAGEMENT

3-100. Data management software allows an operator to enter data and then retrieve it in a variety of ways. First, data fields must be defined and also a display screen setup with prompts. Next, enter the data records according to the prompts (see Figure 3-10). The system writes the records on a floppy disk or CD-ROM disk. Once a file has been keyed and stored, records can be retrieved by a field or several fields or by searching the records for specific data. For example, if a list of all personnel who reported aboard ship before January 2004 is needed, tell the system to search the file and print selected fields of all records that meet that condition. Tell the system what fields to print (such as name, grade, SSN, and date reported) and where (what print positions) to print them. At the same time, specify in what order the records are to be printed. Figure 3-11 shows an example of the records printed in alphabetical order by last name. The software also provides routines so records can be easily added, deleted, or changed.

Last Name:	MOORE
First Name:	Lawrence
Grade:	E-6
SSN:	987-65-4321
Reporting Date:	1 JUNE 2003
Telephone No.:	(804) 765-4321

Figure 3-10. Data Management Example (Prompts in Bold and Data in Italics)

Grade:	Last Name:	First Name:	SSN:	Reporting Date:
E-6	Frederick	Jan	123-45-6789	15 May 2003
E-6	Moore	Lawrence	987-65-4321	1 June 2002
E-7	Shepherd	Vernon	000-01-2000	1 May 2002
E-5	Smith	Ronald	000-02-3000	1 August 2001

Figure 3-11. Data Management Example (Sample Printed Report Sorted by Last Name)

3-101. Reports can also be generated by specifying what records to use, what fields to print, where to print the fields, and which data fields, if any, need to be combined. For example, the supply officer wants to know the value of the inventory. The extended price can be calculated by multiplying the item quantity by the unit price and then the extended prices can be totaled (see Figure 3-12). The information to be used in report and column headings can also be specified.

INVENTORY VALUE			
<i>ITEM</i>	<i>QUANTITY</i>	<i>UNIT PRICE</i>	<i>EXTENDED PRICE</i>
Swabs	47	1.65	77.55
Brooms	62	2.25	139.50
Cleanser	36	1.85	66.60
TOTAL			283.65

Figure 3-12. Data Management Example (Calculation of Inventory Value)

3-102. Data management programs on microcomputers are not as sophisticated as the database management systems on mainframes and minicomputers. However, they do provide an extremely useful capability in offices or aboard ship.

SPREADSHEETS

3-103. A spreadsheet is a table of rows and columns (see Figure 3-13). Spreadsheet processors allow the set up a table of rows and columns to add headings and specify what calculations to perform on the columns. Values for the basic information must be entered into the appropriate rows and columns. Then the processor performs the calculations. Figure 3-13 is an example of a spreadsheet showing the projected magnetic media requirements costs. Enter the item descriptions, column headings, report title, and data for columns 1, 2, and 4. The software calculates column 3 by adding columns 1 and 2. The software also multiplies column 3 times column 4 and puts the result in column 5. It can also subtotal and total the columns specified, in this case, columns 1 through 3 and column 5.

MAGNETIC MEDIA REQUIREMENTS SPREADSHEET					
<i>ITEM</i>	<i>NUMBER TO BE REPLACED</i>	<i>NUMBER FOR EXPANSION</i>	<i>TOTAL NEEDED</i>	<i>COST PER ITEM</i>	<i>TOTAL COST</i>
	(1)	(2)	(3)	(4)	(5)
CD-ROM	<i>15</i>	<i>30</i>	45	<i>27.50</i>	1,237.50
ZIP DISK	<i>4</i>	<i>5</i>	9	<i>130.00</i>	1,170.00
DISKETTES 3 1/4"	<i>10</i>	<i>50</i>	60	<i>1.90</i>	114.00
TOTAL	29	85	114		2,521.50

ITALICS -- DATA YOU ENTER
BOLD -- INFORMATION CALCULATED BY THE PROGRAM

Figure 3-13. Spreadsheet Example

GRAPHICS

3-104. Graphics capability is available on many microcomputers. One use is to produce data displays like graphs, pie charts (see Figure 3-14), and bar charts (see Figure 3-15). On some micros line drawings can be created, on others, sophisticated engineering drawings can be created. High-resolution color graphics are also available for specialized applications.

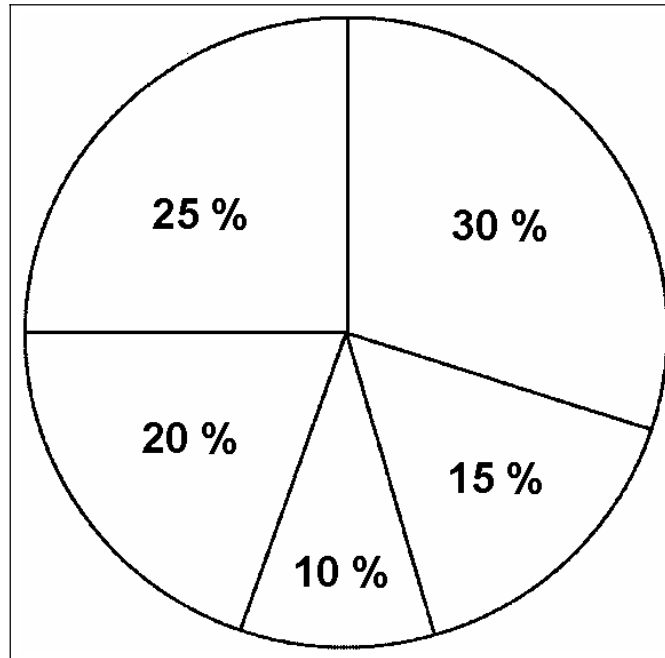


Figure 3-14. Graphics Examples (Pie Chart)

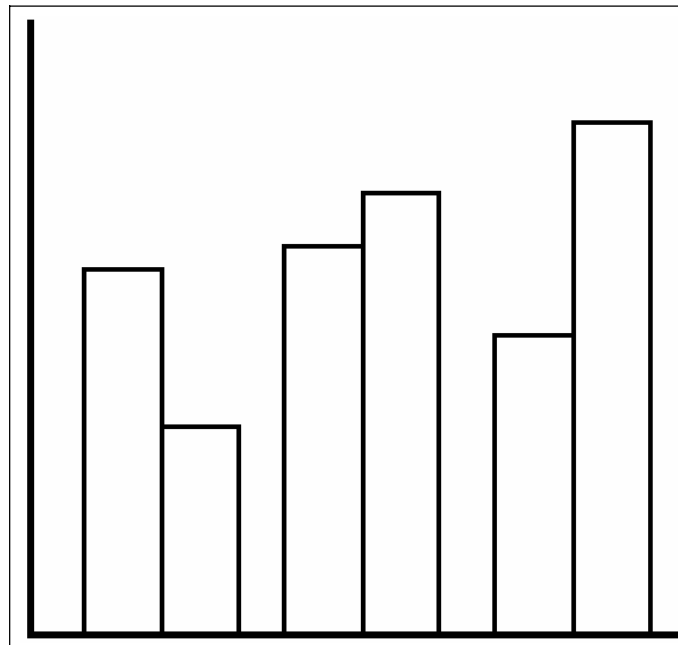


Figure 3-15. Graphics Examples (Bar Chart)

3-105. All printers are not compatible for graphics output. They must be capable of producing graphics and also be compatible with the software. Some character printers can be used for limited graphics. Dot-matrix printers and plotters work well for graphics output. Laser and ink jet printers are also good for both text and graphics.

SUMMARY

3-106. This chapter has presented information on digital computers software. The information that follows summarizes the important points in this chapter.

3-107. OPERATING SYSTEMS are a collection of many programs used by the computer to manage its own resources and operations and to perform commonly used functions like copy, print, and so on.

3-108. UTILITY PROGRAMS perform such tasks as sorting, merging, and transferring (copying) data from one input/output device to another (such as from card to tape, tape to tape, tape to disk, and so on).

3-109. SORT-MERGE PROGRAMS arrange data records in a predefined sequence or order and are capable of combining two or more ordered files into one file.

3-110. REPORT PROGRAM GENERATORS are used to generate programs to print detail and summary reports of data files.

3-111. PROGRAMMING LANGUAGES are the means by which human-to-computer communication is achieved. They are used to write the instructions to tell the computer what to do to solve a given problem.

3-112. A MACHINE LANGUAGE uses a string of numbers that represent the instruction codes and operand addresses to tell the computer what to do.

3-113. SYMBOLIC LANGUAGES improves the program preparation process by substituting letter symbols (mnemonic codes) for basic machine language instruction codes.

3-114. A PROCEDURE ORIENTED LANGUAGE is a programming language oriented toward a specific class of processing problems. Examples are BASIC, COBOL, and FORTRAN.

3-115. PROGRAMMING is the process of planning and coding the computer instructions to solve a problem.

3-116. FLOWCHARTING is one method of pictorially representing a procedural (step-by-step) solution to a problem before actually starting to write the computer instructions required to produce the desired results.

3-117. PACKAGED SOFTWARE is designed for specific classes of applications. Examples are word processing, spreadsheets, data management, and graphics. The manufacturer, a software house, or a central design agency usually writes these off-the-shelf programs.

Check-on-Learning Questions

1. What is the heart of any computer system?
2. Which types of operating systems are the simplest and most common on microcomputers?
3. What types of operating systems lets more than one user access the same resources at the same time?
4. Why is the availability of applications software for a particular operating system critical?
5. How is the applications software compatibility problem overcome?
6. What is the name of the record that takes control of the PC and loads the IO.SYS into RAM?
7. What is the simplest way to upgrade an operating system?
8. What programs eliminate the need for programmers to write new programs when all they want to do is copy, print, or sort a data file?
9. How do we tell a utility program what we want it to do?
10. What is the term given to arranging data records in a predefined sequence or order?
11. What must the sort program be told to do in order to sort a data file?
12. For what purpose are report program generators used?
13. With early computers, the programmer had to translate instructions into what type of language form?
14. When were mnemonic instruction codes and symbolic addresses developed?
15. Why were procedure-oriented languages developed?
16. What computer language was developed for mathematical work?
17. What are two disadvantages of procedure-oriented languages?
18. What is programming?
19. In programming, how many steps are involved in solving a problem on a computer?
20. What is required before starting to write or code a program?
21. What are the two types of flowcharts?
22. What type of flowchart does the programmer construct to represent the sequence of operations the computer will perform to solve a specific problem?
23. How many tools are used in flowcharting?
24. Is there a "best way" to construct a flowchart?
25. What is the first thing that needs to be known before developing a flowchart?
26. What is the fundamental element in program preparation?
27. What types of instruction permit addition, subtraction, multiplication, and division?
28. What must be done to the program after the coding is completed?
29. How do we refer to errors caused by faulty logic and coding mistakes?

30. What is the purpose of testing a program?
31. When does documentation begin?
32. What is packaged software?
33. What does data management software allow you to do?
34. What are spreadsheets?
35. Are all printers capable of handling graphics output?

Chapter 4

Data Representation and Communications

INTRODUCTION

4-1. One of the major problems faced today with digital computers is communication. Chapter 2 covered the several types of input devices that read data into a computer. However, there is more than one way of getting data into the computer to be processed. In order to process data you must determine the following:

- How to prepare the data to be used as input.
- How to convert human-readable documents into a computer-readable form.
- What type of input media to use.
- How to transmit data that will be used by another computer some distance away.

There are several ways to perform this conversion and transmission process.

DATA

4-2. Data is a general term used to describe raw facts. Data is nothing more than a collection of related elements or items that, when properly coded into some type of input medium, can be processed by a computer. Data items might include service number, name, pay grade, or any other fact. Until some meaning is given to determine what the data is about, it will remain just data. When this data has been processed together with other facts, it then has meaning and becomes information that could be understood and properly used.

DATA REPRESENTATION

4-3. Data is represented by symbols. Symbols convey meaning only when understood. The symbol is just a representation of the information and may convey one meaning to some, another meaning to others, and no meaning at all to those that do not know their significance. Data must be reduced to a set of symbols that the computer can read and interpret before there can be any communication with the computer (see Figure 4-1). The first computers were designed to manipulate numbers to solve arithmetic problems. Many other symbols can be created, used, and manipulated to represent facts in the world. Manipulating these symbols is possible if an identifying code or coded number is assigned to the symbol to be stored and processed. The letters in a name such as "ALBERT" or "CAROL" can be represented by different codes, as can all special characters (such as #, &, \$, and @). The data represented is called source data.

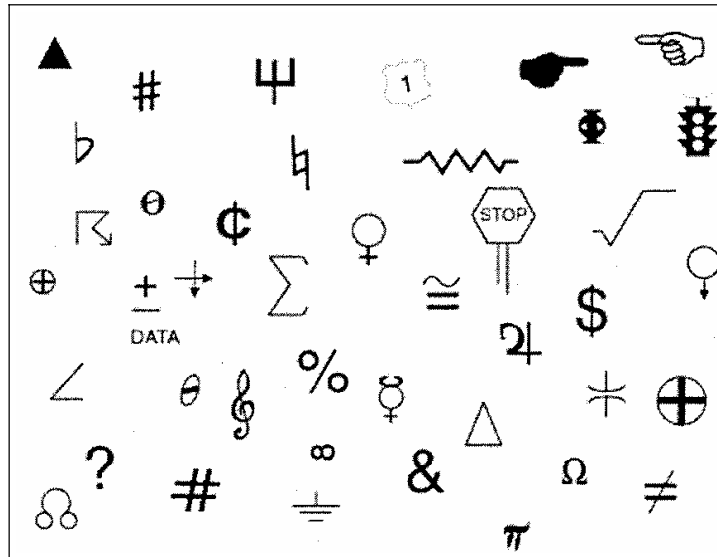


Figure 4-1. Communications Symbols

SOURCE DATA

4-4. Source data or raw data is typically written on some type of paper document, which is referred to as a source document. The data contained on the source document must be converted into a machine-readable form for processing either by direct or indirect means. The data may be entered directly into the computer in its original form (such as, right from the source document on which it is recorded by way of magnetic ink characters, optically recognizable characters, or bar code recognition). Or the data on the documents may be entered indirectly on input media (such as floppy disk, OCR, CD-ROM disk, and zip disk). It may also be keyed directly into a computer from a keyboard.

4-5. In 1973, the Uniform Grocery Code Council recommended the adoption of the Universal Product Code (UPC) symbol. This symbol is still in use in the United States (US) today. In 1981, the United States Department of Defense adopted the use of Code 39 for marking all products sold to the US military with a system called Logistics Marking System (LOGMARS). The complete UPC bar code number consists of the UPC bar code prefix and a unique product identification (ID) number that the supplier randomly assigns. The UPC bar code symbol is created by inputting the UPC bar code number into the UPC bar code software. The UPC bar code software will then generate the UPC bar code symbol that is printed on the retail product packaging. The UPC bar code number (and associated bar code symbol) consists of the manufacturer number, which is combined with a product number (assigned by the manufacturer) and a check digit (assigned by a mathematical equation found in most bar code software). The UPC symbol is consists of a row of 59 black and white bars. Printed beneath the bars is a series of 12 numbers. In the UPC bar code sample in Figure 4-2, a 6 digit number, "012345" has been assigned (leaving 5 digits to represent items - plus one 'check digit'). This initial number "012345" represents the manufacturer on all of their products as well as in any Electronic Data Interchange (EDI) applications. For example, the UPC prefix code for the Coca-Cola Company is 049000 (this prefix was assigned and licensed to the

Coca-Cola Company from the Uniform Code Council [UCC]). Therefore, this 049000 number will appear in the first 6 digits of the Coca-Cola Company's entire product UPC bar codes. The 5 numbers that follow, identify the specific product, and its size, color, flavor, and so on (depending on type of product). The last number (in Figure 4-2) is called the 'check digit' and is used to guard against errors (when numbers are manually keyed in) and fraud. There is a mathematical formula that, when applied, produces the proper check digit. The UPC bar code (see Figure 4-2), aside from speeding checkout, allows retailers and manufacturers to manage and replenish inventory more efficiently. The bar code also automates many processes and operations, like special promotions, coupons, and product returns.



Figure 4-2. UPC Bar Code

COMPUTER CODING SYSTEMS

4-6. Some sort of coding system must be used to represent numeric, alphabetic, and special characters in a computer's internal storage and on magnetic media. In computers, the code is made up of fixed size groups of binary positions. Each binary position in a group is assigned a specific value (for example, 8, 4, 2, or 1). In this way, every character can be represented by a combination of bits that is different from any other combination.

4-7. There are a number of selected coding systems that are used to represent data. The coding systems that will be covered are EBCDIC and ASCII. Regardless of what coding system is used, each character will have an additional bit called a check bit or parity bit.

EXTENDED BINARY CODED DECIMAL INTERCHANGE CODE

4-8. Using an 8-bit code, it is possible to represent 256 different characters or bit combinations. This provides a unique code for each decimal value 0 through 9 (for a total of 10), each uppercase and lowercase letter (for a total of 52), and for a variety of special characters. In addition to four numeric bits, four zone bit positions are used in 8-bit code (see Figure 4-3). Each group of the eight bits makes up one alphabetic, numeric, or special character and is called a byte.

ZONE BITS				NUMERIC BITS			
Z/ 8	Z/ 4	Z/ 2	Z/ 1	8	4	2	1

Figure 4-3. Format For EBCDIC and ASCII Codes

4-9. Figure 4-3 shows that the four rightmost bits in EBCDIC are assigned values of 8, 4, 2, and 1. The next four bits to the left are called the zone bits. Figure 4-4 shows the EBCDIC coding chart for uppercase and lowercase alphabetic characters and for the numeric digits 0 through 9 (along with their hexadecimal equivalents). Hexadecimal is a number system used with some computer systems. It has a base of 16 (0-9 and A-F). A represents 10; B represents 11; C represents 12; D represents 13; E represents 14; and F represents 15. In EBCDIC, the bit pattern 1100 is the zone combination used for the alphabetic characters A through I, 1101 is used for the characters J through R, and 1110 is the zone combination used for characters S through Z. The bit pattern 1111 is the zone combination used when representing decimal digits. For example, the code 11000001 is equivalent to the letter A; the code 11110001 is equivalent to the decimal digit 1. Other zone combinations are used when forming special characters. Not all of the 256 combinations of 8-bit code have been assigned characters. Figure 4-5 shows how the characters DP-3 are represented using EBCDIC.

4-10. Since one numeric character can be represented and stored using only four bits (8-4-2-1), using an 8-bit code allows the representation of two numeric characters (decimal digits) (see Figure 4-6). Representing two numeric characters in one byte (eight bits) is referred to as packing or packed data. By packing data (numeric characters only) in this way, it allows us to conserve the amount of storage space required, and at the same time, increases processing speed.

ALPHABETIC CHARACTERS					
UPPERCASE			LOWERCASE		
PRINTS AS	EBCDIC		PRINTS AS	EBCDIC	
	IN BINARY	IN HEXA-DECIMAL		IN BINARY	IN HEXA-DECIMAL
	ZZZZ 8421			ZZZZ 8421	
A	1100 0001	C 1	a	1000 0001	8 1
B	1100 0010	C 2	b	1000 0010	8 2
C	1100 0011	C 3	c	1000 0011	8 3
D	1100 0100	C 4	d	1000 0100	8 4
E	1100 0101	C 5	e	1000 0101	8 5
F	1100 0110	C 6	f	1000 0110	8 6
G	1100 0111	C 7	g	1000 0111	8 7
H	1100 1000	C 8	h	1000 1000	8 8
I	1100 1001	C 9	i	1000 1001	8 9
J	1101 0001	D 1	j	1001 0001	9 1
K	1101 0010	D 2	k	1001 0010	9 2
L	1101 0011	D 3	l	1001 0011	9 3
M	1101 0100	D 4	m	1001 0100	9 4
N	1101 0101	D 5	n	1001 0101	9 5
O	1101 0110	D 6	o	1001 0110	9 6
P	1101 0111	D 7	p	1001 0111	9 7
Q	1101 1000	D 8	q	1001 1000	9 8
R	1101 1001	D 9	r	1001 1001	9 9
S	1110 0010	E 2	s	1010 0010	A 2
T	1110 0011	E 3	t	1010 0011	A 3
U	1110 0100	E 4	u	1010 0100	A 4
V	1110 0101	E 5	v	1010 0101	A 5
W	1110 0110	E 6	w	1010 0110	A 6
X	1110 0111	E 7	x	1010 0111	A 7
Y	1110 1000	E 8	y	1010 1000	A 8
Z	1110 1001	E 9	z	1010 1001	A 9
NUMERIC CHARACTERS					
0	1111 0000	F 0	5	1111 0101	F 5
1	1111 0001	F 1	6	1111 0110	F 6
2	1111 0010	F 2	7	1111 0111	F 7
3	1111 0011	F 3	8	1111 1000	F 8
4	1111 0100	F 4	9	1111 1001	F 9

Figure 4-4. Eight-Bit EBCDIC Coding Chart (Including Hexadecimal Equivalents)

D	P	-	3
1100 0100	1101 0111	0110 0000	1111 0011

Figure 4-5. DP-3 Represented Using 8-Bit EBCDIC Code

DECIMAL VALUE	9	2	7	3
EBCDIC CODE	1001	0010	0111	0011
BIT PLACE VALUES	8421	8421	8421	8421
	BYTE 1		BYTE 2	

Figure 4-6. Packed Data

AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE

4-11. Another 8-bit code known as ASCII was originally designed as a 7-bit code. Several computer manufacturers cooperated to develop this code for transmitting and processing data. The purpose was to standardize a binary code to give the computer user the capability of using several machines to process data regardless of the manufacturer (such as IBM, Honeywell, UNIVAC, Burroughs, and so on). However, since most computers are designed to handle (store and manipulate) 8-bit code, an 8-bit version of ASCII was developed. ASCII is commonly used in the transmission of data through data communications and is used almost exclusively to represent data internally in microcomputers.

4-12. The concepts and advantages of ASCII are identical to those of EBCDIC. The important difference between the two coding systems lies in the 8-bit combinations assigned to represent the various alphabetic, numeric, and special characters. Notice that the selection of bit patterns, when using ASCII 8-bit code, are different from those when using EBCDIC. Table 4-1 shows a comparison of the characters DP-3 in EBCDIC and ASCII.

Table 4-1. DP-3 Characters in EBCDIC and ASCII

Character	D	P	3
EBCDIC	1100 0100	1101 0111	1111 0011
ASCII	0100 0100	0101 0000	0011 0011

4-13. In ASCII, rather than breaking letters into three groups, uppercase letters are assigned codes beginning with hexadecimal value 41 and continuing sequentially through hexadecimal value 5A. Lowercase letters are also assigned hexadecimal values of 61 through 7A. The decimal values 1 through 9 are assigned the zone code 0011 in ASCII rather than 1111 as in EBCDIC. Figure 4-7 is the ASCII coding chart showing uppercase and lowercase alphabetic characters and numeric digits 0 through 9.

ALPHABETIC CHARACTERS							
UPPERCASE				LOWERCASE			
PRINTS AS	ASCII CODE		IN HEXA-DECIMAL	PRINTS AS	ASCII CODE		
	IN BINARY				IN BINARY	IN HEXA-DECIMAL	
	8421	8421			8421	8421	
A	0100	0001	4 1	a	0110	0001	5 1
B	0100	0010	4 2	b	0110	0010	5 2
C	0100	0011	4 3	c	0110	0011	5 3
D	0100	0100	4 4	d	0110	0100	5 4
E	0100	0101	4 5	e	0110	0101	5 5
F	0100	0110	4 6	f	0110	0110	5 6
G	0100	0111	4 7	g	0110	0111	5 7
H	0100	1000	4 8	h	0110	1000	5 8
I	0100	1001	4 9	i	0110	1001	5 9
J	0100	1010	4 A	j	0110	1010	5 A
K	0100	1011	4 B	k	0110	1011	5 B
L	0100	1100	4 C	l	0110	1100	5 C
M	0100	1101	4 D	m	0110	1101	5 D
N	0100	1110	4 E	n	0110	1110	5 E
O	0100	1111	4 F	o	0110	1111	5 F
P	0101	0000	6 0	p	0111	0000	7 0
Q	0101	0001	6 1	q	0111	0001	7 1
R	0101	0010	6 2	r	0111	0010	7 2
S	0101	0011	6 3	s	0111	0011	7 3
T	0101	0100	6 4	t	0111	0100	7 4
U	0101	0101	6 5	u	0111	0101	7 5
V	0101	0110	6 6	v	0111	0110	7 6
W	0101	0111	6 7	w	0111	0111	7 7
X	0101	1000	6 8	x	0111	1000	7 8
Y	0101	1001	6 9	y	0111	1001	7 9
Z	0101	1010	6 A	z	0111	1010	7 A

NUMERIC CHARACTERS							
0	0011	0000	8 0	5	0011	0101	8 5
1	0011	0001	8 1	6	0011	0110	8 6
2	0011	0010	8 2	7	0011	0111	8 7
3	0011	0011	8 3	8	0011	1000	8 8
4	0011	0100	8 4	9	0011	1001	8 9

Figure 4-7. Eight-Bit ASCII Coding Chart (Including Hexadecimal Equivalents)

PARITY BIT

4-14. This additional bit in each storage location is used to detect errors in the circuitry. Therefore, a computer that uses an 8-bit code (such as EBCDIC or ASCII) will have a ninth bit for parity checking.

4-15. The parity bit provides an internal means for checking the validity, the correctness, of code construction. This means that the total number of bits in a character, including the parity bit, must always be odd or always be even, depending upon whether the particular computer system or device being used is odd or even parity. Therefore, the coding is said to be in either odd or even parity code and the test for bit count is called a parity check.

4-16. The following covers bits and bytes, primary storage, and storage capacities. Also covered are the ways data may be stored and retrieved inside the computer.

DATA STORAGE CONCEPTS

4-17. Chapter 2 covered that a computer's primary storage area, each serving a specific purpose, is divided into the following four areas:

- Input storage area accepts and holds input data to be processed.
- Working storage area holds intermediate processing results.
- Output storage area holds the final processing results.
- Program storage area holds the processing instructions (the program).

Remember, each of these separate areas do not have built-in physical boundaries. Instead, the boundaries are determined by the individual programs being used.

BITS AND BYTES

4-18. A bit is a single binary digit that represents the smallest unit of data. However, computers usually do not operate on single bits. Instead, they store and manipulate a fixed number of bits. The smallest unit or number of bits a computer works with is most often eight bits. These eight bits make up a byte. EBCDIC and ASCII codes use eight bits (excluding the parity bit), and that eight bits represent a single character (such as the letter A or the number 7). Therefore, the computer can store and manipulate an individual byte (a single character) or a group of bytes (several characters or a word) at a time. These individual bytes, or groups of bytes, form the basic unit of memory.

4-19. Primary storage capacities are usually specified in number of bytes. The symbol "K" is used whenever reference is made to the size of memory, especially when the memory is quite large. The symbol K is equal to 1,024 units or positions of storage. Therefore, if a computer has 512K bytes (not bits) of primary storage, then it can hold $512 \times 1,024$ or 524,288 characters (bytes) of data in its memory.

MAGNETIC CORE STORAGE

4-20. In primary storage, many magnetic cores are strung together on a screen of wire to form what is called a core plane (see Figure 4-8). Each core can store one binary bit (0 or 1) of data. Current flowing through the wires on which the core is strung magnetizes a core. Therefore, a core magnetized in one direction represents a binary 0 and when magnetized in the opposite direction, a binary 1. It is the direction in which the core is magnetized that determines whether it contains a binary 0 or a binary 1 (see Figure 4-9). These core planes look very much like small window screens and are arranged vertically to represent data (see Figure 4-10). Figure 4-10 shows the nine planes needed to code in 8-bit EBCDIC. The ninth plane provides for a parity (check) bit. Figure 4-10 also shows DP-3 in EBCDIC code, even parity.

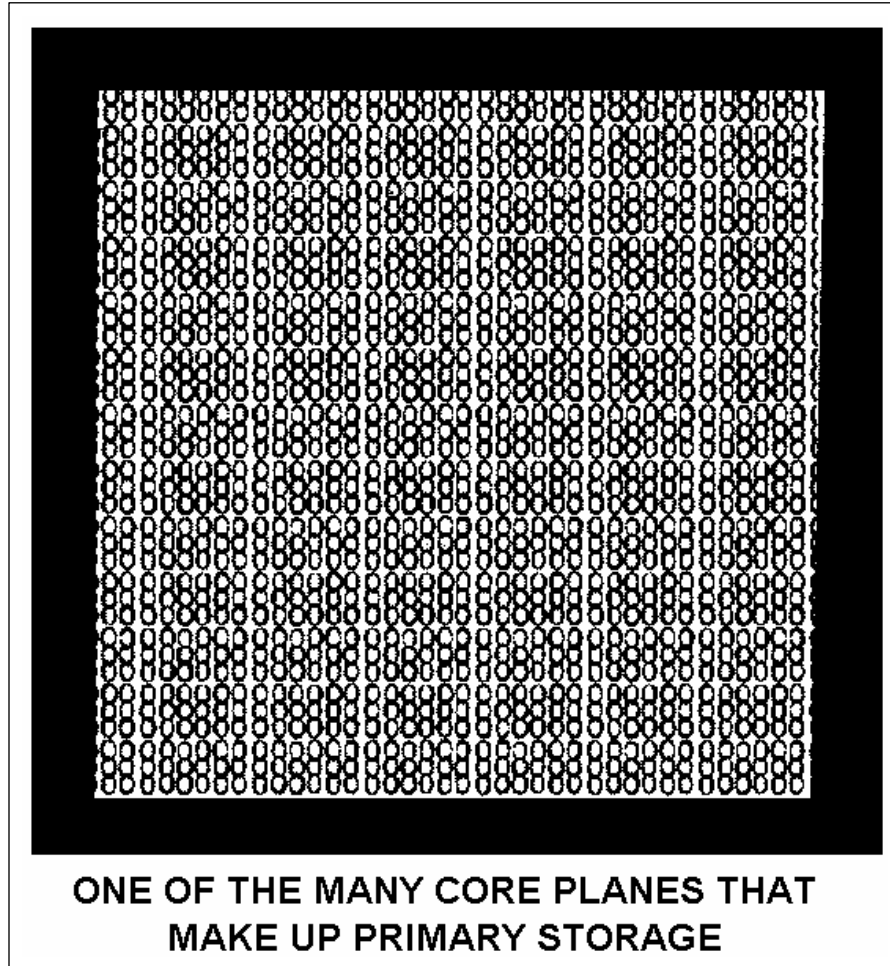


Figure 4-8. Core Plane

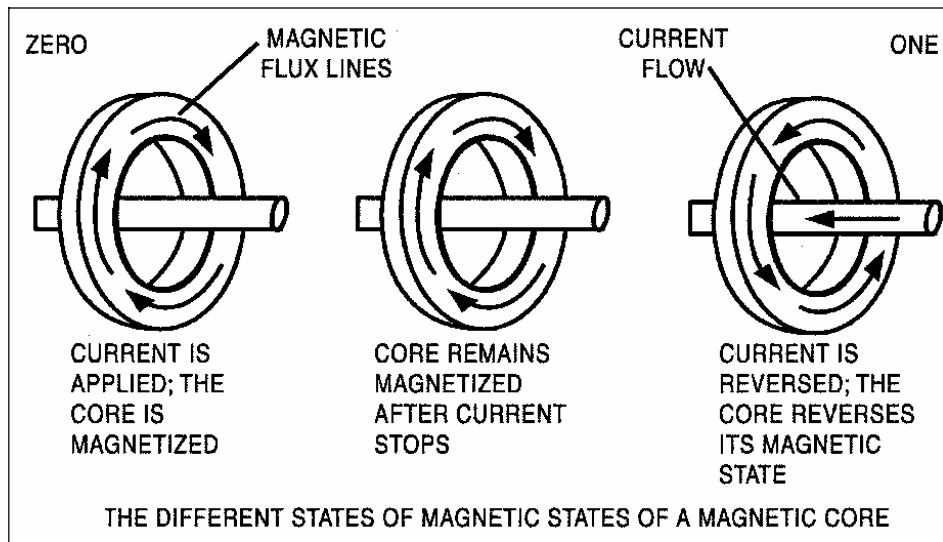


Figure 4-9. Core Magnetized in One Direction

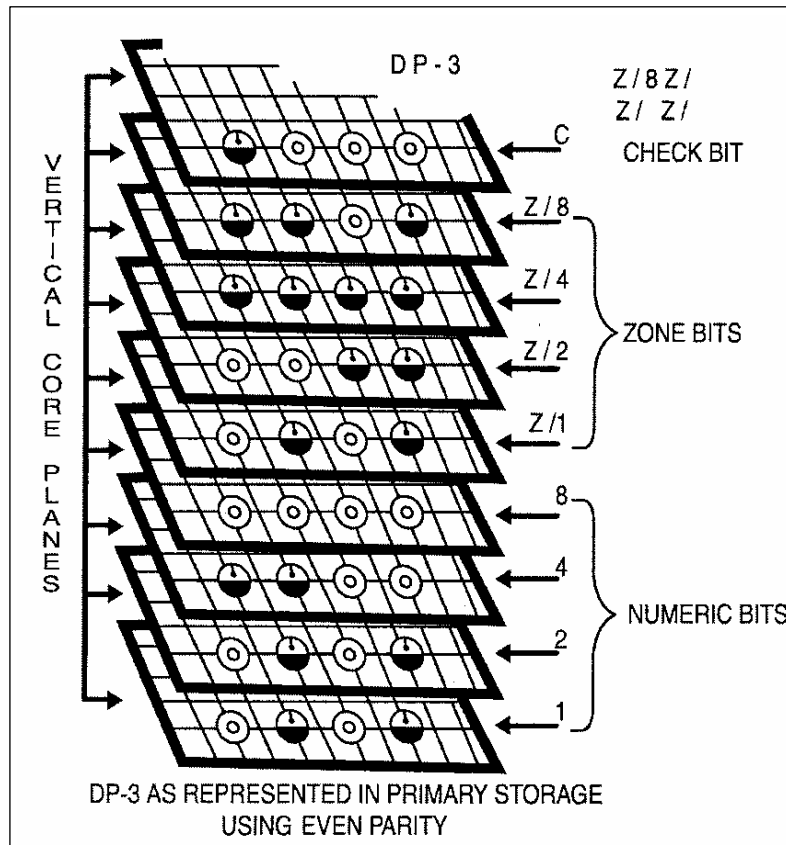


Figure 4-10. Core Planes Arranged Vertically to Represent Data

STORAGE CAPACITY AND ADDRESSES

4-21. The storage capacity of an address is designed and built into the computer by the manufacturer. Over the years, several different design approaches to partition primary storage have been used. With this in mind, let us take a look at some of the ways primary storage is partitioned into addresses.

4-22. One way to design or organize the primary storage section is to store a fixed number of characters (bytes) at each address location. These characters can be referenced as a single entity called a word (see Figure 4-11). The name "CHARLIE" (address location 400) or the amount he is paid, in this case \$69.00 (address location 401), are each treated as a single word. Computers that are built to retrieve, manipulate, and store a fixed number of characters in each address are said to be word-oriented, word-addressable machines, or fixed-word-length computers.

4-23. Another way to design the primary storage section is to store a single character, such as the letter L or the number 8, in each address location. An address is assigned to each location in storage. Computers designed in this way are said to be character-oriented or character addressable. They are also called variable-word-length computers. Therefore, the name "CHARLIE" (see Figure 4-12) now requires seven address locations (300 through 306), while the amount paid (\$69.00) occupies six address locations (307 through 312).

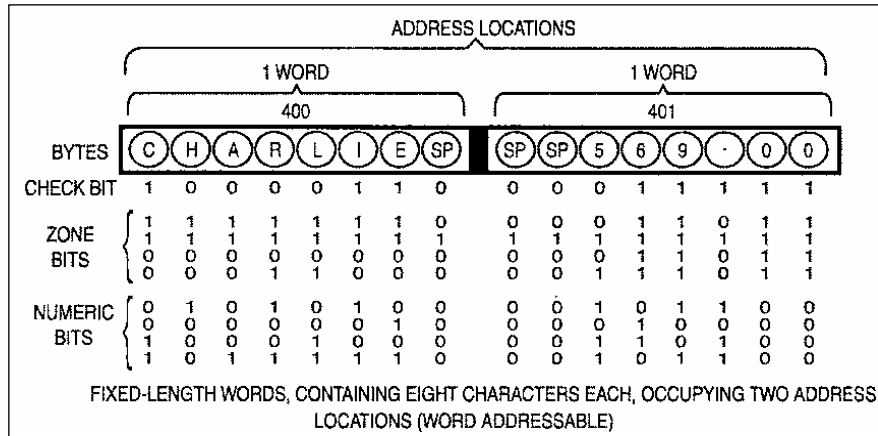


Figure 4-11. Fixed-Word-Length Versus Variable-Word-Length Storage, Fixed-Length Words, Containing Eight Characters Each, Occupying Two Address Locations (Word Addressable)

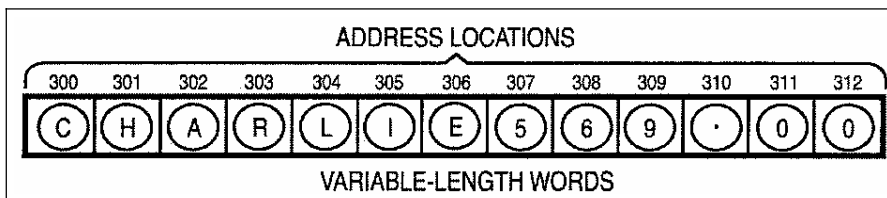


Figure 4-12. Fixed-Word-Length Versus Variable-Word-Length Storage, Variable-Length Words (Character Addressable)

4-24. Whether a computer addresses a group of bytes as a word or addresses each byte individually is a function of the circuitry. Both designs have advantages and disadvantages. Variable-word-length computers make the most efficient use of available storage space, since a character can be placed in every storage location. In a fixed-word-length computer, storage space may be wasted. For example, if the storage capacity in each address of a fixed-word-length computer is eight bytes, and some of the data elements to be stored contain only three or four characters, then many of the storage positions in each word are not being used.

4-25. Fixed-word-length computers have faster calculating speeds since they can add two data words in a single operation. This is not so with character-addressable computers since they can only add one digit (byte) in each number during a single machine operation. Therefore, eight steps are required to complete the calculation.

4-26. The larger mainframe computers (super-computers like the CRAY-1 and CYBER 205) use only fixed-word-length storage. Most microcomputers use the variable-word-length approach, allowing them to operate on one character at a time. Somewhere in between these two extremes are the dozens of existing minicomputer and mainframe models that have what is called built-in flexibility.

4-27. These flexible computers are byte-oriented but can operate in either a fixed- or variable-word-length mode through the use of proper program instructions. Let us take a look at how these flexible computers operate in a variable- and fixed-word-length environment.

4-28. Working in a variable-word-length environment, each address holds one alphanumeric character (see Figure 4-12). Since a byte usually represents a single alphanumeric character, unless packed data is being used, a flexible computer is often said to be byte-addressable. Do not be confused; the terms “character-addressable,” “character-oriented,” and “byte-addressable” all have the same meaning. By using the appropriate program instructions, a programmer can retrieve a stored data element by identifying the address of the first character (say, position 300 as in Figure 4-12) and specifying the number of address locations to be included in the word. In this case there are seven locations—positions 300 through 306.

4-29. When a flexible computer is working in a fixed-word-length environment, each address identifies a group of bytes that can be operated on as a unit. This processing method helps to achieve faster calculating speeds. A programmer can use program instructions to cause the computer to automatically retrieve, manipulate, and store, as a unit, a fixed word of say, two, four, or eight bytes of data in one machine operation by identifying the address of the first character of data. At the same time, all remaining bytes are acted upon as a unit moving from left to right. Figure 4-13 shows the different word lengths possible with many byte-addressable computers. They are half-word (2 bytes), full-word (4 bytes), and double-word (8 bytes). The next step is to see how these bits and bytes are represented (coded) on some of the more common secondary storage media.

SECONDARY STORAGE DATA ORGANIZATION

4-30. Secondary storage devices (also called auxiliary or mass storage devices) are those devices that are not part of the CPU. They include external core, semiconductor, thin film, and several different types of mass storage such as RAID and CD-ROM disk.

4-31. Remember, it takes a certain number of bits to make one byte (normally eight), and when bytes are grouped together at a single address they make up a word in the computer's memory. When data is recorded on some type of magnetic storage medium (such as disk or tape); it is normally organized by bits, characters (bytes), fields, records, and files (see Figure 4-14). The following definitions will help in understanding the relationship between bits, characters, bytes, words, fields, records, and files:

- **BIT** - The smallest unit of data; it represents one binary digit (0 or 1).
- **CHARACTER (BYTE)** - A group of related bits (usually eight) that make up a single character-letter, number, or special character.
- **WORD** - A group of related bytes that are treated as a single addressable unit or entity in memory.

- **FIELD** - One or more related characters that are treated as a unit of information. A field (also referred to as a data item) may be alphabetic, numeric, or alphanumeric, and may be either fixed or variable in length. For example, a SSN is of a fixed length; that is, it is always 9 positions in length. However, names are variable length because they may be from 2 to 25 positions in length.
- **RECORD** - A group of related fields, all pertaining to the same subject (a person, a thing, or an event). For example, a payroll record (LES) might include fields for a name, amount paid, taxes withheld, earned leave, and any allotments that are deducted. On the other hand, a supply inventory record might consist of fields containing the stock number, the name of the item, its unit price, the quantity on hand, and its bin location.
- **FILE** - A collection of related records such as the payroll or supply inventory records. Normally, all records within the file are in the same format.

4-32. Processing data is thought of in terms of data files. For example, to process a parts inventory, a master parts inventory file and the file that contains up-to-date information on each part that has been issued is needed. The master parts inventory file would have a record for every part in the inventory. The update file and the parts issued file would have a record for each part issued. A program is used to read the records on the parts issued file and to update the matching records on the master parts inventory file. Depending on whether the data is stored on magnetic tape or disk or in internal storage, the program would use different methods to access storage to obtain the data.

STORAGE ACCESS METHODS

4-33. How data files are stored in secondary storage varies with the types of media and devices being used. Data files may be stored on or in sequential-access storage, direct-access storage, or random-access storage.

SEQUENTIAL-ACCESS STORAGE

4-34. When operating in a sequential environment, a particular record can be read only by first reading all the records that come before it in the file. This method has now been replaced by other more modern, not sequential, methods (such as RAID and hard drive).

DIRECT-ACCESS STORAGE

4-35. Direct-access storage allows access to data directly from the media without first having to read the front data. Floppy disks and CD-ROM disks are examples of direct-access storage media. Data can be obtained quickly from anywhere on the media. However, the amount of time it takes to access a record is dependent to some extent on the mechanical process involved. It is usually necessary to scan some (but not all) of the preceding data.

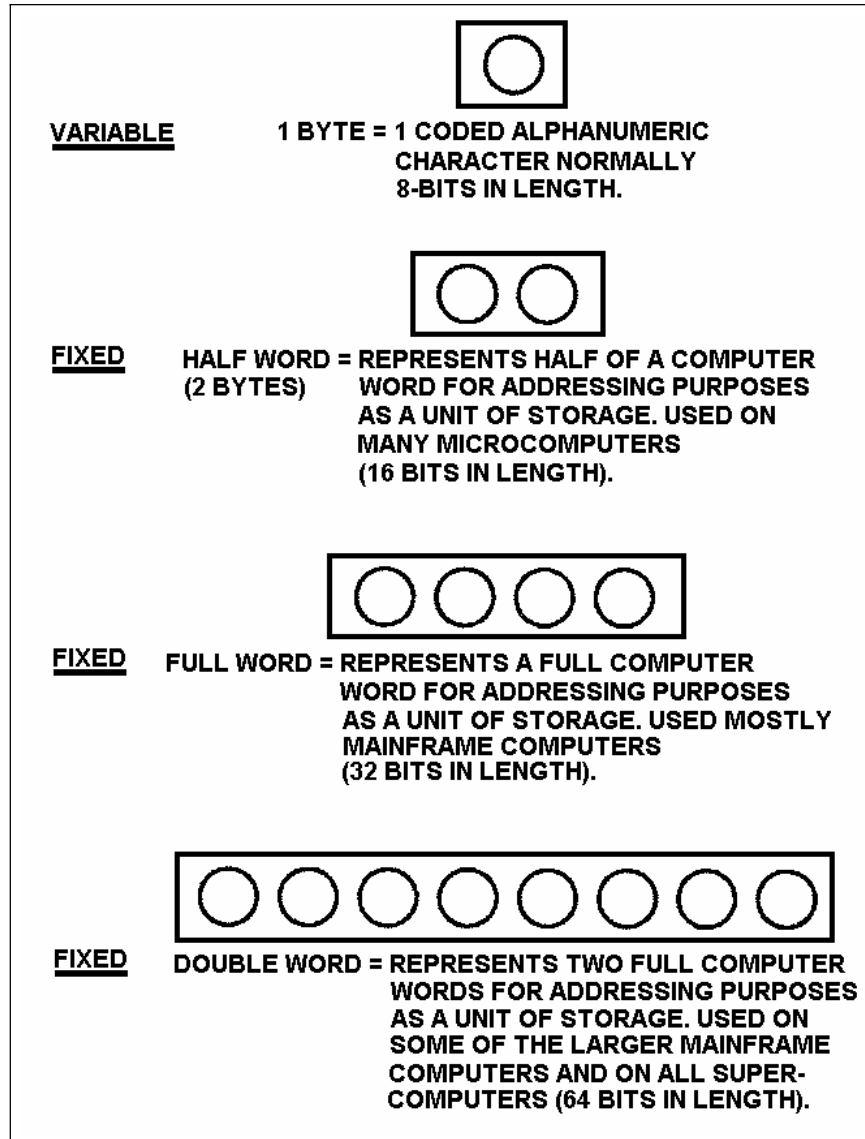


Figure 4-13. Word Lengths Used On Flexible Byte-Addressable Computers

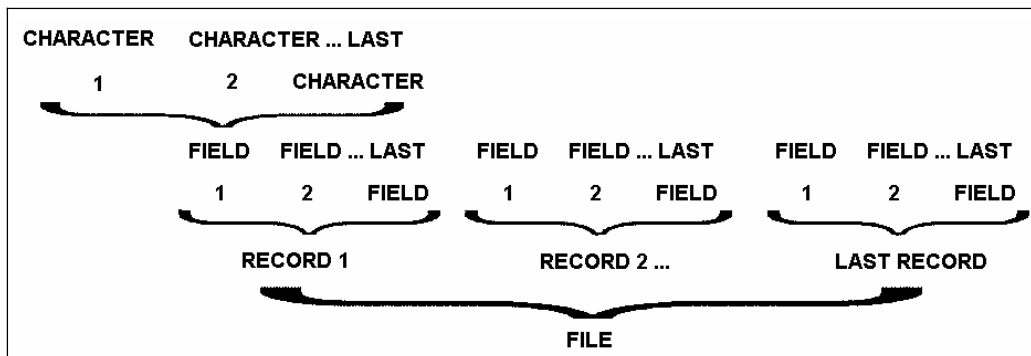


Figure 4-14. Data Organization

RANDOM-ACCESS STORAGE

4-36. Random-access storage media refers to magnetic core, semiconductor, and thin film. In this type of storage, a given item of data can be selected from anywhere in storage without having to scan any preceding items and the access time is independent of the storage location.

NETWORKS

4-37. A network can be defined as any system composed of one or more computers and terminals. However, most are composed of multiple terminals and computers. The below explains how networks allow dissimilar computers to work together as a team.

LOCAL AREA NETWORKS

4-38. In LANs, various machines are linked together within a building or adjacent buildings. Figure 4-15 shows an example of a LAN. A LAN allows dissimilar machines to exchange information within one universal system. With the ability to communicate, the dissimilar machines act as a team. The information that exists in one system can be reused without being reentered via keyboard or disk into another separate system. A universal system for the integration and exchange of information is connected to all input devices. The entire system is usually housed within the same building or the same geographic area. A local area network is made up of a communications facility (for example, a coaxial cable, such as that used for cable television) and interface units creating a link for the computers and terminals to the communications facility. The two designs that can be used are broadband or baseband.

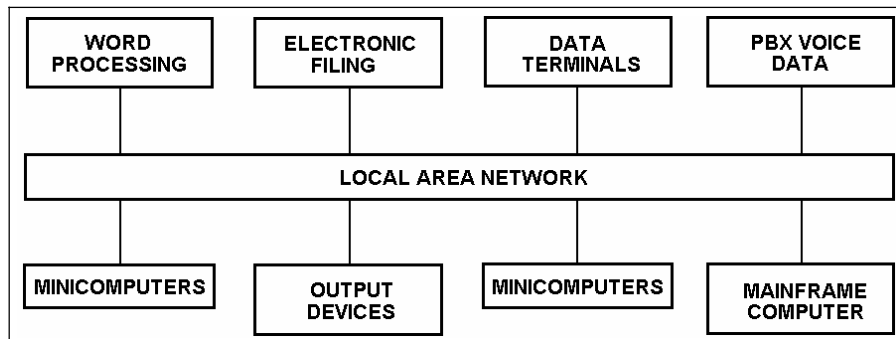


Figure 4-15. Local Area Network System

4-39. A baseband communications channel uses the basic frequency band of radio waves and a coaxial cable. This coaxial cable has one channel, which is like a party line. Only two machines can use this cable at one time, even though many have the channel available. There is no central switching unit to route traffic over the network.

4-40. A more expensive channel, called a broadband communications channel, can handle more advanced applications. This includes transmission of voice as well as data and text. Since the controller can be used to route traffic for a large number of simultaneous users, the users are able to share one of the many individual channels of the system.

WIDE AREA NETWORKS

4-41. Wide area networks provide for global connections and are sometimes referred to as global networks. Organizations are able to send information from city to city, across the nation, and to other countries throughout the world through the expansion of local area networks into larger network configurations. Larger telecommunications networks use combinations of telephone lines, microwave radio links, and satellites to send information. In 1965, the first successful communications satellite for business applications was launched after many tries with primitive satellites. With the launching of larger and more complex satellites, the size and complexity of earth stations have been shrinking. Since satellite service costs have been steadily decreasing, it is becoming more cost-effective to use them for business-type uses.

MODEMS

4-42. Since both signals and data can be transmitted and received through cables (communications lines), they are referred to as input/output channels. When data is transmitted directly to a computer over long distances, it becomes necessary to add two other devices, one at each end of the communications line. These devices are called modems (see Figure 4-16). The word “modem” is an acronym for modulator/demodulator (combines first syllable of each word). A modem converts the digital signal produced by a terminal or the computer to an audio signal suitable for transmission over the communications line. The modem at the other end of the line converts the audio signal back to a digital signal before it is supplied to the computers or a terminal. If this conversion were not carried out, the digital signal would degenerate during transmission and become garbled.

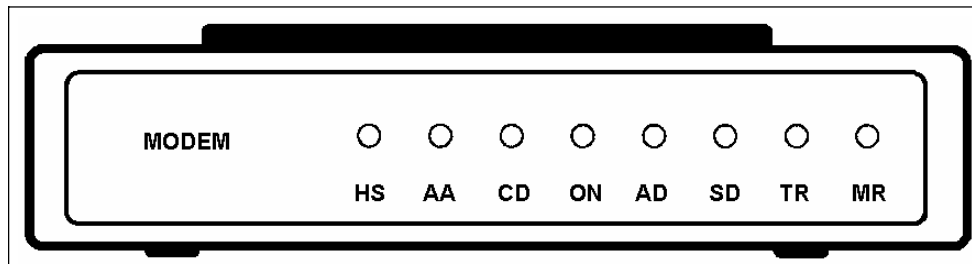


Figure 4-16. Modem

4-43. The physical link or medium that is used to carry (or transmit) data from one location to another is a communications channel. It allows remotely located input/output devices to communicate directly with the computer's CPU. Telephone lines (often referred to as land lines) are a frequently used type of communications channel.

4-44. In a simple data communications system, terminals and other remote I/O devices are linked directly to one or more CPUs to allow users to enter data and programs and receive output information. Interface elements (those devices, such as modems, that serve to interconnect) are used to bridge and control the different data communications environments. Modems permit the system to switch back and forth from computer digital data to analog signals that can be transmitted over communications lines.

4-45. A modem never knows exactly when to expect data. Therefore, it must receive some type of signal warning that data is about to be transmitted. This gives the modem time to get itself aligned and in synchronization with the incoming signal. Special characters, known as message characters, provide this warning and are placed in front of and behind the data to mark the beginning and ending of the message. The two methods of transmission used are called asynchronous and synchronous.

4-46. With asynchronous transmission, each character of data must be surrounded by message characters. As a result, more total bits must be transmitted (transferred) than would be necessary if the synchronous method were used.

4-47. With synchronous transmission, only a single set of start and stop message characters is needed per block of data. This allows more characters to be transmitted per second. Synchronous transmission is more efficient and faster. However, it has the disadvantage of requiring a more complex and expensive modem than does the asynchronous transmission.

4-48. Whenever data is transferred between devices, it also involves an exchange of prearranged signals. This is known as “handshaking”. These signals, in combination with a prearranged pattern of message characters, define the rules for exchanging data over a communications line. The exact rules depend upon each individual computer manufacturer, the telephone company, and the related devices (modems) that make up the computer system. “Protocol” is the term used for the specific set of rules that govern handshaking and message characters.

4-49. In a network system (see Figure 4-17), data to be sent to the main computer's CPU is entered through a remote online user terminal (far left). The data is keyed in digital form and sent to a nearby modem for conversion into an analog signal suitable for transmission. This converted data is then transmitted over the telephone (or land) lines to another modem located near the main computer system's CPU. The data, now in digital form, can be sent directly to the CPU for processing. The same route is followed when information is sent from the CPU back to the remote terminal. Data communications and networks expand the use of computer technology by providing a means for computers and other machines to talk to each other.

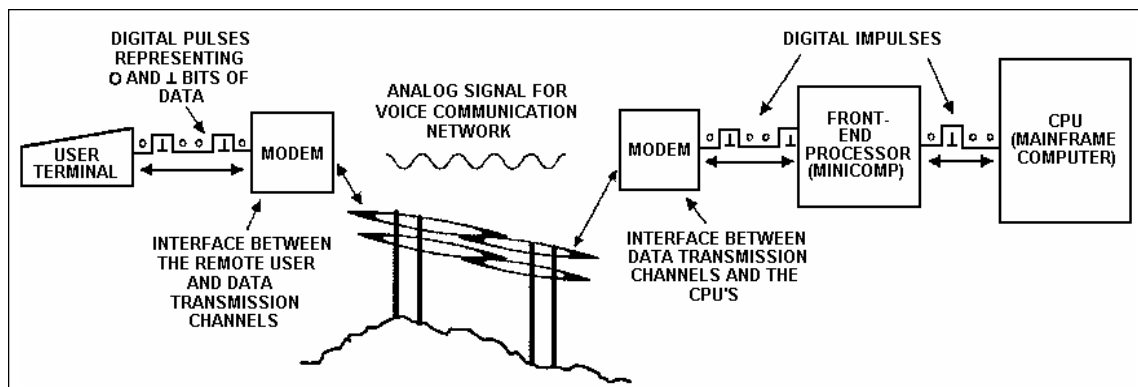


Figure 4-17. Modems Used In Network System

SUMMARY

- 4-50. This chapter has presented information on data. The information that follows summarizes the important points in this chapter.
- 4-51. DATA is a general term used to describe raw facts like a service number, name, and paygrade.
- 4-52. DATA REPRESENTATION is accomplished by the use of symbols. The symbol itself is not the information, but it is merely a representation. Symbols convey meaning only when understood. In computers, symbols are represented by CODES.
- 4-53. SOURCE DATA is raw data typically written on some type of paper document.
- 4-54. COMPUTER CODING SYSTEMS are used to represent numeric, alphabetic, and special characters in computer storage and on magnetic media.
- 4-55. EXTENDED BINARY CODED DECIMAL INTERCHANGE CODE (EBCDIC) is an 8-bit code used in computers to represent numbers, letters, and special characters.
- 4-56. AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE (ASCII) is another 8-bit code developed to standardize a binary code to give the computer user the capability of using several machines to process data regardless of the manufacturer.
- 4-57. A PARITY BIT is used to detect errors in computer circuitry.
- 4-58. MAGNETIC CORE STORAGE is used as primary storage in some computers.
- 4-59. STORAGE CAPACITY AND ADDRESSES are designed and built into the computer by the manufacturer.
- 4-60. Data in SECONDARY STORAGE (like floppy disk or CD-ROM disk) is normally organized by bits, characters (bytes), fields, records, and files.
- 4-61. STORAGE ACCESS METHODS vary with the types of media and devices being used.
- 4-62. SEQUENTIAL-ACCESS STORAGE is associated with RAID and hard drives.
- 4-63. DIRECT-ACCESS STORAGE is obtained by using floppy disks and CD-ROM disks.
- 4-64. RANDOM-ACCESS STORAGE refers to magnetic core, semiconductor, thin film, and bubble storage.
- 4-65. A NETWORK is any system composed of one or more computers and terminals. However, most are composed of multiple terminals and computers.

4-66. LOCAL AREA NETWORKS (LANs) allow dissimilar machines to exchange information within one universal system within a building or small geographic area.

4-67. WIDE AREA NETWORKS provide for global connections and are sometimes referred to as global networks.

4-68. A MODEM converts the digital signal produced by a terminal or the computer to an audio signal suitable for transmission over a communications line. It also converts the audio signal back to a digital signal before it is supplied to a terminal or computer.

Check-on-Learning Questions

1. What is a general term used to describe raw facts?
2. How is data represented?
3. By what two means can the data contained on a source document be converted into a machine-readable form for processing?
4. By using an 8-bit code, how many characters or bit combinations can be represented?
5. What is the base of a hexadecimal number system?
6. What term is used for the representation of two numeric characters stored in eight bits?
7. What was the purpose of several computer manufacturers cooperating to develop ASCII code for processing and transmitting data?
8. Are there any differences in the concepts and advantages of ASCII and EBCDIC?
9. How is the parity bit in each storage location used?
10. What area in the computer holds the processing instructions (the program)?
11. What is a bit?
12. How many bits make up a byte?
13. Primary storage capacities are usually specified in what unit of measure?
14. How many core planes are needed to code in 8-bit EBCDIC?
15. Who designs and builds the storage capacity of an address into a computer?
16. What is another name for computers designed to be character-oriented or character-addressable?
17. Which computer has the faster calculating speeds—the variable-word-length or the fixed-word-length?
18. Any system composed of one or more computers and terminals can be defined as what?
19. What consists of a local area network?
20. How many designs are there of local area networks that can be used?
21. What are the different designs of local area networks called?
22. In what year was the first successful communications satellite for business applications launched?
23. What is the name of the device that is added to each end of a communications line in order to transmit data to a computer over long distances?
24. What are the names of the two methods of transmission over a modem?
25. Whenever data is transferred between devices, it involves the exchange of prearranged signals. What is this process called?

This page intentionally left blank.

Appendix A

Check-on-Learning Answers

CHAPTER 1 (Operational Concepts)

1. Five ways.
2. Analog.
3. Fire control.
4. Use electrical components.
5. Integrated circuitry.
6. Special purpose.
7. Its design.
8. Cannot be used to perform other operations.
9. Designed to perform a wide variety of functions and operations.
10. They can store and execute different programs in its internal storage.
11. Speed and efficiency.
12. Special purpose.
13. Continuous electrical or physical conditions.
14. True.
15. Hybrid computers.
16. For business and scientific data processing.
17. Digital uses discrete data while analog uses continuous data.
18. By the accuracy with which physical quantities can be sensed and displayed.
19. Third.
20. Number of significant figures.
21. Design of the computer processing unit.
22. General purpose digital computers.
23. Four.
24. Technological advancement.
25. Very large.
26. Thousands of a second (millisecond).
27. Unsophisticated and machine oriented.
28. Use of small, long-lasting transistors.

29. Use of magnetic disk storage and magnetic cores.
30. Symbolic machine languages.
31. Reduced physical size of computers.
32. Billionths of a second (nanosecond).
33. Over 100 million.
34. Scientific and business data processing.
35. Microcomputers.
36. How to properly and effectively use the available computer power.
37. Word processing.
38. Six.
39. Operating system.
40. To load an operating system.
41. No operating system loaded on disk.
42. By its outer edges.
43. Ten.
44. Destruction of some or all of the data.
45. 10 and 50 degrees Celsius (50 to 120 degrees Fahrenheit).
46. To recover a file that has become destroyed or unusable.
47. Use floppy disk and the disk copy procedure.
48. CD-ROM disks.

CHAPTER 2 (Hardware)

1. The CPU.
2. Three.
3. Control section, internal storage section, and arithmetic-logic section.
4. A telephone exchange.
5. Transfer, arithmetic, logic, control.
6. Logic.
7. Internal storage.
8. Loading.
9. Tiny doughnut-shaped rings made of ferrite (iron).
10. Hundreds of thousands of tiny electronic circuits.
11. Integrated circuit.

12. All data in memory is lost when the power supply is removed.
13. Core storage.
14. Semiconductor.
15. By passing a current through a control circuit imprinted on top of the crystal.
16. The data is still present after being read.
17. Read-only memory.
18. Computer manufacturer.
19. Yes.
20. Read/write memory.
21. Giving the computer the address of the location where the data is stored or will be stored.
22. Already programmed or blank.
23. If a mistake is made and entered, it cannot be corrected or erased.
24. It can be erased?
25. Burst of ultra-violet light.
26. Their direct-access capabilities.
27. Tracks.
28. Disk address.
29. Bits per inch of track and the tracks per inch of surface.
30. Cylinder and sector.
31. Input and output devices.
32. The keyboard.
33. Gives fast, immediate access to specific data without having to examine each record from the beginning.
34. 3 1/2-inch.
35. Printers.
36. Daisy-wheel, dot matrix, ink jet, and laser.
37. To enter or input alphanumeric character codes.
38. Display alphanumeric data and graphics.
39. Two.

40. 525.
41. Pixels or pels.
42. The number of horizontal scan lines.
43. Character addresses.
44. Depth.
45. External light source (backlight).

CHAPTER 3 (Software)

1. Operating system.
2. Single user/single tasking.
3. Multi-user/multi-tasking.
4. Must be compatible with operating system.
5. Comes in several versions.
6. Boot.
7. From a disk.
8. Utility.
9. Provide it with information about files, data fields, and the process to use.
10. Sorting.
11. What data field or fields to sort on.
12. To generate programs to print detail and summary reports of data files.
13. Machine.
14. Early 1950s.
15. To allow a programmer to work in a language that is close to English or mathematical notation.
16. FORTRAN (FORmula TRANslator).
17. Require more space and they process data at a slower rate than symbolic languages.
18. The process of planning the computer solution to a problem.
19. Four.

20. Advance preparation.
21. System (data) and programming.
22. Programming.
23. Four.
24. No.
25. What problem you are attempting to solve.
26. Instruction.
27. Arithmetic.
28. Debugged and tested before implementation.
29. Bugs.
30. To determine that all data can be processed correctly and that the output is correct.
31. With the problem definition.
32. Software written by the manufacturer, a software house, or central design agency.
33. To enter data and then retrieve it in a variety of ways.
34. A table of rows and columns.
35. No.

CHAPTER 4 (Data Representation and Communications)

1. Data.
2. By symbols.
3. Direct or indirect.
4. 256.
5. 16.
6. Packing or packed data.
7. Standardize a binary code.
8. No.
9. To detect errors in the circuitry.
10. Program storage area.
11. A single binary digit that represents the smallest unit of data.
12. Eight.
13. Number of bytes.
14. Nine.
15. The manufacturer.

16. Variable-word-length computer.
17. Fixed-word-length.
18. Network.
19. Various machines linked together within a building or adjacent buildings.
20. Two.
21. Broadband and baseband.
22. 1965.
23. A modem.
24. Asynchronous and synchronous.
25. Handshaking.

Glossary

Ada	DOD Standard Computer Software Language named after Lady Ada Augusta Byron
ADP	automated data processing
ASCII	American Standard Code for Information Interchange
AT&T	American Telephone & Telegraph
ATTN	attention
BASIC	Beginner's All-Purpose Symbolic Instruction Code
BIOS	Basic Input-Out System
CD-ROM	compact disk-read only memory
CDA	Central Design Agency
CMOS-RAM	Complementary Metal Oxide Semiconductor-Random Access Memory
COBOL	COmmon Business Oriented Language
CP/M	Control Program for Microprocessors
CPU	central processing unit
cps	characters per second
CRT	cathode-ray tube
DA	Department of the Army
D.C.	District of Columbia
DIR	direction
DOD	Department of Defense
DOS	disk operating system
DVORAK	This keyboard arranges its letters according to frequency. The home row uses all five vowels and the five most common consonants (AOEUIDHTNS).
EBCDIC	Extended Binary Coded Decimal Interchange Code
EDI	Electronic Data Interchange
ELD	electroluminescent display
ENIAC	Electronic Numerical Integrator and Computer
EPROM	erasable programmable read-only memory
FM	field manual
FORTRAN	FORmula TRANslator
GPD	gas plasma display
GUI	graphical user interface
HQ	Headquarters
I/O	input/output
IBM	International Business Machines Corporation
IC	integrated circuit
ID	identification
Inc.	Incorporated
KB	kilobyte
LAN	local area network
LCD	liquid crystal display
LED	light emitting diode
LES	Leave and Earning Statement
LINUX	Linus Torvald's UNIX (flavor of UNIX for PCs)
LOGMARS	Logistics Marking System
LSI	large-scale integration

MB	megabyte
ME	Millennium Edition (Microsoft Windows)
MS-DOS	Microsoft-Disk Operating System
9x	Windows 95/98
No.	number
OCR	optical character reader
OS	operating system
PASCAL	high level structured programming language named for 17th century mathematician Blaise Pascal
PC	personal computer
PROM	programmable read-only memory
QWERTY	This name originates from the first six letters in the top alphabet row (the one just below the numbers) of standard typewriter keyboard.
RAID	Redundant Array of Inexpensive Disks
RAM	random-access memory
ROM	read-only memory
RPG	report program generator
rpm	revolutions per minute
rps	revolutions per second
SERVMART	A self-service shopping facility that an ashore supply activity operates to provide a ready supply of relatively low-cost items needed by military customers and some items for public customers in the area.
SNAP	Shipboard Non-Tactical ADP Program
SSN	social security number
TC	training circular
TCP/IP	Transmission Control Protocol/Internet Protocol
TRADOC	Training and Doctrine Command
TV	television
UCC	Uniform Code Council
UIC	unit identification code
UNIVAC I	Universal Automatic Computer I
UNIX	trademark of AT&T
UPC	Universal Product Code
US	United States
VA	Virginia
VISIO	a drawing and diagramming solution to help people transform business and technical concepts into visual diagrams.
VLSI	very large-scale integration
XP	Experience (Microsoft Windows XP)

References

- DA Form 2028. Recommended Changes to Publications and Blank Forms.
- FM 11-72. Communications-Electronics Fundamentals: Digital Computers.
30 September 1977

This page intentionally left blank.

INDEX

A
Analog computers, 1-7

C
Central processing unit,
sections
 arithmetic-logic, 2-3
 control, 2-2
 memory (internal
 storage), 2-3
Computer
 accuracy, 1-8
 analog, 1-7
 classification, 1-4
 definition, 1-1
 digital, 1-7 and 1-14
 electromechanical, 1-5
 electronic, 1-6
 general purpose, 1-7
 history, 1-2 and 1-3
 mechanical, 1-4
 programs, 1-8, 3-1
 special purpose, 1-6
Computer coding systems,
 4-3 through 4-6
Computer programs, 1-8,
 2-2, 3-1

D
Data
 management, 3-25 and
 3-26
 representation (sym-
 bols), 4-1 and 4-2
 source, 4-2
Desktop personal
computer, 1-10, 1-15
Digital computer,
 Generations
 first, 1-10
 fourth generation and
 beyond, 1-12
 second, 1-10
 third, 1-11
Digital computers, 1-7 and
 1-14

Disk data (storage)
 cylinder method, 2-11
 sector method, 2-12
Display devices, 2-18
 through 2-23

E
Electromechanical
computers, 1-5
Electronic computers, 1-6
ENIAC, 1-3

F
Flowcharting
 constructing, 3-17
 definition, 3-13
 symbols, 3-15 and
 3-16
 tools, 3-14 and 3-15

G
General purpose com-
puters, 1-7
Generations (digital com-
puters)
 First, 1-10
 Fourth and beyond,
 1-12
 Second, 1-10
 Third, 1-11

I
Input devices
 keyboards, 2-17 and
 2-18
Internal storage (types of),
 2-4

L
Languages
 machine, 3-9
 procedure-oriented,
 3-10
 programming, 3-9
 symbolic, 3-10

M
Mechanical computers,
 1-4
Memory
 erasable program-
 mable read-only, 2-8
 programmable read-
 only, 2-8
 random-access, 2-8
 read-only, 2-7

N
Networks, 4-14 through
 4-16

O
Operating principles, 1-1
Operating system
 (booting), 1-16 and 1-17
Operating systems
 definition, 3-1
 design, 3-3
 functions, 3-4 and 3-5
 types of, 3-2 and 3-3
Output devices (printers),
 2-15, 2-16, and 2-17

P
Printers
 daily-wheel, 2-15
 dot-matrix, 2-15
 ink jet, 2-16
 laser, 2-17
Program (coding), 3-19
 through 3-21
Programming
 definition, 3-11
 overview, 3-12

S

Special purpose computers, 1-6

Storage

 bubble, 2-6

 internal (classifications of), 2-7

 internal (types of), 2-4

 magnetic core, 2-5

 magnetic disk, 2-9

 secondary, 2-8

 semiconductor, 2-5

Storage access methods

 direct-access, 4-12

 sequential-access, 4-12

 random-access, 4-14

Storage devices

 (input/output)

 external, 2-12

 floppy disk drive

 units, 2-14

 magnetic disk drive

 units, 2-13

Storage media, handling

 and backup, 1-17, 1-18, and 1-19

U

UNIVAC I, 1-3

Utility Programs

 sort-merge, 3-7

 report program

 generators (RPG),

 3-8

W

Word processing, 1-14,

 3-23 and 3-24

TC 9-72
12 SEPT 2005

By Order of the Secretary of the Army:

PETER J. SCHOOMAKER
General, United States Army
Chief of Staff

Official:



SANDRA R. RILEY
Administrative Assistant to the
Secretary of the Army
0523407

DISTRIBUTION:

Active Army, Army National Guard, and United States Army Reserve: Not to be distributed.
Electronic Media Only.

