CHAPTER 8

COMPUTERS

As late as the middle 1970's, the phrase "kick the tires and light the fires" was the main theme in launching an aircraft sortie. This meant that as long as there was an airframe with nothing falling off, an engine that would start and achieve takeoff speed, and air in the tires, the aircraft would be launched. This was done so the pilots would get their flight time every month. Now the *mission* has become the prime objective of the aircraft. This is not meant to belittle the importance of the engine and airframe. Obviously, they are important, but the aircraft and pilots are designated to perform certain missions. The performance of these missions is dependent upon the status of the various avionics packages. If one or more of these packages are degraded or not working at all, the aircraft is considered to be partial mission capable or not mission capable.

This lack of mission capability has thrust many an avionics work center supervisor into the spotlight. If you are one of the supervisors who has been there, then you know how *pleasant* the maintenance chief is to you. It is then that you realize that aircraft maintenance is not a game. As we head toward the 21st century, newer and more sophisticated aircraft are being designed and built. The avionics systems are becoming more complex, thus allowing the aircraft to perform more difficult missions. The increased complexity forces the solutions to problems in microseconds. The only system capable of performing these solutions is the computer. In turn, each associated avionics system will act as a sensor that feeds continuously updated information to the computer. The computer processes the data and sends out information to where it is needed.

Because computers are used so extensively in Navy aircraft, the avionics supervisor must have a basic understanding and working knowledge of computers.

COMPUTER MAKEUP

Learning Objective: *Identify computer hardware and software.*

The electronic components of a computer are commonly called *hardware*. Examples of computer hardware are cathode-ray tubes, transistors, microchips, printed circuit cards, etc. *Software,* on the other hand, is a term that is applied to a set of computer programs, procedures, and possibly associated documentation concerned with the operation of a data processing system. Software includes compilers, assemblers, executional routines, and input/output libraries. The advances in computer software provide the industry with the greatest realm of application possibilities. The problem of attempting to communicate with a computer has led to the development of symbolic languages that approach human language. The fact that a person can tell a computer what to do, just as one directs the actions of another person, has been made possible by the advances in software.

Software is also used to overcome design deficiencies in computers. Programming around design deficiencies is a common practice in the computer industry. Software is, in fact, often used to determine design feasibility. The practice of designing a computer with a computer is a common practice of design engineers.

Perhaps the best software application has been in the area of real-time processing. Real-time processing is a situation where the data is submitted to a computer, and an immediate response is obtained. The capability of a computer to perform real-time processing could determine the success or failure of an aircraft's mission.

Programming in a universal language has led to the development and refinement of a number of computer languages. Many of these languages are for a special area or purpose. For example, FORTRAN (FORmula Translator) for business and scientific programs, COBOL (COmmon Business Oriented Language) for business, and Jovial for large scale, computer-based, command and control systems. PL/1 (Programming Language/one) is a language for real-time systems. Each of the languages fulfills a specific need for a specific problem, but lacks the universal ideal application.

COMPUTER APPLICATIONS

Learning Objective: *Identify computer applications.*

Computer applications fall into a variety of broad categories. Information retrieval is one such application, or in a narrower sense, indexing or cataloging. Information is stored under a variety of key words or index headings. By calling up one of these headings, a listing of all or part of the information will be outputted by the computer.

Another application is simulation. This involves simulating the operation of a new computer by using an older computer model. In this way, design deficiencies can be identified without going through the time-consuming and expensive process of building the newer unit.

Real-time control of a production process is another application. For example, the petroleum and chemical industries put this process to good use. The computer can detect minute changes in the production process and initiate immediate corrective action.

The advent of personal (home) computers has greatly expanded the computer-use horizon from the routine upkeep of a checkbook balance to the more complex functions of financial planning, home security, and computer video games.

The application of the computer and its functions is virtually endless. For this reason, there are some people who believe that the computer will soon control everything and everyone. This is not necessarily the case, however, as computers can do only what their creators have intended them to do. The computer enables people to do more than they have been able to do in the past. For example, computations that required years to calculate by human methods can now be accomplished in a matter of moments by modern computers. This has become particularly evident in our space program. The ability to put a man on the moon and send *Voyager I* and *Voyager II* on their journeys would have been impossible without the use of computers. Fears over job losses are, for the most part, needless. While some jobs may be eliminated, new ones are created. Thus, a worker may have to learn a new skill. For example, a laborer may have to be retrained as a computer programmer or operator. Rather than destroying jobs, the computer creates opportunities where none existed before.

TYPES OF COMPUTERS

Learning Objective: *Identify the types of computers and the analytical processes used by each type.*

In general, there are two basic types of computers— *analog* and *digital.*

ANALOG COMPUTERS

The term *analog,* as applied to computers, pertains to representation by means of continuously variable physical quantities. For example, an analog computer can be a device that solves problems by setting up electrical circuits that represent the physical equivalents of certain phenomena. Then, measurements are made as these circuits are varied in accordance with changes in the phenomena. The analog computer is by no means restricted to electrical circuits as equivalents. The physical equivalents may be gear trains, gases, fluids, etc.

Analog computers, because of their nature, have some inherent limitations. The use of physical equivalents limits their versatility. They are limited to performing only the tasks for which they were designed or, in certain instances, closely related tasks.

DIGITAL COMPUTERS

A digital computer is a device that solves problems by manipulating the numerical equivalents of phenomena in accordance with mathematical and logical processes. These numerical equivalents may be expressed as binary numbers, octal numbers, decimal numbers, etc. In an electronic digital computer, the numerical equivalents are generally expressed as binary numbers 1 or 0. Values of voltage and current are used to represent the 1s and 0s.

The versatility of digital computers is based on the fact that they use numerical equivalents not only to represent the data to be processed, but also the instructions for processing the data. In other words, digital computers are generally provided with a wide variety of instructions. They are designed to respond in certain ways to the numerical equivalent of these instructions. Programming is merely a matter of modifying and/or arranging these instructions so that the computer will respond in a predictable manner to a given situation. While much more versatile than an analog system, digital systems are still limited as to the variety of tasks they can perform by the following factors:

- The design of their central processors
- The variety of input/output devices used
- The programmer's capability to develop a numerical method for representing and solving the problem

There are two basic types of digital computers—the special-purpose and the general-purpose computer.

Special-Purpose Digital Computers

Special-purpose digital computers are designed to follow a specific set of instruction sequences that are fixed at the time they are manufactured. To change the operation of this type of computer, the actual construction of the machine has to be altered.

General-Purpose Digital Computers

General-purpose digital computers follow instruction sequences that are read into and stored in memory prior to the calculation performance. This type of computer operation can be altered by inputting a different set of instructions. Since the operation of general-purpose digital computers can be changed with relative ease, as compared to special-purpose computers, they provide a far greater usage flexibility.

DIGITAL COMPUTER OPERATION

Learning Objective: *Recognize the operating principles of a digital computer.*

Each major section of the digital computer is comprised of various electrical circuits. These circuits include flip-flops (bistable devices), amplifiers, gates (such as AND and OR gates), and passive memory elements. These elements are organized into registers, counters, and gates. Registers are a series of electronic devices for temporary storage of a binary word. Counters are a series of electronic devices that progress through a specific binary sequence. The gates are used to set a flip-flop or generate a times condition signal. The computer manipulates binary numbers representing numerical values or conditions. Devices to retain these binary figures comprise the majority of the computer registers, and each register has a distinct purpose or function. Many operations require that the binary word or data be transferred from one register to another. It is possible for several different words to be transferred simultaneously.

Gates are used to control the transfer of data words from one register to another. These gates consist of diode and resistor networks. The gate circuit generates a signal to transfer the contents of one register to another at a particular time if certain conditions are met. For example, if the instruction being executed is an add, and if one of the numbers being added is a negative number, then the gate will generate a command signal. If these conditions are not met, the gate will not generate the command signal.

Several gates in the computer are active only during specific instructions, such as divide or multiply, and then only during that particular instruction. On the other hand, some gates are active during several instructions, generating command signals. In the design of a computer, each instruction that the computer is to perform is very methodically analyzed, and for each signal required, a gate is designated to generate the signal.

The size of the registers determines the general size of the computer. Not all registers in the computer have the same word length. Some are determined by the accuracy required, while others are determined by the instruction word, number of addresses in the memory, and various other parameters.

DIGITAL DATA PROCESSOR

Learning Objective: *Referring to various schematic and block diagrams, recognize the components of a digital data processor and the function(s) of each.*

Figure 8-1 is a functional block diagram of a digital data processing set. Of the processes that take place within a computer, the manipulation of data is

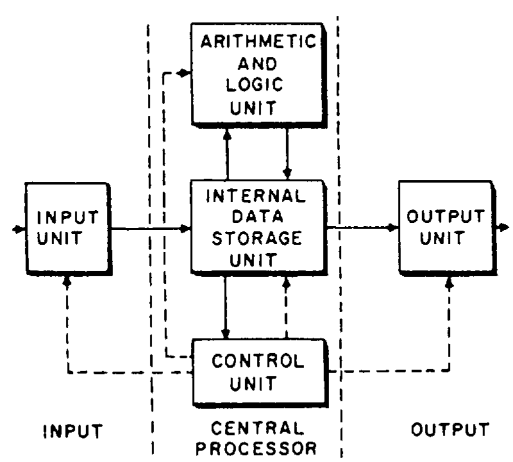

Figure 8-1.-Digital data processor block diagram.

the most important. This data manipulation takes place within the central processor (CP) (area encompassed by vertically dashed lines). The CP consists of three basic units. These units are as follows:

- Control unit—This unit directs the overall operation of the computer in accordance with a prescribed plan.

- Arithmetic-logic unit-This unit performs the actual processing.

- Internal data storage unit-This unit stores the data to be processed and the prescribed plan (program).

CONTROL UNIT

In a typical digital computer, the control section includes the instruction register, the P register, the general register(s), and the SC register. A brief explanation of each type of register follows:

- Instruction register—This register holds the instruction code during execution. The size of the register is dependent upon the instruction word and makeup of the computer. The instruction code usually has more than one part or field.

- P register—This register contains the address of the next sequential instruction to be executed. The contents of the P register are automatically advanced by one by the P + 1 adder.

- General register—This register stores the quantity used for address modification. In addition, it usually has the properties of automatic increment or decrement. Most computers have more than one general register.

- SC register—This register consists of one or two registers to accomplish the holding of a shift count. Its size is dependent on the maximum number of places that a word can be shifted.

An easy way to comprehend the operation of the control unit is to draw a comparison to a telephone exchange. The act of dialing a phone number energizes certain switches and control lines in a telephone exchange. In a similar manner, each program instruction, when executed, causes the control section to energize certain “switches” and “control lines.” This enables the computer to perform the function or operation indicated by the instruction.

A computer program can be stored in the internal circuits of the computer, or it may be read instruction-by-instruction from external media. The internally stored program type of computer is the most practical type to use when speed and fully automatic operation are desired. This type of computer is known as a stored-program computer.

In addition to the command that tells the computer what to do, the control unit also dictates how and when each specific operation is to be performed. It is also active in initiating circuits that locate information stored in the computer and in moving this information to the point where the actual manipulation or modification is to be accomplished.

In the stored-program computer, the control unit reads an instruction from the memory section. The information read into the control unit from memory is in the form of varying voltage levels that make up the binary word. This word represents a specific operation that is to be performed. The location of the data to be operated on is generally a part of the instruction and energizes circuitry that causes the specified operation (add, subtract, compare, etc.) to be executed. Subsequently, the control unit reads the next instruction or jumps as directed to find the next instruction to execute.

Computer instructions are broken down into four general categories. The instructions are transfer, arithmetic, logic, and control.

Transfer commands transfer data from one location to another. One of the instructions is usually an address in memory, and the other is either a register or an input/output device.

Arithmetic instructions combine two pieces of data to forma single piece of data by using one of the arithmetic operations. In some computers, one of the pieces of data is in a location specified by an address contained in an instruction. The other piece of data is already in a register (usually the accumulator). The results are usually left in the accumulator.

Logic instructions make the digital computer much more than an expensive adding machine. The use of logic instructions enables the programmer to construct a program capable of a number of tasks. These instructions enable a computer used for inventory maintenance to follow one set of procedures if an inventory item count is too high, and another if the count is too low. The choice of which set of procedures to use is made by the control unit under the influence of the logic instructions. Logic instructions provide the computer with the ability to make decisions based on the results of previously generated data.

Control instructions send commands to devices not under direct control of the control unit, such as input and output units. The address portion of the control instruction does not specify a location in memory, but is usually a coded group that specifies an action be required of a particular piece of equipment.

In a single-address computer, where each instruction refers to only one address or operand, the instructions are normally taken from the memory in sequential order. If one instruction comes from a certain location, such as X, the next instruction is usually taken from location X + 1. However, the execution of a logic instruction may produce a result that dictates that the next instruction is to be taken from an address as specified in a portion of the logic instruction. For example, the logic instruction may initiate certain operations in the computer to determine if the content of a given register in the arithmetic section is negative. If the answer is yes, the location of the next instruction is specified in an address section of the logic instruction. If the answer is no, the next instruction would be taken from the next sequential location in the memory.

Every computer provides circuitry for a variety of logic instructions, thus providing the capability of selecting alternate instruction sequences if certain desirable or undesirable conditions exist. The ability to branch at key points is the special feature of the computer that makes it able to perform such diverse tasks as missile control, accounting, and tactical air plotting.

ARITHMETIC-LOGIC UNIT

The arithmetic-logic unit (ALU) is the section in which arithmetic and logic operations are performed on the input or stored data. The operations performed in this unit include adding, subtracting, multiplying, dividing, counting, shifting, complementing, and comparing.

Generally, information delivered to the control unit represents instructions, while information routed to the arithmetic unit represents data. Frequently, it is necessary to modify an instruction. This instruction may have been used in one form in one step of the program, but must be altered for a subsequent step. In such cases, the instruction is delivered to the arithmetic unit where it is altered by addition to or subtraction from another number in the accumulator. The resultant modified instruction is again stored in the memory unit for use later in the program.

All arithmetic operations can be reduced to one of four processes: addition, subtraction, multiplication, or division. In most computers, multiplication involves a series of additions, and division is a series of subtractions.

The arithmetic unit contains several registers. Each register is a unit that can store one “word” of computer data. These registers generally include the D, X, and Q register, and a unit called the accumulator (A register). The registers are so named for identification purposes only. During an arithmetic process, the D, X, and Q registers temporarily hold or store the numbers being used in the operation. These numbers are called the operands. The accumulator stores the result of the operation. The control unit instructs the arithmetic unit to perform the specified arithmetic operation as requested by the program. The control unit transfers the necessary information into the D, X, and Q registers from memory and controls the storage of the results in the accumulator or in some specific location in memory.

The arithmetic unit also makes comparisons and produces yes or no or GO/NO-GO outputs as a result. The computer can be programmed so that ayes or GO result causes the computer to perform the next step in the program, while a no or NO-GO result causes the computer to jump several steps in the program. A computer can also be programmed so that a no result at a certain point in the program will cause the computer to stop and await instructions from a keyboard or other input device.

INTERNAL DATA STORAGE UNIT

In some digital computers, the internal data storage unit, or memory section, is constructed of small, magnetic cores, each capable of representing anon or off condition. An on condition is represented by a 1 and an off condition is represented by a 0. A system of these cores arranged in a matrix can store any computer word that is represented in binary form.

All computers must contain facilities to store computer words or instructions until these instructions or words are needed in the performance of the computer calculations. Before the stored-program type of computer can begin to operate on its input data, it is first necessary to store, in memory, a sequence of instructions and all numbers and other data that are to be used in the calculations. The process by which these instructions and data are read into the computer is called *loading.*

The first step in loading instructions and data into a computer is to manually place enough instructions into memory by using the console or keyboard. These instructions are then used to bring in more instructions as desired. In this manner, a few instructions are used to “bootstrap” more instructions. Some computers use an auxiliary (wired) memory, which permanently stores the bootstrap program, thereby making the manual loading unnecessary. These instructions may be stored in chips. These chips are referred to as “read only memories” or ROMs.

The memory section of a computer is essentially an electronically operated file cabinet. It is actually a large number of storage locations. There are generally between 1 and 40,000 locations. Each one is referred to as a storage address or register. Every computer word that is read into the computer during the loading process is stored or filed in a specific storage address and is almost instantly accessible.

The types of memory storage devices used most frequently in present-day computer technology are magnetic cores, semiconductor, thin film, magnetic drum, magnetic tape, and magnetic disks.

Magnetic Cores

One of the methods for storing internal data in a computer is realized by using magnetic cores. Cores are generally constructed by two methods. The first type of core, called a tape wound core, is fabricated by wrapping a tape of magnetic material around a nonmagnetic toroidal form. A *toroid* is a term used to describe a doughnut-shaped solid object. The second type of core is called a ferrite core, and it is made by molding finely ground ferrite into a toroidal form. The ferrite used in this application is a ceramic iron oxide possessing magnetic properties. The ferrite particles are then heat-fused or “sintered” by the application of heat and pressure.

In magnetic core memories, each data bit is stored in the magnetic field of a small, ring-shaped magnetic core (fig. 8-2). Magnetic cores generally have four wires running through them. Two wires are used for READ selection. These same two wires are used for WRITE by reversing the direction of current flow. An inhibit wire prevents writing a 1 when a 0 is to written. The sense wire picks up the signal voltage generated by the shifting of core from 1 to 0 in a READ cycle.

Since a single core stores only one bit of a word, a large number of cores are required to handle all the bits in every word to be stored. These cores are arranged in arrays to assign memory address locations and quickly write data and locate data for read-out purposes. The technique used most frequently for writing and reading data in magnetic core arrays is known as the coincident-current technique.

In computer memory applications, the ferrite core is magnetized by a flux field produced when a current flows in a wire (drive line) that is threaded through the core. The core retains a large amount of this flux when the current is removed. Flux lines can be established clockwise or counterclockwise around the core, depending on the direction of the magnetizing current. A current in one direction establishes a magnetization in a given direction. Reversing the direction of the current flow reverses the direction of the flux field and the core magnetization. These two unique states represents 0 and 1, respectively.

Semiconductor Memories

Semiconductor memories are used in many modern computers. Most of the semiconductor memories are of the MOS LSI type, which may be

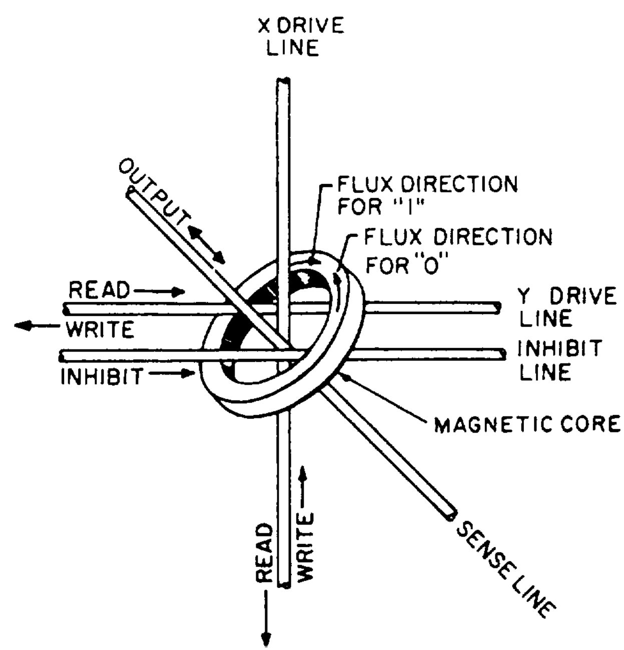

Figure 8-2.-Magnetic core showing X, Y, inhibit, and sense lines.

static or dynamic. MOS means metal oxide semiconductor, and LSI means large scale integration.

Thin Film

Thin film memory consists of Permalloy, a ferromagnetic material, deposited on a supporting material (substrate) of thin glass. This is done under controlled conditions in a vacuum chamber. When all air has been removed from the chamber, a shutter arrangement is opened, and vapors from molten Permalloy pass through a mask and are deposited on the supporting material. The pattern thus formed is determined by the shape of the mask. The thickness of each spot (magnetized area) is controlled by the amount of time the shutter is open.

A magnetic field is applied parallel to the surface of the substrate during deposition. The film spots become easier to magnetize in the direction parallel to that in which the magnetic field was applied during the deposition process. This direction is known as the preferred direction; likewise, the axis of this magnetism is called the preferred axis.

Magnetic Drums

The magnetic drum storage device is a cylinder that rotates at a constant velocity. Information is written on or read from the drum when its magnetic surface passes under magnetic heads, which are similar to the magnetic heads found on commercial tape recorders.

Magnetic drums provide a relatively inexpensive method of storing large amounts of data. A magnetic drum (fig. 8-3) is made from either a hollow cylinder (thus the name drum) or a solid cylinder. The cylinder may consist entirely of a magnetic alloy, or it may have such an alloy plated upon its surface. Many drums are made by spraying on magnetite, an iron oxide. The surface is then coated with a thin coat of lacquer, and buffed.

Representative drums have diameters ranging from 12.7 to 50.8 centimeters (about 5 to 20 inches respectively). The surface of the drum is divided into tracks or channels that encircle the drum. A number of READ and WRITE heads are used for recording and reading. There is at least one head per track. The drum is rotated so that the heads are near, but not touching, the drum surface at all times.

As the drum rotates, the tracks are continuously passing under their respective head. Each track is subdivided into cells, each of which can store one binary bit. All the cells that are positioned under the heads of a multitrack drum at the same time are called a "slot." With some drums, each head reads or writes one bit of a word. Thus, when a word is written into or read from a slot, each track contains one bit of that word. The number of heads used depends on the size of the word that the computer is designed to handle.

One of the tracks provides timing signals for the drum rotation. The timing track determines the location of each set of storage cells around the drum. Each timing signal denotes a unit of time of drum rotation. For example, if the timing track is 80 inches long and timing signals are recorded at 120 pulses per inch, there are 9,600 locations for bit storage on the track. If the drum has 32 tracks in addition to the timing track, the drum has the capacity to store a total of 307,200 bits.

Some drums use two or even three timing tracks. The timing tracks are used for synchronization purposes and are sometimes called "control" or "clock" tracks. The timing pulses establish the time scale to which all circuits through the computer are synchronized.

The retrieval of data from a rotating drum can be a rather involved process, as can be realized by drawing a comparison to the core memory of a computer. When core memory is used, all the data is stored in the cores in a static condition. The data can be located at a given place at any instant, and can be easily read from that location in serial or parallel form

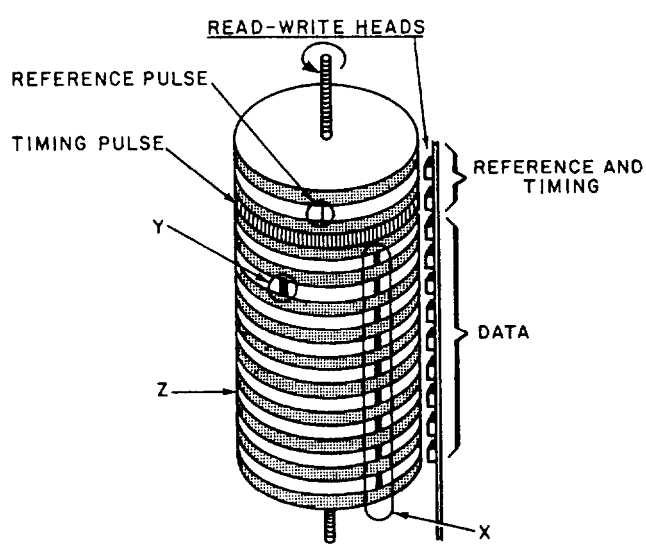

Figure 8-3.-Magnetic drum.

to represent the same data that was stored in that location.

Transfer of the data from constantly rotating magnetic drums, on the other hand, is complicated. Timing pulses are not used to synchronize the drum speed, which may vary slightly from time to time. Thus, some method must be used to ensure that data read into the drum memory in a given bit position will be read from the memory with the same time reference. The probability y of an incompatible time relationship between the drum speed and synchronizing pulses makes it necessary to establish some means of compensating for variations in drum speeds.

In practice, the drum contains a control point and a number of sectors in a specific format. The control point is a magnetic mark that specifies a starting location on the drum. All data stored on the drum is referenced to this indexing point or reference pulse, as shown in figure 8-3.

Magnetic Tapes

Magnetic tape is widely used as a storage medium for large amounts of data, or it may be used as a main storage backup. However, it is normally not used as an internal (main) storage medium because of its long access time. This is readily realized if you consider that needed information is widely, and sometimes randomly, distributed along the tape. Magnetic tape has two main advantages—its large storage capacity and its low cost.

Magnetic Disks

The magnetic disk is a convenient medium for semipermanent storage of mass volumes of production programs. For many applications, disks

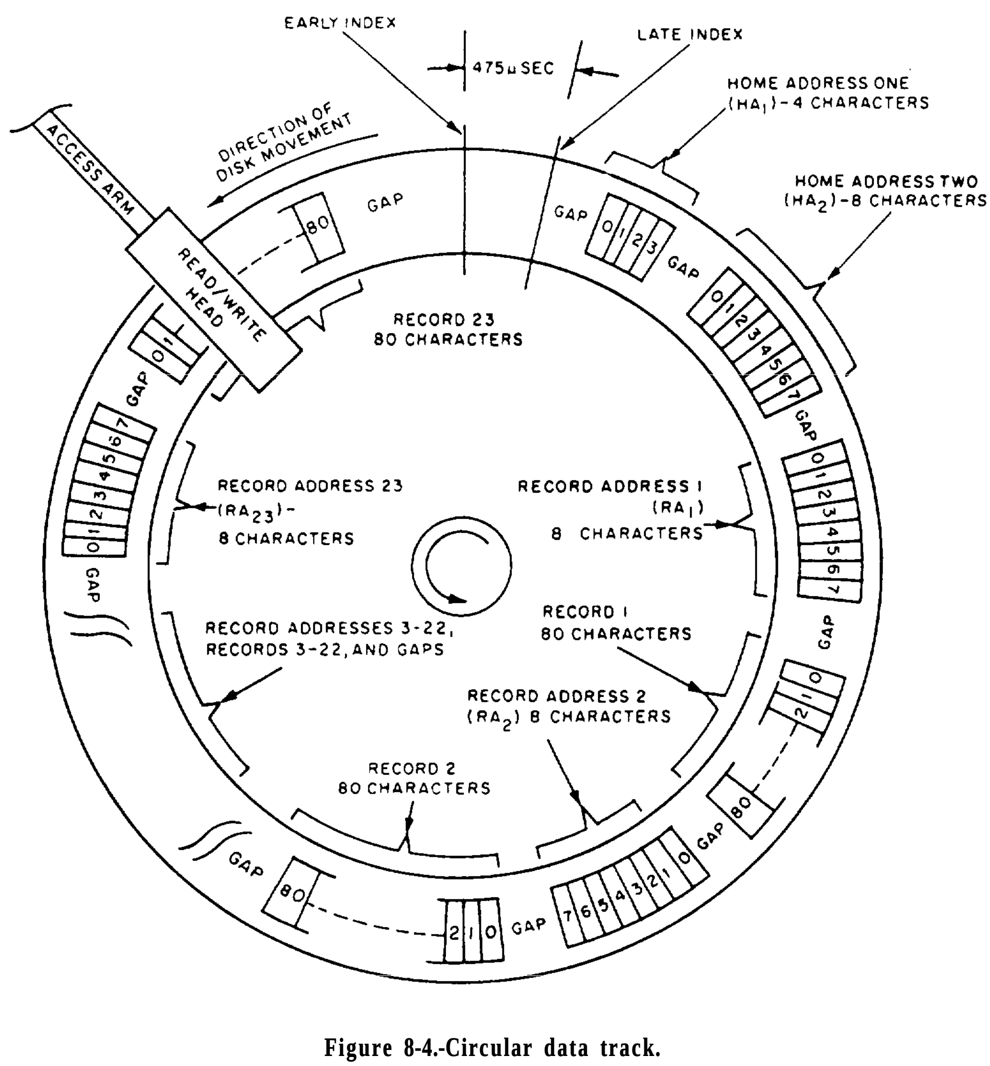

Figure 8-4.-Circular data track.

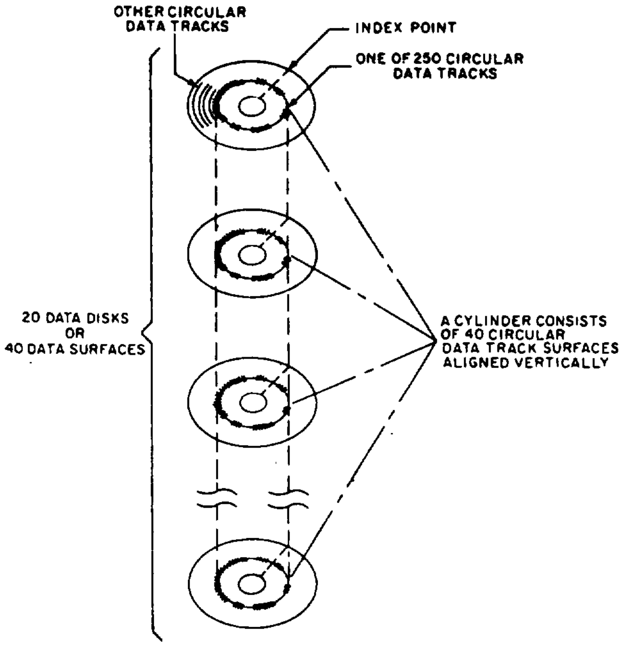

Figure 8-5.-Data storage disk assembly.

are superior to magnetic tapes for rapid acquisition and storage of mass volumes of system programs and data.

Magnetic disks resemble phonograph records that have been coated with iron oxide. The disks are arranged in stacks in much the same way as a record stack in a jukebox. All the disks are continuously revolving and spaced apart so that a record head driven by an access mechanism can be positioned between the disks.

Data is recorded at a certain address on a specified disk. When readout of a particular bit of data is desired, the recording head is automatically positioned, and the data is read serially from the surface of the selected disk.

The basic unit of information on the disk is called a character. By design, each character contains a given number of bits for fixed-word applications. One or more of these characters in a group form a record. A circular data track (fig. 8-4) consists of one or more records, associated record addressees, gaps, and data track identification. A number of data tracks aligned on vertically arranged disks (fig. 8-5) form a cylinder of information. A magnetic disk file system may contain one or more bands (modules). Each module contains a specified number of disks with their associated cylinders and data tracks. The flow chart in figure 8-6 illustrates the procedures necessary to retrieve or store information.

INPUT/OUTPUT (I/O) SECTION

Learning Objective: *Describe how a digital computer communicates with external peripheral devices.*

The I/O section is that portion of the digital computer through which the CPU communicates with the external peripheral devices. In a useful computer function, data is read into the computer, processed, and then transferred to the output. The peripheral units handle the data input and output display functions. The I/O section controls the transfer of data between the computer and the peripherals.

The I/O section is the interface between the computer and any extenal devices. An interface is an assembly of electronic circuits that make the computer compatible with the peripheral units. This

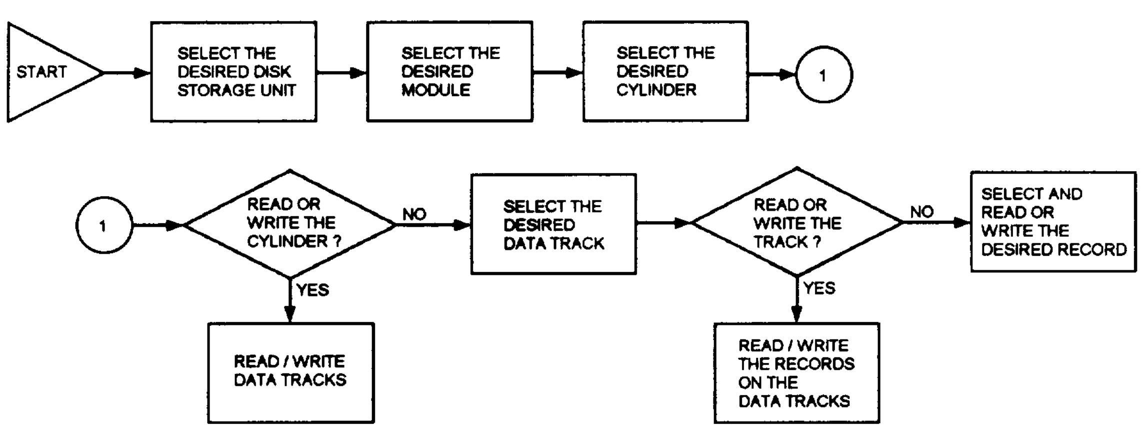

Figure 8-6.-Flow chart for storage and retrieval of data from disks.

compatibility permits the computer and the peripheral units to communicate intelligently. The compatibility involves logic levels, timing or speed, and control.

When digital data is transmitted between two units, the binary voltage or current levels must be compatible. Logic-level conversion is often required to properly interface different types of logic circuits. For example, logic-level shifting is often required to properly interface bipolar and MOS circuits. The speed of the data transmission must also be compatible. Some type of temporary storage between the two units may be required as a buffer to match the high-speed CPU to a low-speed peripheral unit.

Control is another function of the interface. There are status lines that tell when the computer or peripheral unit is ready or busy, and strobe lines that actually initiate the data transfers. This process is often referred to as "handshaking."

The type of information exchanged between the I/O unit and the peripheral devices includes data, addressing, and control signals. Since multiple I/O units can usually be connected to a computer, some coding scheme is required to select the desired unit. This is usually done with a binary word used as an address. The address is transmitted to all the peripheral devices. The unit recognizing the address is connected to the I/O section. Data can then be transmitted to or from the device over the interconnecting data lines. The actual data transfers are controlled by control signals between the two devices.

Programmed data transfers that take place as the result of executing an I/O instruction usually cause the data to be transferred between the peripheral unit and the accumulator register in the CPU. Other CPU registers may also be used, depending upon the computer architecture and the instruction. In some computers, peripheral units are addressed as storage locations, and all memory reference instructions can be used in performing I/O operations. No special I/O instructions are used in these computers.

PARALLEL VERSUS SERIAL DATA TRANSMISSIONS

There are two methods of transmitting digital data. These methods are parallel and serial transmissions. In parallel data transmission, all bits of the binary data are transmitted simultaneously. For example, to transmit an 8-bit binary number in parallel from one unit to another, eight transmission lines are required. Each bit requires its own separate data path. All bits of a word are transmitted at the same time. This method of transmission can move a significant amount of data in a given period of time. Its disadvantage is the large number of interconnecting cables between the two units. For large binary words, cabling becomes complex and expensive. This is particularly true if the distance between the two units is great. Long multiwire cables are not only expensive, but also require special interfacing to minimize noise and distortion problems.

Serial data transmission is the process of transmitting binary words a bit at a time. Since the bits time-share the transmission medium, only one interconnecting lead is required.

While serial data transmission is much simpler and less expensive because of the use of a single interconnecting line, it is a very slow method of data transmission. Serial data transmission is useful in systems where high speed is not a requirement. Serial data transmission techniques are widely used in transmitting data between a computer and its peripheral units. While the computer operates at very high speeds, most peripheral units are slow because of their electromechanical nature. Slower serial data transmission is more compatible with such devices. Since the speed of serial transmission is more than adequate in such units, the advantages of low cost and simplicity of the signal interconnecting line can be obtained.

PARALLEL DATA TRANSMISSION

In a parallel data transmission system, each bit of the binary word to be transmitted must have its own data path. There are a variety of ways to implement this data path. The two basic classifications of transmission line circuits are single-ended and balanced. Single-ended transmission systems use a single-wire data path for each bit. When combined with a ground or return reference, the electrical circuit between the sending circuit and the receiving circuit is complete. In a balanced transmission line system, two conductor cables are used to send the data. The data on the dual-transmission line is complementary. The dual-transmission lines also use a ground return reference. While a single-ended transmission line is simpler and less expensive, it is subject to more noise problems than the balanced or dual-transmission line system.

SERIAL DATA TRANSMISSION

The simplest, most economical, and easiest to use method of transferring digital information from one point to another is serial data transmission. In a serial system, the digital data is sent one bit at a time. This means only one pair of transmission wires is required. The serial transmission of data is far slower than parallel transmission. In most computer systems, the low speed is not a disadvantage. Data rates achievable in serial data systems are sufficiently high to make them very practical.

Serial data transmission is preferred because it is inexpensive. It is especially beneficial in transferring data over long distances. For long distances, you can see that multiple parallel lines are far more expensive than a single cable.

Low-speed serial data transmission also offers another benefit. The data rate is slow enough to permit the transmission of data over telephone lines. In this case the digital data is converted into a pair of audio tones representing binary 1s and 0s. These can be transmitted economically for long distances over standard telephone lines. In addition, low cost tape recorders can be used to record the serial data. This provides a low cost means of mass data storage and retrieval. Standard audio cassette recorders are widely used in this application.

Serial data transmission also permits transmission of data by radio. A radio communications path represents only a single interconnecting link similar to a transmission line pair. Therefore, for data to be transmitted by radio, it must be in serial format. Serial digital data is used to modulate a radio carrier in various ways.

In digital computer systems, you will find that both serial and parallel data transmission methods are used. Where high speeds and short distances are required, the parallel method is used. The serial method is used where low cost, simplicity, low speed, and long distances are necessary.

INPUT/OUTPUT (I/O) DEVICES

I/O devices are similar in operation but perform opposite functions. It is through the use of these devices that the computer can communicate with devices external to the computer itself (peripheral devices).

The I/O section of a computer provides the necessary lines of communication and generates such signals necessary for the computer to communicate with, and when necessary, control the operation of the I/O devices. The I/O section, once it has been initiated by the control section, usually operates independently of the control section, except when it must time-share memory with the control section.

Input Devices

Input data may be in any one of three forms:

- Manual inputs from a man-machine interface (MMI), such as a keyboard or console
- Analog and/or digital inputs from instruments or sensors
- Inputs from a source on or in which data has been previously stored in a form intelligible to the computer

Computers can process hundreds of thousands of computer words per second. Thus, 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. This actually translates to a data input rate of less than one computer word per second due to coding time. Since the computer can read several thousand times this amount of information per second, it is clear that manual inputs should be minimized to make more efficient use of computer time.

Instruments used as input sensors are capable of supplying several thousand samples regarding pressure, temperature, speed, and other measurements per second. This is equivalent to 10,000 to 20,000 bits or binary digits per second. Digital computers that use these devices must be equipped with analog-to-digital (A/D) converters (assuming the input is in an analog format) to convert physical change to specific increments.

Input data that has been previously recorded on punched cards, perforated tapes, magnetic tapes, or magnetic drums or disks in a form understood by the program may also be entered into the computer. This method is much faster than entering data manually from a keyboard. The most commonly used input devices in this category are magnetic tape readers and paper tape (perforated tape) readers.

One of the main features of computers is their ability to process large amounts of data quickly. In most cases, the processing speed far exceeds the ability of input devices to supply information. One common limitation of most input devices is that each involves some mechanical operation. For example, the movement of a tape drive or card feeder. Because a mechanical movement of some part of these devices cannot take place fast enough to match electronic speeds with the computer, these input devices limit the speed of operation of the associated computer. This is particularly evident in cases where successive operations are dependent upon the receipt of new data from the input medium.

Several methods of speeding up mechanical operation have been devised, all of which are designed to move a smaller mass a shorter distance and with greater driving force. Many of these designs have been directed toward increasing the drive speed of magnetic tapes. For example, present-day tape drives can pass up to 400 inches of tape per second of a tape reading head. Card readers can read up to 2,000 cards per minute. Present-day disk systems operate at speeds up to 3,600 RPM.

The comparative rates of data for these systems are as follows:

- Card systems—2,700 characters per second
- Tape systems—350,000 characters per second
- Disk systems—15,000,000 characters per second

Another method of entering data into a computer, which we have not previously mentioned, is to link two or more computers together and program them to communicate with each other. This is perhaps the fastest method of entering or extracting data from a computer. An example of this method is the data link.

Output Devices

Output information is also made available in three forms:

- Displayed information, such as codes or symbols presented on a monitor screen, that is used by the operator to answer questions or make decisions.
- Control signals, which is information that operates a control device, such as a lever, aileron, or actuator.
- Recordings, which is information stored in a machine language or human language on tapes or printed media.

Devices that store or read output information include magnetic tapes, punched cards, punched paper tapes, monitors, electric typewriters, and high-speed printers.

INTEGRATED CIRCUIT (IC) TECHNOLOGY

Learning Objective: *Identify integrated circuit classifications and various types of integrated circuits.*

Modern digital computers are made with ICs. ICs provide the most economical and practical method of implementing the circuits required to carry out the functions of a digital computer.

IC CLASSIFICATIONS

ICs fall into four basic levels of complexity. These are small scale integration (SSI), medium scale integration (MSI), large scale integration (LSI), and very large scale integration (VLSI). The designations define the size and complexity of individual ICs.

Small Scale Integration (SSI)

SSI circuits contain very little circuitry, generally fewer than 10 circuits no more complex than a typical logic gate. Typical SSI circuits include multiple logic gates, flip-flops, and simple combinational logic circuits.

Medium Scale Integration (MSI)

MSI circuits are more complex and sophisticated than SSI circuits. Most MSI circuits contain 12 or more circuits equivalent in complexity to a typical logic gate. MSI circuits are functional in nature in that the y perform a specific logic operation with no further interconnection. Typical MSI circuits include counters, shift registers, arithmetic-logic circuits, decoders, multiplexer, and other combinational and sequential logic circuits. MSI circuits are highly beneficial because they significantly cut design time. They also reduce the number of ICs in design, minimize circuit wiring, and reduce size and power consumption. Most digital equipment can be implemented by simply combining the proper MSI circuits.

Large Scale Integration (LSI)

LSI circuits are large and complex, and generally contain circuitry equivalent to 100 or more typical logic gates. LSI circuits are also functional in nature. Typical LSI ICs include memories and microprocessors.

Very Large Scale Integration (VLSI)

VLSI circuits are large, complex circuits that contain the equivalent of 1,000 or more logic gates.

Early digital computers use a mix of SSI MSI, and LSI circuits. Over the years, the trend has been toward the greater use of MSI, LSI, and VLSI circuits. Today, most computers are predominately MSI, LSI, and VLSI circuits. The SSI circuits are used only in interfacing the other three larger and more sophisticated types.

DIGITAL CIRCUIT CHARACTERISTICS

Many factors influence the choice of logic circuits used to implement a digital computer. The most important of these are speed, power dissipation, and the availability of MSI, LSI, and VLSI circuits. Other factors include the cost, noise immunity, and interface capabilities.

The most important choice of a type of logic involves speed and power dissipation. Speed refers to the frequency of operation or propagation delay of the logic circuits. The higher the speed of operation, the greater the amount of data that can be processed in a given time. Therefore, computer designers try to achieve as much speed as possible in their designs.

Many high-speed circuits and techniques are available. Besides choosing circuits with low propagation delay times, circuit reduction techniques are used to minimize propagation delay through proper physical layout as well as logic simplification. However, speed is always obtained at the expense of increased power dissipation and cost. Invariably, the designer faces a speed-power tradeoff, and attempts to optimize the design for maximum speed within realistic power consumption and cost guidelines.

Choosing ICs for use in a digital computer involves not only choosing a specific logic family, but also in choosing the correct mix of the various IC technologies available. By being flexible in the circuit choice of mix, a designer can more easily optimize the design. This means freely mixing different families of bipolar and MOS, as well as VLSI, LSI, MSI, and SSI circuits.

For a detailed look at bipolar and MOS theory, as well as more information on VLSI, LSI, MSI, and SSI circuits, refer to *Aviation Electronics Technician 2 (Intermediate),* NAVEDTRA 12334, chapter 2.

PROGRAMMING FUNDAMENTALS

Learning Objective: *Recognize concepts and procedures used in the construction of a computer program.*

Computer programming is the process of planning a solution to a problem. You can derive a general outline for calculating total resistance of a parallel resistance circuit by using the following steps:

1. Take the reciprocal of the resistance in ohms of all resistors in a circuit.
2. Calculate the sum of the values from step 1.
3. Compute the reciprocal of the value from step 2.

The process of preparing a program from this explanation is not difficult. One basic charcteristic of the computer must be considered-it cannot think. It can only follow certain commands, which must be correctly expressed and must cover all possibilities. If a program is to be useful, it must be broken down into specifically defined operations or steps. These operations or steps, along with other data, must be communicated to the computer in a language that it can understand.

NOTE: The instructions are read sequentially unless otherwise stated.

Generally, the steps that a computer follows in the execution of a program are as follows:

1. Locates parameters (constants) and such data as necessary for problem solving
2. Transfers the parameter and data to the point of manipulation
3. Performs the manipulation according to certain rules of logic

4. Stores the results of manipulation in a specific location

5. Provides the user with an useful output

Even with a simple program, such as the resistance program, each step must be broken down into a series of machine operations. These instructions, along with the parameters and data, must be translated into a language or code that the computer can understand.

Programming is a complex problem that may involve writing a large number of instructions. It may also involve keeping track of a great many memory cells that are used for instruction and data storage, which is time-consuming and can lead to errors.

To reduce time and the possibility of errors for complex program preparation, the compiler has been developed. The compiler is a program that takes certain commands and then writes, in a form the machine understands, the instructions necessary for a computer to execute these commands. Compilers can bring many instructions into the final program when called upon or signaled by a single source statement. The compiler is problem oriented because the operations produced are those needed to work the problem as set out by the problem statement. Compilers are built at various levels or degrees of complexity. The simplest form of compiler takes one mnemonic phrase and writes one machine instruction. A mnemonic code is an abbreviated term describing something to assist the human memory. For example, to shift the contents of the A-register right nine places, the mnemonic code RSH.A9 is used. This causes the compiler to write an instruction that shifts the contents of the A-register right 11_8 places. A compiler written on this level is commonly called an assembler. Note the advantages as follows:

1. No opportunity to use the wrong function code

2. No necessity to convert the shift count to octal

A more sophisticated compiler may take a statement, such as “multiply principal by rate,” and generate all the instructions necessary for the computer to do the following:

1. Locate the factors involved (in this case the principal and rate)

2. Transfer these factors to the arithmetic unit

3. Perform the indicated arithmetic operation (in this case multiplication)

4. Store the resultant (which, in this case, will be the interest or the principal)

The compiler also keeps track of all memory allocations, whether being used for data or instructions.

Depending on the complexity of the problem to be solved, programs may vary in length from a few instructions to many thousands of instructions. Ultimately, the program could occupy a significant, perhaps even an excessive, portion of computer memory. One method used to preclude this possibility is to segment the program, storing seldom used portions in auxiliary storage, and reading these portions into the main memory only when they are required. An important method of developing this ability is through the use of subroutines.

SUBROUTINES

As a program grows larger, certain functions must be repeated. The instructions necessary to perform each of these repeated functions are grouped together to form subroutines. These subroutines may then be referenced by a relatively few instructions in the main program. This eliminates repeating certain groups of instructions throughout the program.

EXECUTIVE ROUTINES

The instructions that control access to the various subroutines are called the executive routine of the main program. Depending upon the complexity of the program, there may also be subexecutive routines within the executive routines.

Housekeeping is a term used frequently with subroutines. At the time of entry into a subroutine, the contents of the various addressable registers may or may not be of value. An addressable register is defined as any register whose contents can be altered under program control. The programmer must take steps to preserve the contents of these registers unless they are of no value.

JUMP AND RETURN JUMP INSTRUCTIONS

The jump and return jump instructions are used to assist in the construction of executive routines. These instructions provide the computer with the ability to leave the sequential execution of the main program, execute any of the subroutines, and then return to the main program.

Execution of a return jump instruction causes the address of the next instruction to be executed in the main program to be stored (usually in the entry cell of the subroutine). It then causes the instruction of the second cell of the subroutine to be executed. The last instruction to be executed will usually be a straight jump to the address contained in the entry cell. Since a jump instruction specifies the address of the next instruction to be executed, the computer is provided with a means of returning to the main program once the subroutine has been executed.

PROGRAM CONSTRUCTION

The process of writing a program is broken down into six basic steps.

1. Statement. A statement forms a clear comprehensive statement of the problem.

2. Analysis. Analysis consists of laying out the problem in a form that will lend itself to arithmetical and/or logical analysis, determining what logical decision must be made, and if data manipulation is required.

3. Flow diagram. A flow diagram, or chart, is an expansion of the steps in which special symbols are used to represent the various operations to be performed and the sequence in which they are to fall.

4. Encoding. The process of converting the operations listed in the flow chart into language the computer will use, either machine instructions, words, or compiler statements.

5. Debugging. The process of locating errors in the program is called "debugging." Various techniques are available for this purpose. A program may be written to include some aids for itself, or a separate debugging program may be run to test the operation of a malfunctioning program. For a simple program, a trial solution may be done on paper, and the computed results compared with those actually obtained at each step.

6. Documentation. Documentation is very important because later changes may be warranted in a program, or it may be desirable to use subroutines from another program. Proper documentation will ensure that this can be accomplished. Documentation should include the following:

- Program title
- Problem statement
- Programmer's name
- Date
- Memory area used and/or number of cells used
- Registers used
- I/O devices required
- Flow diagram(s)
- Hard copy (program listings, especially a listing of the coded instructions)
- Program tapes

FLOW CHARTING

The programmer constructs a program "map" in determining a solution to a problem. This map is commonly called a flow chart and serves a multitude of important functions. The flow chart maps the logical steps required, decisions to be reached, and paths to be followed as a result of the decisions. When properly annotated, it defines input/output requirements, address allocations, data accuracy considerations, and register usage. A flow chart is valuable when debugging a program and when making future changes.

Flow charting can be constructed at various levels of complexity. A high-level flow chart consists of a few symbols and presents a broad overview of the problem. A low-level flow chart may approach a one-to-one correspondence between flow chart symbol and program instruction. Usually, there will be several flow charts for a program area. These may be compared to the prints found in a maintenance manual. These prints include a block diagram to show the relationship of major units (high-level), functional block diagrams showing the major circuits in a unit (intermediate-level), and the schematics of the circuits (low-level). Flow charts should beat such a level that they will implement all the uses previously discussed.

MAINTENANCE PROGRAMS

As we have previously stated, a routine or program is a series of instructions that control the operations of a computer. Each instruction is used to cause some action that is part of the overall task the computer must perform. Therefore, an instruction may be considered as the basic building block of a computer program.

An overall check of a computer can be done by the use of a maintenance program. The maintenance program provides a thorough and rapid method for the detection of failures in a specific portion of the computer. This type of overall maintenance check is flexible and efficient. The programs may use the same type of tape, memory, computing, and external storage media as operational programs. The maintenance program can be altered when the computer or auxiliary components are changed. The program can also be constantly improved. Generally, no extra test equipment is required since the computer circuits are used to perform the test. Testing by means of maintenance programs results in the computer circuits being used in a more comprehensive manner than during normal program execution. When a program has been checked and accepted as a good maintenance tool, it is not subject to deterioration. In contrast, test equipment may be checked and accepted only to become unreliable shortly after being placed in use.

Maintenance programs are divided into two main classes: reliability and diagnostic. Maintenance programs that are used to detect the existence of errors are called reliability programs. Reliability programs should be arranged to check as many computer circuits as possible.

Maintenance programs that are used to locate the circuits in which computer malfunctions originate are called diagnostic programs. An effective diagnostic program should locate the source of trouble as closely as possible. Actually, in many cases, reliability programs have some diagnostic features, just as diagnostic programs have some reliability features. For convenience, a program is called either a reliability or diagnostic program, depending on its intended emphasis. In general, programs that check rather than diagnose are shorter and simpler.

PERIPHERAL AVIONICS SYSTEMS

Learning Objective: *Identify peripheral avionics systems and describe their interaction with the computer.*

The aircraft computer is considered the most important avionics system in achieving the mission of the aircraft. However, the success of the computer depends upon its external sensors or other avionics systems. The quality of data fed to the computer determines the quality of data fed out of the computer. The following avionics systems provide inputs to and receive outputs from the computer: navigation, radar, ordnance/weapons, and data link. These are only a few of the major aircraft avionics systems that interface with the airborne computer.

NAVIGATION

Navigation systems are designed to tell pilots where they are, where they have been, and where they are going. The TACAN/DME system provides known station reference points, while an inertial navigation system provides continuous updating of such information as latitude and longitude. This information is fed to the computer where it is compared, updated, and sent out to other systems.

SEARCH/TRACK RADAR

A search radar system is designed to give visual indications of what is around the aircraft. Some of the present-day aircraft have a 150-mile or greater range. Depending upon the size and/or speed of the radar indications, a computer can determine whether the target is stationary or moving, a land mass or an aircraft, friendly or unfriendly, and many other items of information. If a target is determined to be unfriendly, a tracking radar can be used to tell the pilot what to do to eliminate the target.

ORDNANCE/WEAPONS

The design characteristics and ballistics of the many types of ordnance, weapons, and missiles require the use of a computer to store the information. The airborne computer aids the pilot by telling him/her when to release the weapons. The computer greatly increases the pilot's chances of destroying designated targets.

DATA LINK

Combat aircraft have to have the most up-to-date information available to successfully complete combat missions. On an aircraft carrier, the combat information center, CIC, is normally in constant contact with an airborne CIC. The airborne CIC is usually an E-2 or P-3 aircraft. These two CICs will crosstalk by use of the data link system. Basically, data link involves a series of transmitted pulses that represent information. The pulsed information is sent to the computers of all combat aircraft to enhance their chances of success.

REVIEW QUESTIONS

Q1. What are some examples of computer hardware?

Q2. True or False. An analog computer designed to measure fuel quantity can be used to measure elapsed time in flight?

Q2. In a digital compute, how are some of the numerical equivalents expressed?

Q4. What are the three basic units of a central processor?

Q5. What is the most frequently used technique for writing and reading data in a magnetic core array memory?

Q6. Representative magnetic drums have diameters in what size range?

Q7. What method of inputting or outputting data is considered the fastest?

Q8. Computer designers must face a tradeoff between what two choices when using ICs?

Q9. What method is used to preclude the possibility of a computer program from using an excessive amount of memory?

Q10. What are the six steps of writing a computer program?