GTE LENKURT

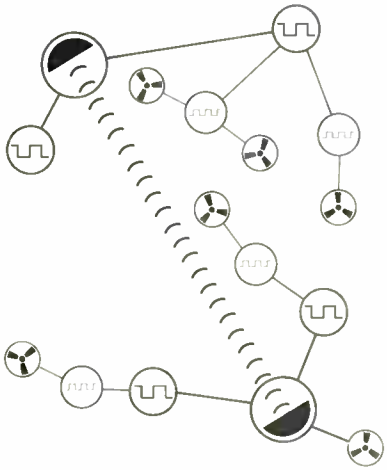
DEMODULATOR

JANUARY/FEBRUARY 1977

Digital Data System

Also in this issue:

Introduction to Microprocessors



Digital data transmission has grown to such an extent that complete networks dedicated entirely to its use are beginning to appear. One such system now in operation provides twenty-four cities with digital data links.

The transmission of digital data over telephone company lines began in the 1950's, but its big boom followed the Federal Communication Commission's 1968 decision to open the voice-frequency (vf) network to the use of non-Bell data equipment. The characteristics of the standard 4-kHz vf channels established the parameters for the design of all manufacturers' data modems and data sets, and helped to standardize certain data rates.

Transmitting digital information over vf channels requires the use of carrier modulation and several methods have been used to great advantage. However, the digital-to-analog conversion process restricts the total amount of data that can be transmitted since a part of the channel bandwidth is always occupied with synchronization information. This factor, plus the burgeoning demands of data transmission, has led to the development of a network designed to handle data in its digital form.

The Digital Data System (DDS) was established in 1974 when five cities—New York, Boston, Philadelphia, Chicago and Washington, D.C.—were interconnected and tests conducted to prove the viability of the system. During 1975, another 19 cities were added to the network, and plans call for 96 metropolitan areas to be linked within a few years.

System Design

The Digital Data System provides private line, point-to-point, and multipoint channels operating at 2.4, 4.8, 9.6 and 56 kb/s with a lower error rate than most dedicated lines or the Direct Distance Dialing (DDD) network. In addition, the inclusion of automatic protective features in both the transmitting and terminal equipment makes the system less susceptible to outages.

From the beginning, system design has focused on two performance aspects: the quality of transmission and the dependability of the channels. The DDS system has been designed with a quality objective of 99.5 percent error-free seconds, while channel availability has been set for 99.96 percent, or an average annual down-time of less than 0.04 percent.

Digital data customers are connected through four-wire loops to local central offices where data signals from many customers are multiplexed onto T1-type repeatered lines for

transmission to a hub office. When the multiplexed signal arrives at a hub office, individual signals are recovered from the bit stream by demultiplexing, then remultiplexed with other data going in the same long-haul direction. Hub offices are interconnected through the existing network of digital carrier systems consisting primarily of microwave radio links using the Data Under Voice (DUV) technique (see Figure 1). Long-haul digital data transmission may also be over T1-type lines up to the maximum range restriction of about 100 miles, or over the new 96-channel, T2-type lines ranging up to 500 miles in length.

Synchronization of the network is critical to the proper functioning of the Digital Data System. In order to maintain synchronization, the data links themselves are used to distribute synchronizing signals throughout the system. Hub offices contain 'nodal' timing supplies having highly stable oscillators with memory circuits that permit them to continue operation in case the synchronization signal is lost. One hub office, the location of which may be changed, functions as a master timing source and all other hub offices are slave units. When the plan is fully implemented, a master source located at Hillsborough, Missouri, will synchronize the entire network. Intermediate and local offices also contain timing sources but their oscillators are not as stable as those in 'nodal' supplies.

Digital Serving Area (DSA)

In its final configuration, the DDS network will contain three levels, or classes, of "digital serving areas." Each DSA will consist of the geographical area around a hub office and will be designated as a Class I, II, or III DSA depending on the class of the hub office. The highest level regional hub offices (Class I) are located along major radio and cable routes and are characterized by large cross-section (high-density) transmission capabilities. Class II sectional hub offices have links with only one Class I hub and act as the center of Class II DSA's. Metropolitan hub offices are linked to only one sectional (Class II) hub and are designated Class III digital serving areas (see Figure 2).

Customer Premises Equipment

Two new data interfaces, a data service unit (DSU) and a channel

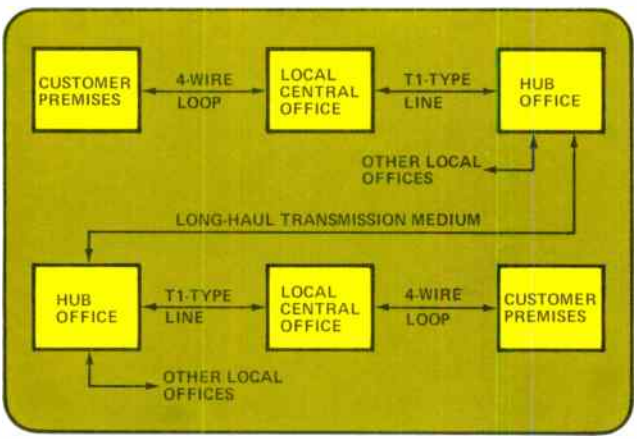


Figure 1. Point-to-point DDS service provides an efficient path for data transmission without the need for cumbersome conversion equipment.

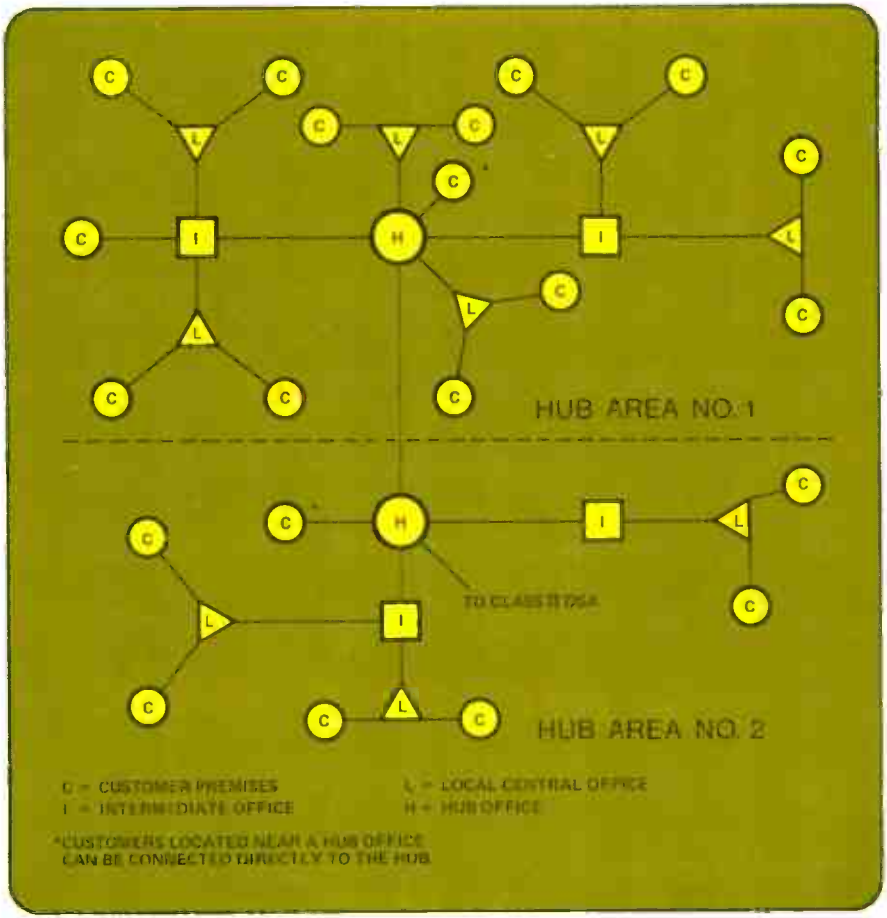


Figure 2. A typical Class III digital serving area. Combined geographical serving areas need not be contiguous and may overlap state and associated telephone company boundaries.

service unit (CSU), have replaced the data modems and data sets which have customarily provided the digital/analog interface at the customer's end of a data transmission facility. With DDS, the digital-to-analog conversion is no longer necessary; consequently, DSU's and CSU's are more compact and simpler in operation than their analog counterparts. The design of these devices, as well as digital multiplexers, has been constrained by established data rates which, as noted earlier, resulted from the characteristics of the vf channels.

A data service unit, such as the GTE Lenkurt L500A, has two basic parts with separate functions. The channel terminator portion of a DSU terminates the four-wire loops and provides circuit paths for loopback tests. The encoder-decoder section contains a transmitter, a receiver and circuits to recover timing signals. In

addition, it provides the familiar EIA RS232 interface at 2.4, 4.8 and 9.6 kb/s and a CCITT V.35 interface at 56 kb/s. The DSU uses a bipolar format to interface the customer's data terminal with the four-wire loop leading to the central office.

A channel service unit, like GTE Lenkurt's L550A, requires that the customer's data terminal equipment provide encoding/decoding and logic functions as well as extraction of timing pulses from the bipolar signal. The primary functions of the CSU are the same as the channel terminator portion of a DSU; i.e., line termination and provision for loopback tests. Both the DSU and CSU provide network protection and automatic line build-out (ALBO) features.

The maximum length of a four-wire loop over which DDS service may be provided using DSU's or CSU's is determined by the type and gauge of the cable, and by the data transmission rate.

Central Office Equipment

The four-wire customer loops are terminated at the central office by an office channel unit (OCU) which forms the data signals into "bytes," provides timing, replaces any zeros that have been removed and detects idle codes generated by a DSU or its equivalent.

The signal coming from the customer's equipment is regenerated and organized into eight-bit bytes with six bits of each byte at the three lower speeds (2.4, 4.8 and 9.6 kb/s), and seven bits of each byte at 56 kb/s, representing customer data. At the three lower speeds, the first bit of each byte is reserved as a framing bit for a subrate digital multiplexer (SRDM). Since framing bits are not needed at 56 kb/s, the first seven bits are all available for transmitting data. The eighth bit of each byte at all speeds is used as a status indicator.

The OCU converts all data rates to 64 kb/s (by repeating bytes) which permits the use of standard test equipment and makes cross-connecting easier. This will also simplify the design of switching equipment for switched DDS service. The OCU generates bipolar violations when a long series of zeros is encountered and produces appropriate codes when the "idle code," "network trouble code," or "remote test code" is present. An "idle code" indicates that the far end terminal is not sending data, and a "network trouble code" indicates a failure in the DDS network. A "remote test code" is generated by test procedures and triggers the automatic loopback circuit in the DSU.

Besides OCU's for all four data speeds, each central office may contain timing supplies, multiplexers, multipoint junction units (MJU's) and test equipment. Hub offices contain all of the above equipment plus additional cross-connect facilities (DSX-0, A and B), system test facilities, and a means to access the long-haul transmission facilities. The MJU's, which are located at the DSX-0, permit several terminals at different locations to share a common transmission channel in a multipoint private line.

Multiplex Equipment

Local and hub offices may be equipped with three types of time-division multiplexers to combine the outputs of the OCU's.

The lowest-order multiplexer is the subrate digital multiplexer (SRDM), or digital submultiplexer, which assembles the eight-bit bytes from the OCU's into superframes containing 5 bytes for 9.6 kb/s, 10 bytes for 4.8 kb/s and 20 bytes for 2.4 kb/s (see Figure 3). The 4.8- and 9.6-kb/s

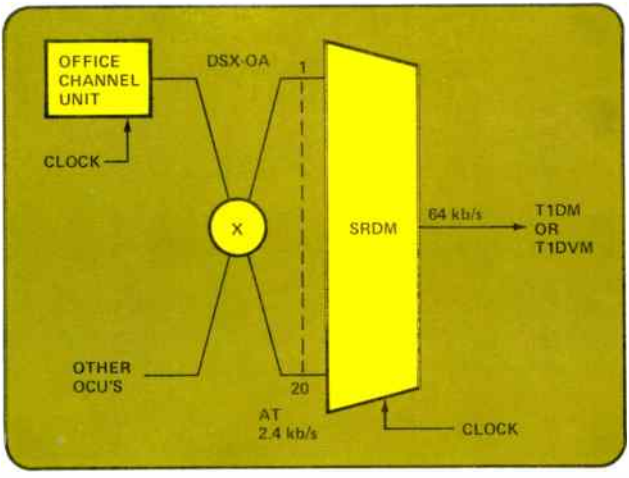


Figure 3. The output signal of a subrate digital multiplexer (SRDM) contains synchronization information which permits it to operate independently of second-stage multiplexers.

SRDM's can accept signals from OCU's operating at either 2.4 or 4.8 kb/s, and the 9.6-kb/s multiplexer can accept any of the three low-speed OCU signals, although there is a loss of efficiency when a submultiplexer is used at less than its maximum rate. All three versions of the digital submultiplexer have 64-kb/s outputs which connect to one port of a higher-order multiplexer.

The T1 Digital Multiplexer (T1DM) is a second-stage multiplexer which combines the output of 23 first-stage submultiplexers into a 1.544-Mb/s bit stream for transmission over T1-type lines (see Figure 4). The standard T1-type PCM line frame length of 193 bits is retained. Since the output signals from all three versions of the digital submultiplexer and all four versions of OCU's are identical 64-kb/s bit streams, any port of the T1DM can accept an input from any one of these seven pieces of equipment or a combination of signals from them. Efficiency drops whenever a port is not connected to a 56-kb/s OCU or a fully loaded submultiplexer.

An additional second-stage multiplexer, called a Data-Voice Multiplexer (T1DVM), allows 64-kb/s data signals and digitized voice circuits to share a T1-type line when used with a D1D- or D3-type channel bank (the GTE Lenkurt 9002B, for example). The T1DVM (see Figure 5) was developed for situations where the volume of data traffic does not justify the full capacity of a T1DM operating over a dedicated T1-type line. The T1DVM can replace from one to twelve 64-kb/s digital voice signals from a channel bank with a like number of 64-kb/s data signals. Each 64-kb/s data stream derived from the T1DVM can be used in the same way as those provided by the T1DM digital multiplexer. The PCM channel bank used in conjunction with the T1DVM must operate synchronously with the DDS network and is, therefore, controlled by the office clock.

The T1DVM frame format is identical to the D1D- and D3-type format, consisting of 193 bits divided into 24 eight-bit bytes. The eighth bit in each byte is set to logic 1 when the customer's data terminal is transmitting, otherwise it is set to 0. Any byte in which the eighth bit is a 0 is called a "control byte."

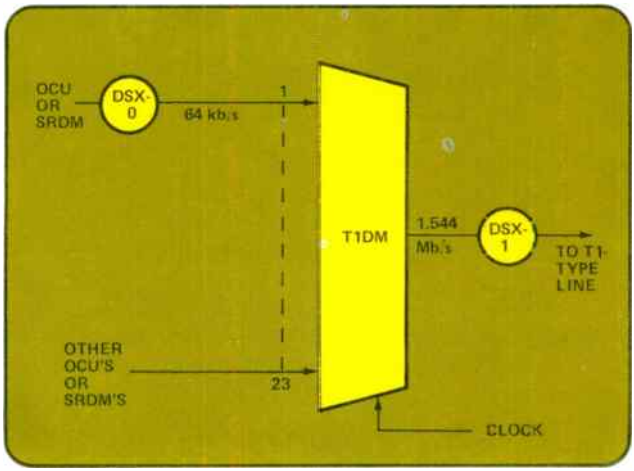


Figure 4. The T1 Digital Multiplexer (T1DM) designed for high-density data routes.

The T1DVM can operate in three modes: combined data-voice, independent data, or chained operation in which data can be added or dropped at intermediate offices without back-to-back terminals.

All these multiplexers contain not only monitors and alarms to indicate outages, but also standby spares which can automatically restore service when the operating unit fails.

Long-Haul Data Links

Primary use is made of the existing microwave radio network employing the Data Under Voice (DUV) technique which encodes a T1-type signal into a seven-level partial response format at the transmit terminal and decodes it at the receive end. Most metropolitan areas already have access to this network so no major expenditures are involved in providing DDS

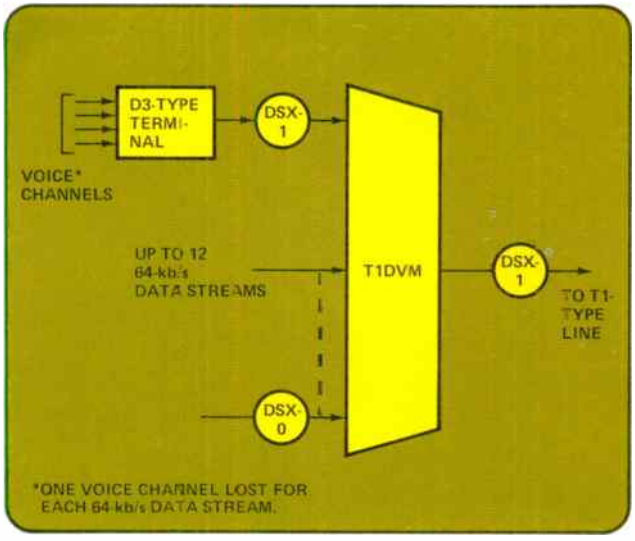


Figure 5. T1 Data-Voice Multiplexer (T1DVM) designed for low-density data routes permits substitution of 1 to 12 data streams for digitized voice signals.

service. However, the limited number of channels available has led to the development of several high-capacity systems, some of which have not yet been put into service.

For example, the M12 multiplexer can combine up to four 24-channel signals into a 6.312-Mb/s format for transmission over the new T2-type digital line, while the M21 multiplexer is capable of combining two T2-type signals into a 13-Mb/s signal for transmission over L4- or L5-type coaxial cable carrier systems. This 13-Mb/s bit stream replaces a mastergroup (600 voice channels) and requires regeneration about every 300 miles. Attempts are also being made to put as many as 4032 one-way voice channels on one coaxial cable. Adequately synchronized, such a system could transmit 200 Mb/s.

Future systems composed of fiber optics, millimeter waveguide, digital radio or PCM coaxial cable will be ideally suited for DDS, and the economies of handling digital data in its original form will become more evident as experience is gained with these new systems.

BIBLIOGRAPHY

1. "Expansion of AT&T DDS Net Limited to 24 Service Areas," *Telephony*, Vol. 191, No. 8 (August 23, 1976), 14.

2. Hanna, D.L. *DDS Planning Guide*. Stamford, Conn.: GTE Service Corp, November 1972.

3. Kretzmer, E.R. "The New Look in Data Communications," *Bell Laboratories Record*, Vol. 51, No. 9 (October 1973), 259.

4. McKay, K.G. "Digital Communications — A Tutorial," *Bell Laboratories Record*, Vol. 49, No. 9 (October 1971), 279-84.

5. Moster, C.R. and L.R. Pamm. "The Digital Data System Launches a New Era in Communications," *Bell Laboratories Record*, Vol. 53, No. 11 (December 1975), 420-26.

6. *The Bell System Technical Journal*, Vol. 54, No. 5 (May/June 1975).

Introduction to Microprocessors

Because of its increasing utilization of digital technology, the telecommunications industry has come to rely more than ever on devices and processes originally developed for the data processing field.

A digital computer is basically a stored-program machine that automatically performs computations and solves problems too lengthy or complex to be easily handled by any other means. In the years since they were first introduced, they have become increasingly efficient, and more widely used, with every technological improvement. The microcomputer, which represents the latest developments, realizes the capabilities of a digital computer in an integrated circuit form.

Computer Architecture

At the very least, a computer must contain, or have access to, a memory, an input/output (I/O) interface, a device or circuit for performing arithmetic operations, and a control section.

The memory of a digital computer allows storage and retrieval of information in the form of binary digits, or "bits," that are typically arranged into "words." To ease manipulation, each word may be subdivided into "bytes"; a 16-bit word may thus be treated as two eight-bit bytes. The information stored is of two types: the data that are to be processed and instructions that, collectively, constitute a "program."

Each storage location in a memory is identified by a particular number, or "address." For example, in one type of random access memory (RAM) structure (see Figure 1), using a solid-state memory chip and an eight-bit word format, an address word provides access to any given location. The first four, or highest-order, bits are decoded

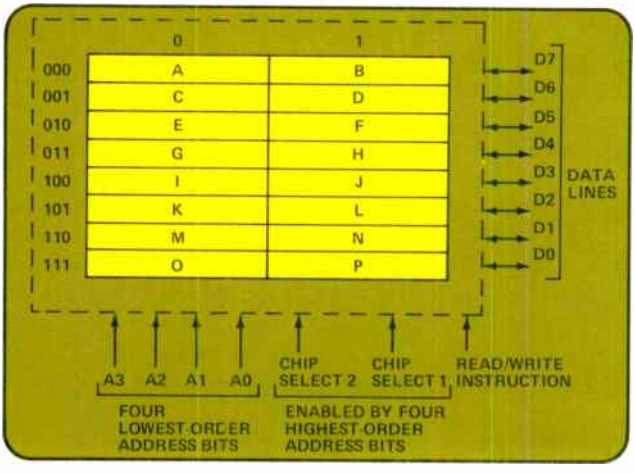


Figure 1. The unique address of each cell in a random access memory (RAM) structure allows the random recall or storage of data bits. In a read only memory (ROM) structure, data can only be recalled, but addressing is the same.

to serve as enabling signals for the group of memory cells. Once the chip is enabled, the four lower-order bits identify the specific cell being addressed. If the system were in a read mode, the contents of the addressed cell would be placed on the data lines; in a write mode the word appearing on the data lines would be stored in the addressed cell. When a read only memory (ROM) structure is employed, addressing can be accomplished in the same manner, the only difference being that no write capability is provided. Program instructions, which rarely change, are generally stored in ROM's, while RAM's are used for storing transient data.

A program instruction consists of an "operator," which details the operation to be performed, and at least one "operand," which may indicate the location of the data to be manipulated, or may itself be the data. Programs are typically written so that sequential instructions are contained in consecutive cells. For example, the contents of program memory cell C, identified by address 0010, might command the computer to transfer (operator) to a temporary storage facility, or register, the contents (operand) of one particular data memory location. The instruction in cell D, address 0011, might then command that this data be compared with the contents of another location and the word in cell E, address 0100, cause the result to be stored in yet another data memory cell.

Communication with the external world is provided by one or more input/output interfaces, or ports. As with the memory locations, the I/O ports must be addressable, allowing the system to select a source of information, such as a keyboard, or a recipient of the processed data (printer, magnetic tape, etc.).

Typically, those portions of a digital computer that perform the control and arithmetic functions are treated as a single "central processing unit" (CPU). The microprocessor is essentially a miniaturized CPU which, when interconnected with peripheral devices, performs the logical and control functions of a microcomputer (see Figure 2). Execution of a program instruction in such a system is initiated by an activity command (I/O port read, memory write, etc.) being placed

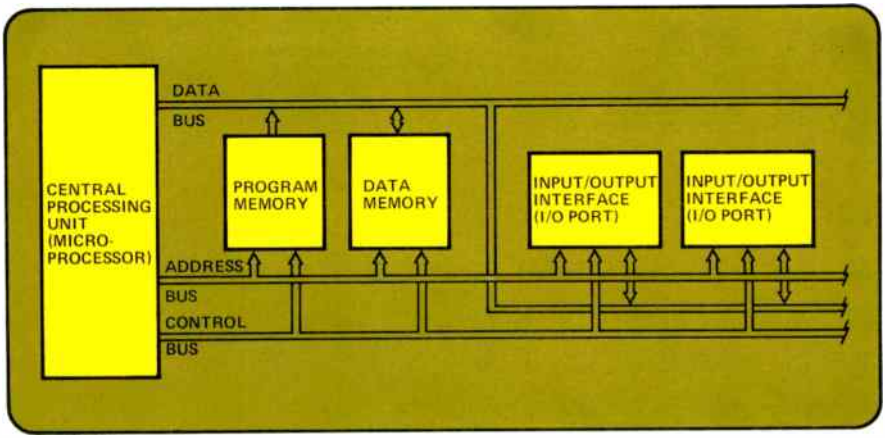


Figure 2. A microprocessor becomes useful only when it is associated with peripheral devices to form what is essentially a microcomputer.

on the control bus by the CPU. The binary-coded address of the required memory location or I/O port is then placed on the address bus. Data is exchanged over the data bus, after which the CPU issues the next activity command.

Microprocessor Architecture

A typical microprocessor (see Figure 3) consists of an arithmetic/logic unit (ALU), control circuitry, and a number of registers. Registers are temporary storage units, some of which are dedicated to specific functions while others serve more general purposes. The "accumulator" of a microprocessor, for example, is a register that interconnects the other elements.

Data arriving from an input port are initially entered into the accumulator, from which they are either moved to the internal data bus for transfer to the data memory, or entered into the ALU for processing. Data intended for an output port are also loaded into the accumulator while awaiting an output instruction. The accumulator is thus both an operand and a result register; that is, it stores an operand until needed by the ALU and, after the arithmetic/logic operation is performed, temporarily retains the result.

The simplest arithmetic/logic unit is an "adder" that combines two inputs to produce an output according to the logic of binary arithmetic. With such a basic circuit, operating routines can be written to produce subtraction, multiplication and division capabilities. Included in the ALU are "flag bits" that indicate certain conditions (carry, overflow, zero, etc.) that occur during arithmetic and logical processing.

A group of general-purpose registers serves as an easily accessible "scratchpad" memory within the microprocessor, eliminating the need to shuttle intermediate data between accumulator and memory. The types of data stored in these registers depend upon the processor's operating mode and

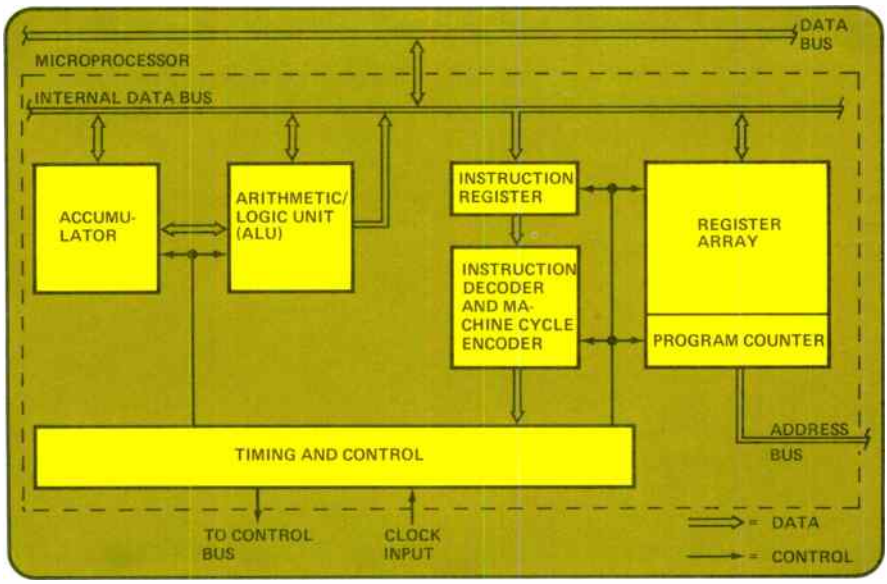


Figure 3. The structure of a microprocessor typically includes an arithmetic/logic unit, registers, and control circuitry.

may include, in addition to intermediate processing results, address information and instruction bytes.

Yet another special-purpose register is the "program counter," which allows the processor to know in what memory location the next instruction lies. The program counter at any given time contains the memory address of the next instruction. Because of the way in which programs are normally written, the processor can maintain this up-to-date status by adding a logic 1 to the contents of the counter immediately after the fetching of an instruction. This sequential rule is violated only when the fetched instruction commands a "jump," causing the next instruction to be read from a memory location specified by the program rather than from the next sequential location.

The program counter is an essential element of the microprocessor's control section, which also includes an "instruction register," an "instruction decoder and machine cycle encoder," and control circuitry that uses clock inputs to maintain the proper sequence of events.

When an instruction word is fetched from the program memory, its first byte is stored in the instruction register; if the word contains more than one byte, succeeding bytes are placed in the general-purpose register array. The register's contents are retained until the required operation has been executed, and are also available to the decoder. The decoder logically translates the binary-coded information to selectively activate one of a number of output lines, each of which represents a series of operations associated with the execution of the instruction. The output of the decoder is combined with timing information to produce the signals that control the other processor elements.

Microprocessor Cycles

The operation of a microprocessor is based on the instruction cycle, which encompasses the time required to fetch an instruction from the program and to execute it. The instruction cycle may be further divided into any number of "machine," or "processor," cycles. Every microprocessor operation requires at least one reference to memory, even if it is only to fetch the next sequential instruction word; one such fetch phase is possible during any given machine cycle, while the number of machine cycles occupied by an execute phase depends upon what activity is to be carried out.

For example, when the one-byte instruction word, "add register B," is fetched from memory, it requires only that the contents of one general-purpose register, whose address is included in the single byte, be added to the contents of the accumulator. All of the information needed for execution is contained within the processor's temporary storage facilities, so no further reference to memory is required and the execute phase can begin immediately; the instruction cycle thus occupies one machine cycle (see Figure 4A). This is a "register" mode of address, in that the register location of the data to be manipulated is specified by the instruction. If, however, the instruction word commands that the contents of a particular data memory location be added to the accumulator, the instruction cycle occupies three machine cycles (see Figure 4B). In this "register indirect" mode of address, the instruction must specify the register containing the address of the memory location in which the desired data are stored. During the first machine cycle, the instruction code for the addition operation and the register location are fetched from the program memory. During the sec-

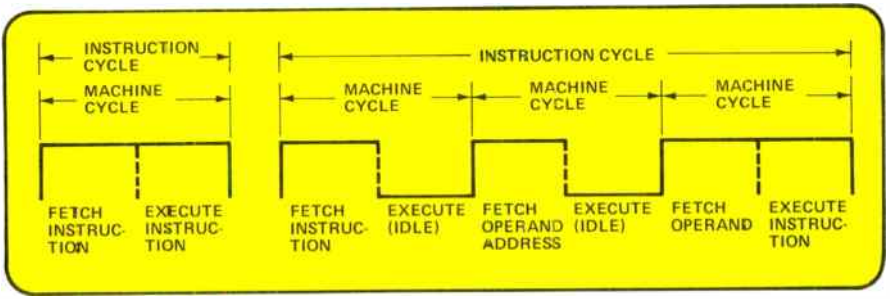


Figure 4. The instruction cycle of a microprocessor consists of one or more machine cycles, depending on the type of activity to be carried out.

ond cycle, the address of the data memory location is read out of the register, completing the instruction. The data are read out of the addressed location during the fetch phase of the third cycle and added to the accumulator contents during the third-cycle execute phase.

Two other address modes are usually available to the microprocessor. In one, the "direct" mode, a multi-byte instruction word contains the operation code and the address of the desired data memory location. When the address mode is "immediate," the data to be manipulated are actually contained within the multi-byte instruction word (see Figure 5).

The sequential movement of data – including instruction bytes – among storage, computing and I/O elements is the normal operational mode of the microcomputers that are built around microprocessors. In another mode, however, the microprocessor responds to "interrupt request" signals. When such a request is received, it replaces the next sequential instruction address held in the program counter with the address of an "interrupt subroutine." The instruction being executed is then completed and the interrupt subroutine fetched. After execution of this subroutine, the normal instruction sequence is resumed. In many applications, more than one interrupt subroutine is available to the processor, depending upon the source of the request. Such a multiple-interrupt system typically accepts inputs from several devices and assigns priorities for obtaining access to the subroutines.

Instruction Sets

As with any computer, a microprocessor-based system can only do what it is "told" to do by its program. When it is designed, a microprocessor is given the ability to perform a certain set of operations that define its "instruction set"; every time a particular instruction is decoded, the same specific operation is carried out. This makes the microprocessor extremely flexible, allowing its entire system application to be altered simply by changing the program memory contents.

An instruction set generally includes data transfer and storage, logic, arithmetic, branch and input/output instructions. The group of data transfer and storage instructions controls the movement of data between the various register and memory elements, including such functions as moving register contents to and from memory, loading the accumulator and reading it into memory, and exchanging the stored data among the general-purpose registers. The arithmetic group per-

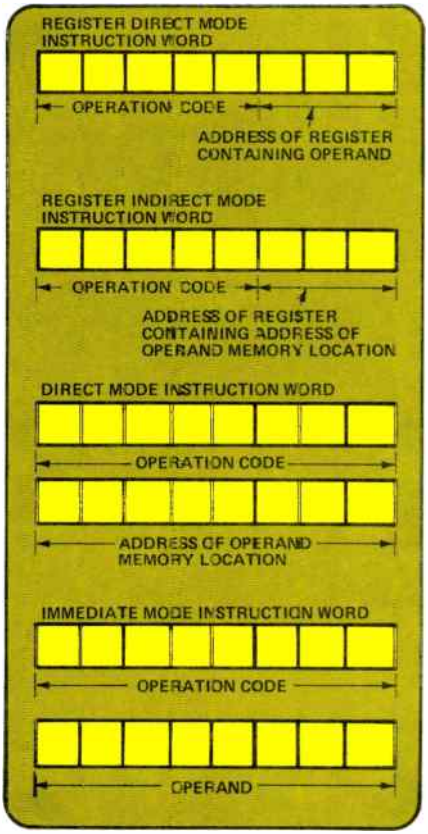


Figure 5. A microprocessor program may utilize several address modes to gain access to operands.

forms such operations as addition and subtraction on data in the registers and memory. Boolean algebraic operations (OR, AND, compare, rotate, etc.) are controlled by the logic instructions, while the exchange of data with I/O ports and external devices depends upon the input/output group of instructions. Branch instructions alter the normal sequential program progression and can be either conditional or unconditional. An unconditional jump, for example, would simply comply with the operation specified by the data fetched from memory; "jump to memory location B" would be an unconditional branch instruction. The command, "if one condition is true, transfer program control to the instruction whose address is specified," would constitute a conditional jump.

Languages

A microprocessor can only execute instruction set commands that are in a binary-coded "machine language." For example, the instruction, "transfer the contents of data memory location 3 to the accumulator," might be identified by the binary number 00111010. When handling such relatively long series of 1's and 0's, however, humans tend to make errors, so "program languages" have been developed to make programming more practical.

Most such languages use "mnemonic" (memory-aiding) abbreviations. One of these languages is "assembly language," which assigns a unique mnemonic to each instruction. Every microprocessor manufacturer develops a special assembly language, but the principle is the same: the data-transfer example given above, which is a "loading" type of instruction, is assigned the mnemonic "LDA addr," followed by the binary-coded address of memory location 3, by one manufacturer, but could be identified in a different manner by another.

A microprocessor program writer can put together a group of instructions using an assembly language and, using an "assembler program," convert it to the requisite machine language. Higher-level programming languages such as ALGOL, PLM and FORTRAN allow several instructions to be combined into a statement which is then identified by mnemonics.

Applications

While the use of microprocessors has spread throughout many industries, there are some applications that

are of particular interest to those in the telecommunications field. For example, an alarm system using traditional techniques might compare an input at one time with a previously received signal; if variations exceed pre-determined limits, an alarm mechanism is activated. Such a system would, hopefully, never have to perform any function other than comparison of inputs. This means, however, that until an alarm condition is detected the logic and control circuitry dedicated to the system is doing no useful work.

The same alarm operation can be performed with a microprocessor-based network that, in many cases, affords several advantages. Not only would the microprocessor compare inputs with previously received data stored in its memory, but during the inter-alarm periods could perform other, useful, work. When associated with a pipeline, for example, flow rate monitoring and control could be a primary responsibility of the network; if an alarm condition were detected, the I/O device serving that purpose would generate an interrupt request signal and cause an alarm-activating subroutine to be fetched and executed.

Telephone switching system control applications are also readily adaptable to microprocessor implementation. There are, in fact, several switching facilities now available that rely upon microprocessor-based controllers to perform line and trunk scanning, assign voice channels and, in general, coordinate all of the necessary system decision making.

Embodying as they do many of the concepts of larger-scale digital computers, microcomputers – and the microprocessors around which they center – differ in that their micro-instruction format allows control of data transfers, and all other functions, to reside in computed results of previously processed data. The instruction sets designed into specific processors permit "custom tailoring" of systems to applications by the simple expedient of changing the contents of the program memory. The fast, reliable and versatile configurations possible with microprocessors make them ideal for the rapidly changing telecommunications industry, and their increased utilization is forseen.

BIBLIOGRAPHY

1. Barna, A. and D.I. Porat. *Introduction to Microcomputers and Microprocessors.* New York: John Wiley & Sons, 1976.

2. *8080 Microcomputer Systems User's Manual.* Santa Clara, Calif.: Intel Corp., 1975.

3. Gray, B.H. "Impact of Microcomputers on System Design," *Proceedings 1975 IEEE International Symposium on Circuits and Systems.* New York: The Institute of Electrical and Electronics Engineers, Inc., 1975.

4. Hoertt, D.J. "Microcircuits: The Brain and the Brawn," *IEEE Transactions on Industrial Electronics and Control Instrumentation,* Vol. IECI-22, No. 2 (May 1975), 159 – 163.

5. Martin, D.P. *Microcomputer Design.* Northbrook, Ill.: Martin Research, 1976.

6. *Microprogramming Handbook.* Santa Ana, Calif.: Microdata Corp., 1972.

GTE LENKURT
1105 COUNTY ROAD
SAN CARLOS, CALIFORNIA 94070

ADDRESS CORRECTION REQUESTED

Bulk Rate
U. S. Postage
PAID
San Carlos, CA
Permit No. 37

The GTE Lenkurt Demodulator is circulated bimonthly to selected technicians, engineers and managers employed by companies or government agencies who use and operate communications systems, and to educational institutions. Permission to reprint granted on request.