No. 1 ESS: System Organization and Objectives

By W. KEISTER, R. W. KETCHLEDGE and H. E. VAUGHAN

(Manuscript received January 22, 1964)

This paper is an introduction to the No. 1 electronic switching system, a new general-purpose switching system developed for use in the Bell System. Organization and objectives of the system are outlined to provide a background for the detailed technical papers which follow.

I. INTRODUCTION

1.1 *General*

The continuing expansion of present-day services, the demand for additional types of service, and the anticipation of new service offerings in the future all indicate a need for a new general-purpose switching system for the Bell System. The No. 1 electronic switching system (ESS) has been developed to meet this need. It is a general-purpose telephone switching machine capable of providing either two-wire or four-wire switching for local, toll, or tandem applications. No. 1 ESS provides today's services as well as several new ones, and is sufficiently flexible to provide for the ever-changing need for future services. It is economically competitive for present applications and will provide new services at relatively low cost.

The purpose of this paper is to outline the objectives of the system and to provide a brief description as background for the several more

technical and more detailed papers which follow. Switching systems are, in general, complex and not easy to describe. No. 1 ESS not only entails many new concepts of switching system organization, but the equipment and apparatus used are also different from those of previous switching systems. As a consequence of this, No. 1 ESS is the largest development project ever undertaken by Bell Laboratories for the Bell System. The papers in this issue should give an adequate description of all the major concepts and techniques used in the system. Other papers, particularly those treating programming and testing aspects, will be published in forthcoming issues of this and other journals.

1.2 *History*

Since 1945, Bell Laboratories has conducted an active program of research and exploratory development in electronic switching. The Morris, Illinois, trial[1] of the world's first electronic switching office demonstrated the value of many of the basic concepts used in No. 1 ESS, in particular the use of a stored program and the basic maintenance philosophy. The system at Morris provided customers regular commercial-grade service and some additional new services. The trial established valuable guides for production design in hardware, data formats and programming. In addition, the trial also provided excellent training for many young engineers who later had a major part in the design of the No. 1 ESS. Although all of the hardware units and data processing programs for No. 1 ESS are new, their design relied heavily on the experience gained at Morris. The development of No. 1 ESS has now relegated the Morris system to history.

II. OBJECTIVES

2.1 *Economics*

Essential objectives for a new switching system are that it be at least the equivalent in service features of existing systems and be economically competitive with these systems for some significant segment of the market. These objectives must be attained on the basis of today's market rather than some hypothetical market of the future. The new system must be introduced in an environment of a very large number of older systems, and the general capability of the over-all switching network will be determined largely by the capability of the majority of the offices. The process of achieving improvements in the over-all system must be one of evolution over a period of years. However, the new

system must have greater capability than the older systems in order that this improvement can be realized as the percentage of new offices increases. This points up another important essential. A new office must be compatible with the older offices with which it must operate, since modification of older units in service is inconceivable if we would attain the economic advantages of a new office design.

The economic balance over a range of sizes in the No. 1 ESS plan is quite different from that in electromechanical common control offices such as crossbar. The electronic system is based on a single high-speed central processor which is essentially the same in both large and small offices. In a system like No. 5 crossbar, a multiplicity of control units must be used due to their slower speed. This allows the amount of control equipment — as, for instance, the number of registers and markers in the crossbar system — to be increased as the office grows, so that the full burden of a control capable of handling a large office need not be borne by a small initial installation.

The new system must meet the standards of reliability that have been established by electromechanical systems. The operation of a complete office by a single central processor is a definite problem when reliability is considered. In a multi-marker office of No. 5 crossbar, the failure of a single marker merely reduces the traffic capacity of the office. However, the failure of the controller in an office depending on a single central processor would make the office completely inoperative. The solution followed in No. 1 ESS is to duplicate all units essential to proper operation of the office.

The economic objectives of No. 1 ESS are being realized through a basic design which has been optimized in its details and will be economical in production and in operation over a period of years. The economics of the equipment design have been based on quantity manufacture of its component parts. By the use of the stored program, it has been possible to plan a system which requires no wired options during manufacture and therefore leads to efficiency in production. The system units are designed for a minimum of interconnections between frames, so that most of the wiring and much of the detailed testing can be done at the factory rather than on location during the process of installation. Trouble detection and fault location have been highly automated through the use of stored program. Since most of the system logic involving telephone service features has been placed in the stored program, the introduction of new service features and modifications of existing features will be greatly simplified. In many cases, modifications will be made by changing programs rather than by wiring changes.

It is believed that important savings will be realized in the current development effort required to maintain the system design over a period of years during which its service features and operational environment change.

2.2 *Size*

An important item in planning the system was to determine the range of sizes over which the system would be applicable. Two major parts of the system are greatly influenced by size. One is the network, which must be planned to serve the entire range, with systematic growth from the smallest to the largest size encompassed by the design. The other is the central processor, whose call handling capability must be sufficient to care for the largest office while also being economical when used at only a fraction of its total capacity in smaller offices.

The appropriate size range was determined by making a survey of the range of office sizes. Information for this survey was obtained from the operating telephone companies, who provided information on the number of wire centers and the size of each wire center in each company. This information, covering a total of over thirty million lines in the Bell System, was analyzed by a digital computer to provide statistical information on the makeup of the potential market. The general nature of the results is indicated in Tables I and II. From these it is apparent that there are a very large number of very small wire centers, but that there is a large volume of business in large wire centers. Fifty per cent of the total lines are served from wire centers of 19,000 lines or greater. It was decided that it was not necessary to design an office large enough to accommodate the largest wire centers, because the differential cost per line becomes negligibly small in the higher office sizes. For example, the cost of two 50,000-line units will not be significantly more than the cost of a single 100,000-line unit. For this reason, a decision was made and verified by a number of studies that an upper size limit of 65,000 lines would be reasonable. The lower size limit is determined purely by

TABLE I — TOTAL BELL SYSTEM LINES IN SERVICE VS CENTRAL OFFICE SIZE*

Per Cent of Total No. of Lines	Office Size
75	over 7,500
50	over 19,000
25	over 32,000

* Data as of January 1, 1960 (includes community dial offices).

TABLE II — CENTRAL OFFICE BUILDING TOTALS VS NUMBER OF LINES SERVED*

Per Cent of Total No. CO Buildings	Lines Served
25	less than 230
50	less than 750
75	less than 3000

* Data as of January 1, 1960 (includes community dial offices).

economics, but should extend down to at least the 4,000- to 5,000-line size, since the survey showed that a large number of offices are initially installed in this size range.

From a traffic standpoint, the maximum size of the system is set by the capacity of the central processor for handling calls in real time. The maximum capacity of the No. 1 ESS has been set at 100,000 calls in the busy hour. This figure was determined through studies of the cost and complexity of central processors of various designs, weighed against the traffic needs in the wire centers in the Bell System. The capacity of the central processor is determined by the basic speeds of the electronic circuits, its basic clock cycle time, the complexity of its individual logic operations and the amount of processing done in parallel.

2.3 *Flexibility*

The system has been planned to make maximum use of the flexibility inherent in a stored program. Central control is designed so that its wired logic represents basic logic operations which are related to telephone switching functions only through the sequences of instructions in the program. By means of the stored program, most of the logic decisions in call processing have been converted to basic logic operations. This philosophy is extended to items such as trunk circuits. The physical equipment in a trunk circuit is in most cases limited to that necessary for detecting and generating the signals required on these trunk circuits and performing basic switching operations such as loop closure or loop reversal. All sequencing of these operations, including timing of the duration of signals, is performed by central control under instructions from the program. This has the effect of minimizing the variety of trunk circuits required and reducing their cost, since changes in timing or sequence of operation can be made through changes in the program.

The switching network has been designed for flexibility in a variety of situations. The basic pattern consists of line frames and trunk frames connected together by groups of junctors of three types. Junctors are

provided between line frames and trunk frames for connections between lines and trunks. Junctors from line frames back to line frames are used for line-to-line connections, and likewise junctors between trunk frames are used for tandem traffic and other trunk-to-trunk connections. The number of line frames and trunk frames can be varied independently and the junctor group sizes adjusted according to the mix of inter- and intra-office and tandem traffic. The network has been designed so that frames of four-wire ferreed switches can be used without modification of the central processor complex or its basic network control programs. In this way, the single basic system design can be adapted to a variety of local, toll, and tandem applications which in the past have required quite different system designs.

III. SYSTEM ORGANIZATION

3.1 *Outline of System Plan*

The basic concept of the No. 1 ESS is that of a high-speed electronic central processor operating with a stored program to control the actions of the central office on a time-sharing basis. The general system organization is illustrated in the block diagram (Fig. 1). The switching network provides the means for making interconnections between the lines and trunks to be served by the system and also provides access to the various service circuits required in handling telephone calls. These include tone sources, signaling detectors, ringing sources and the like. All information processing is handled by a central processor consisting of central control and the temporary and semipermanent memories. The temporary memory is used for storage of the transient information required in processing calls, such as the digits dialed by the subscriber or the busy and idle states of lines and trunks. The semipermanent memory contains the stored program and translation information. The contents of this memory need not change during the processing of a call. When the semipermanent information must be changed for any reason, these changes can be made on a manual basis without interfering with the operation of the office. The semipermanent memory also has the advantageous characteristic that its stored information is not erased by circuit malfunctions.

The central control consists of wired logic for performing information processing operations. It is organized on a word basis, with a word length of 24 bits, and operates on a basic cycle time of 5.5 microseconds. It is important to realize that the logic wired in central control is designed for information processing. The telephone switching logic is contained

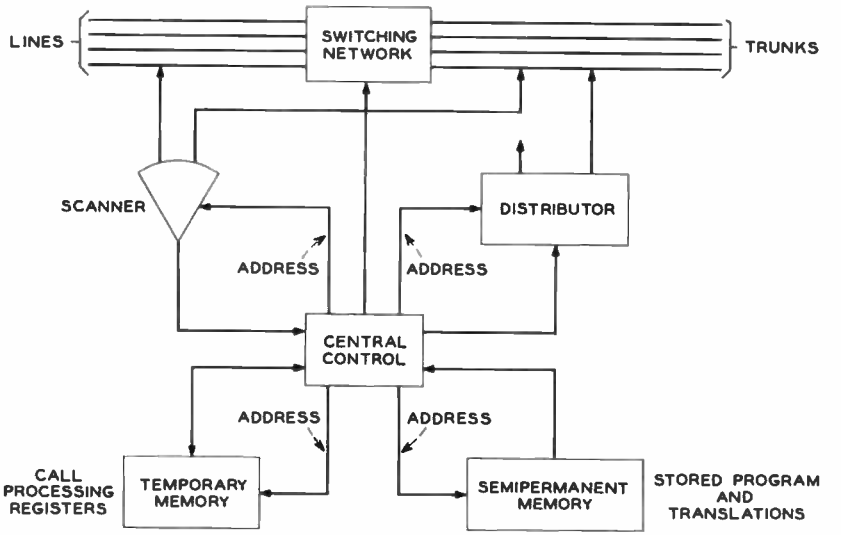


Fig. 1 — Central information processor with stored program (No. 1 ESS).

in the stored program. Thus, the hardware of the control complex is largely independent of the type of telephone service to be provided and the service treatments to be offered to the subscribers. This leads to great flexibility, since the same equipment can be used with different programs to provide a variety of services.

Input information for the central processor is provided by scanners connected to various points in the system where information must be obtained. These include the lines and trunks and signal receivers. The scanners are directed periodically to the lines to detect service requests, to the trunks to detect incoming calls, and to the signal receivers for detecting dialed digits and other control information. The process of recognizing this information involves periodic scanning and recording the results of these scans in temporary memory.

The signal distributor is the inverse of the scanner. It is connected to the various points in the system where actions must be controlled by the central processor. Central control can address the distributor to a particular terminal where a flip-flop or other memory device can be set to start an action. At a later time, central control can address the distributor to terminate this action. In the present design, the system handles this distributor action through two types of units. One is the central pulse distributor, which is all-electronic and is used for high-speed actions. The other, the signal distributor, makes use of a relay tree to perform lower-speed actions such as the control of trunk relays.

Thus the facilities provided in the system can be divided into several categories, such as the switching network with its associated terminal circuits which perform the physical functions required in making connections and detecting and producing signals; the central processor, with its wired logic to perform basic information-processing functions; the scanner and distributor, providing input and output communication for the central processor; and the stored program, which contains instructions for performing all the switching tasks in ordered lists of instruction words.

Particular distinction needs to be made between the program and translation information contained in the semipermanent memory. The program is the lists of instructions for performing all of the service features. It is part of the basic design of the machine and is not influenced by the characteristics of the particular installation. Ordinarily, it will not be changed except for some significant change in features or the sequence in which actions are to be performed, which would correspond to redesign and wiring changes in electromechanical offices. The translation section, on the other hand, contains the specific layout of facilities in the particular office: the association of subscriber directory numbers with the equipment location of their lines, the classes of service to be provided — such as individual and two-party, coin, extended area dialing and so on — and also the specification of trunk routes and their location and alternate routes to be used when available. This information must be changed periodically with the inward and outward movement of subscribers, changes in their class of service, and readjustment of trunk routes. Means are provided for making these changes on a periodic routine basis.

3.2 *Hardware*

The equipment and apparatus of No. 1 ESS has been designed for large-volume manufacture at minimum cost. Note that even a volume of only one million lines a year means over 15 million ferreed crosspoints, $1\frac{1}{2}$ million electronic circuit packages, 5 million transistors, 15 million diodes, etc. Obviously, such production rates must be highly mechanized.

Equipment and apparatus must be of special design to be adaptable to low-cost mechanized manufacture. Further, the high manufacturing volume justifies greater development effort and the creation of special devices and components. Thus No. 1 ESS uses many apparatus items that were developed for this particular application and whose design was tailored for mass manufacture. The ferreed switch is a good example. It is designed to be made as an array, not as an assembly of individual

crosspoints. The control coils are not wound individually and then connected in series. Rather they are wound in simultaneous rows and columns from continuous lengths of wire.

Choice of technology is based on function and cost. In No. 1 ESS, relays are used in substantial quantities. The relay has cost advantages over an electronic approach where speed and function permit. For example, control of ferreed switches involves routing a high-current pulse through a selected path in the matrix of windings. This can be done by means of diodes and PNPN triodes, and indeed such a control structure was developed initially. However, by appropriate design of the control circuits one can, in effect, substitute a relay contact for a high-current diode. The cost advantage of the relay approach is substantial.

Standardization and minimization of codes are other important steps to low-cost volume manufacture. In No. 1 ESS strong efforts were made to standardize the hardware and to achieve the necessary variability by program or translation methods. An example of equipment standardization is the network equipment. Only six codes of frames permit assembly of a switching network suitable for any local office. Two additional codes take care of four-wire networks for toll offices. An example of device standardization is the use of only two codes of transistors, the low-power 29A and the relatively higher-power 20D. The 29A is used in the many logic circuits, in audio amplifiers and oscillators, and in broadband feedback amplifiers and regulators. Another effect of standardization has been the virtual elimination of wired options. Most electromechanical switching equipment makes liberal use of wired options to meet the variations of size and features of different installations. Thus the equipment is specially wired and tested at the factory. In No. 1 ESS the corresponding variations are stored in memory. The factory makes a particular frame the same way every time to a fixed set of test requirements.

The choice of seven-foot-high equipment arrangements (instead of 11.5-foot) eliminates the need for special buildings and allows maintenance from the floor, without ladders. This also contributes to the objective of simplified installation and growth. As a part of the general objective of dependability, and to permit reuse of existing buildings, air conditioning is not required. While most offices will doubtless be air conditioned, it will not be required for machine operability.

No. 1 ESS uses unregulated storage battery power fed by silicon-controlled rectifiers. The equipment has therefore been designed to operate over a ± 10 per cent voltage range. This design eliminates the need for end cells, counter cells and their associated switches.

Transmission has received particular attention. All outgoing trunks include loop compensation networks to improve return loss characteristics. Tones are generated by precision transistor oscillators and are fed to lines and trunks out of precisely balanced terminations. These and other similar measures, including careful control of noise, were taken in recognition of the role of the switching center in the maintenance and improvement of transmission objectives.

3.3 *Programs*

Approximately 90 programs totaling about 100,000 words are used to control the operations required for telephone service and to control the maintenance of the system. These programs, each an ordered set of instructions to provide a particular function, are stored in the program store. The call programs provide the solution to any problem a customer can present to the system, either directly or through some other switching system. An assembly of call programs must tailor-make a connection according to the demands of the customer.

Several approaches toward providing programs for a large number of different offices could be used. A generic program, which is the same for each office, with detailed differences listed in a parameter table, is the approach used in No. 1 ESS. The generic program includes all features for a large number of offices, covering sizes from 2,000 to 65,000 lines and means for handling growth and changing traffic conditions. This approach simplifies record keeping, because only the parameter tables which specify present size and operating conditions are unique to each office. Additional data which characterize a particular office are found in translation tables also in the program store. Typically, 18 different sets of translations are required in each office. These include directory number to equipment number translations for both lines and trunks, class of service, and special treatment for lines and trunks.

In the future, economics may dictate the need for several generic programs—for instance, one for small offices, one for large offices, one for four-wire offices, and perhaps some combinations of these.

The development and preparation of programs for the system require the use of several utility programs written for a commercial computer, an IBM 7094 in this case. These programs are used to convert the language of the programmer to the language of the machine, to assemble and compile the individual pieces of call and maintenance programs, and to load information onto a tape which finally controls the writing of the magnets on the twistor cards. Additional programs are used to assemble, compile, and load parameters and translations. Utility pro-

grams will be used by the Western Electric Co. as tools for manufacturing the programs put into commercial installations.

IV. DEPENDABILITY

4.1 *Objectives*

The dependability requirement on a telephone office is limited only by what the state of the technology can provide. Certainly a new system must at least be comparable to existing offices. This means outages of no more than a few minutes in 40 years. Heretofore no large electronic machines have been able to approach this kind of dependability. In fact, the dependability objective represented one of the major challenges of the No. 1 ESS development. Since No. 1 ESS is a large digital information processor, it is a cousin to the general-purpose digital computer. However, the dependability requirement requires that No. 1 ESS be a very different kind of system with a much higher level of redundancy.

4.2 *The Large Immortal Machine*

The large size of No. 1 ESS and the unique dependability requirement lead to the term "large immortal machine." It also implies that No. 1 ESS is a new kind of information processor, a kind that has never been developed before.

Another way of contrasting No. 1 ESS with a computer is to compare the relative hazards of a machine data processing error and a total machine failure. In a general-purpose computation center a machine stoppage is a nuisance: the problem must be rerun. But a data processing error could be called a "disaster" because the results come out wrong. In a telephone office it is the other way around. A data processing error may cause a particular call to be mishandled. This is a nuisance, particularly to the customer whose call must be redialed. A total machine failure in a telephone office, however, means no telephone service during the outage, and the magnitude of such a disaster need not be argued. The key point is, of course, that the dependability requirement on No. 1 ESS is both remarkably severe and quite different from that of general-purpose computers.

4.3 *Duplication*

Since some failures of individual components are bound to occur over decades of system service, duplication is essential. Every system unit required to maintain service must be provided in duplicate. Not only

must there be duplication, but troubles must be found and corrected quickly to minimize exposure to system failure due to multiple troubles. This implies that all units must be continually monitored so that trouble in the standby can be found just as quickly as in the unit giving service. This further implies automatic detection of troubles and automatic switching of service when units fail. These processes are covered in detail in other papers.

4.4 *Repair*

When a trouble occurs, the telephone actions are interrupted as briefly as possible to reestablish an operational machine. Then, on a less urgent basis, the defective unit is diagnosed by the system itself and the results printed on the maintenance teletypewriter.

Where offices are unattended during at least part of the day, alarms and a remote teletypewriter are provided at some other location where 24-hour attendance is available. For minor troubles, not affecting service, the repair can be deferred. A defective trunk circuit can be taken out of service with a teletypewriter message. Major troubles may, on the other hand, need prompt attention to insure service continuity.

4.5 *Maintenance Programs*

More than half of the stored instructions are used for maintenance programs. Some of these programs, in conjunction with logic wired into the hardware, detect and report faults and troubles; others control routine tests and diagnose troubles and control emergency actions to insure a satisfactorily operating system, either by eliminating faulty subsystems or by reorganizing usable subsystems into a new operating combination. The classes of maintenance programs are arranged in a hierarchy of interrupt levels, so that when an error is detected the control system can be forced to stop what it is doing, keep a record of where it is, and after proper maintenance action is taken, restore itself to proceed with normal operation. In the hierarchy, a higher class can interrupt any lower class of maintenance or operating program.

V. TESTING

The separate development of major sections of hardware as individual subsystems, and of programs as other subsystems, and their subsequent successful combination into an operable system present major problems. First, the operation of all hardware subsystems working together to

form a single system must be verified. Each of the hardware subsystems is factory tested by a combination of manual and automatic tests. The over-all system operation is checked by a special set of programs known as X-ray programs, which check step-by-step the sequences of operation. Such programs will also be used by Western Electric installation people for checking new installations.

This procedure was also used on the Holmdel, New Jersey, No. 1 ESS as the first over-all check of the system design, and it was here that many interface problems were solved. The solution of these required changes in hardware, program, or both. After the hardware was checked and operating satisfactorily, testing of the call and maintenance programs was begun on the actual system. However, some simulation was usually done on the IBM 7094 computer before the program was submitted to the No. 1 ESS laboratory system. When all programs operate satisfactorily, the system will be tested under simulated traffic conditions and finally under actual traffic conditions.

VI. STATUS

At the time of writing of this article, all of the hardware has been tested by means of the test programs and proved to operate satisfactorily. Many of the call and maintenance programs are now operating on both the laboratory and Succasunna, New Jersey, systems. Other programs are still in preparation. Maintenance programs are usually the last prepared because they must be designed after the operating peculiarities of the hardware have been identified. Publication of untested plans, in general, is dangerous. However, some of the authors of the following articles, on programs as yet only partially tested, have decided to present their plans so that a relatively complete story of No. 1 ESS can be presented under a single cover. When the plans have been fully tested, additional details will be reported.

VII. SUMMARY

This paper has attempted to present a general outline of the organization and objectives of No. 1 ESS. It also serves as an introduction to the papers that follow.

VIII. ACKNOWLEDGMENTS

Because No. 1 ESS is such a large undertaking the efforts of many hundreds of people are involved. Even listing the numerous organiza-

tions within Bell Telephone Laboratories would be difficult and this would exclude the many people in the Western Electric Company, the American Telephone and Telegraph Company, and the operating companies who have contributed in important ways.

REFERENCE

1. Keister, W., Ketchledge, R. W., and Lovell, C. A., Proc. IEE, 107, Part B, Suppl. No. 20, 1960.

Organization of No. 1 ESS Central Processor

By J. A. HARR, F. F. TAYLOR and W. ULRICH

(Manuscript received January 28, 1964)

The central processor controls the operation of the No. 1 electronic switching system by executing sequences of program instructions. The logical organization of the central processor is described by the simultaneous evolution of:

(1) an instruction repertoire to carry out the required telephone and system maintenance tasks efficiently, and

(2) a circuit logic design to provide the necessary circuits (flip-flop registers, accumulators, etc.) to execute the instructions at a high data processing rate.

The design aims, order structure, timing, internal sequencing, and communications with the peripheral equipment of the system are described.

I. INTRODUCTION

Telephone central offices must cover a wide range of sizes and provide a large variety of services to customers; in addition, they must be compatible with existing systems, adaptable to varied and changing operating conditions, dependable, reliable, and economical. The development of an electronic switching system capable of satisfying these requirements presented many new problems to the designers. As a result, many techniques new to the telephone switching field were introduced in the system.[1] One of the most important new techniques is the control philosophy, which utilizes a stored program.

A system employing a stored program is one which consists of memories for storing both instructions and data, and a logic unit which monitors and controls peripheral equipment by performing a set of operations dictated by a sequence of program instructions. The stored program philosophy permitted the designers to use centralized logic circuitry and large-capacity memory units as a means of attaining flexibility and over-all economy in the system.

In this paper some of the design characteristics of the central processor are described, followed by a step-by-step development of the central control order structure and the corresponding logic circuitry needed for its implementation.

II. DESIGN CHARACTERISTICS

A simplified diagram of the electronic switching system (ESS) is shown in Fig. 1. The outer circle represents the entire No. 1 ESS, having primary inputs from lines and trunks[2] via scanners,[3] and outputs to the network[4] and signal distributor,[3] with teletypewriters as administrative input-output devices and with a magnetic tape for automatic message accounting (AMA)[5] output. The inner circle, the central processor, contains a central control[6] unit which executes program instructions and memory units used for storing program instructions and data.

In order for the central processor to handle traffic submitted by offices

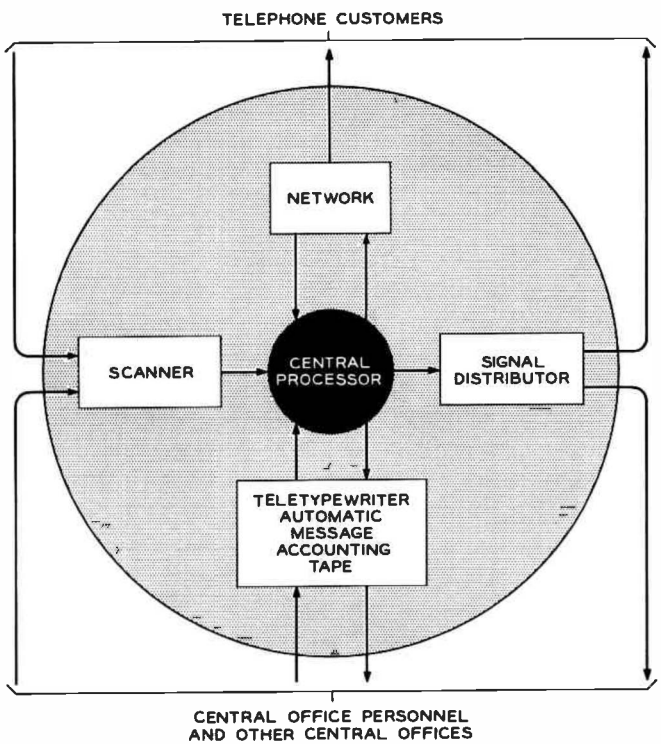


Fig. 1 — Simplified diagram of the No. 1 ESS.

serving 5000 to 60,000 customers, the number of memory units in the system must be expandable over a wide range. Therefore the central processor address registers, memory word size, and address buses are designed to accommodate the largest office. The system design must not only meet initial office size requirements, but must also include growth capability.

Since the data processing operations required for performing telephone functions are accomplished by executing a stored program[7,8] which must be error-free and remain error-free* at all times to insure the proper behavior of the system and good service for the customers, the memory chosen for storing the program is semipermanent and requires off-line operations to change it. (This avoids the risk of an error in operation introducing an error in the program.) Therefore the central processor contains two types of memory: a semipermanent memory system (program store)[9] for storing programs and a high-speed readable and writable memory (call store)[10] for storing call progress data.

An address of at least 21 binary bits is needed to gain access to the total number of words required in both the program and temporary memories. To meet this requirement, along with memory store design considerations, the designers decided upon a word length in the call store of 24 bits, comprising a parity check bit and 23 information bits. For convenience, the same word length is used throughout the central control. Most instructions which operate on memory words require a 21-bit address field. Instructions which contain a data field must allow for 23 bits in the data field to be compatible with the length of temporary memory words. Accordingly, the operation fields of each of these types of instructions are 16 and 14 bits in length, respectively, to accommodate the many types of instructions needed in each category. Therefore the instruction word length is 37 bits. To check and correct program instructions with single errors and to determine whether a word read has a double error in it, 7 check bits are also needed for each program instruction, making a total of 44 bits stored for each entry in the program store.

Engineering studies of the number and kind of telephone functions which the system must perform dictated a need for an efficient instruction repertoire with specific attributes. Some of these attributes will now be described.

To perform the functions required to handle the busy-hour traffic submitted by the largest office, the system can spend only approximately

* As will be seen later, facilities are available for detecting and correcting single errors.

5000 machine cycles per call. To meet this requirement, the instruction repertoire must include efficient multifunctional program instructions.

Since the repertoire must include the storage of data of variable bit length in the temporary memory and retrieval of these data, both read-from-memory and write-to-memory instructions include masking facilities. As used here, the masking of words read from memory means changing all of the bits of the word to zero except those which specify the item of interest; these remain in the same state as they were in memory. Masking of words written into memory is a facility for inserting an item of variable length into a word already in memory; the remaining bits of the word are unchanged.

To assist in achieving program word efficiency, in most cases the same functional program is used for all calls in progress requiring the execution of a given function.[7,8] For reliability, instructions in this system are not changeable at run-time (i.e., while the machine is actually processing calls). Indexing facilities provide means by which the programmer can change the addresses of memory words acted upon by the same program. Also, the indexing facilities can be used to vary the sequence of programs to be read and executed.

To carry out data processing of call information, the repertoire includes the arithmetic operations of addition, subtraction, comparison, shifting, and rotation, and the following logic operations: AND, OR, EXCLUSIVE-OR, and COMPLEMENT. Since the functions which the system must perform do not require multiplication and division operations, it was not necessary to include these in the central processor. To perform the logic functions required to carry out the many telephone functions, a variety of decision instructions are required to transfer control to specific program sequences based on the condition of internal central control registers after data manipulations have been performed on them. For example, a program of three instructions capable of performing the following three operations illustrates the primary way the central processor can vary the sequence of its operations according to input data it has received.

(1) Read, at an address specified by an index register, the word from memory containing the first dialed digit of a call; mask out all of the bits in the word other than the four bit positions used for the first digit; and load the word into an accumulator register.

(2) Compare the word just loaded into the accumulator with the value 10 (i.e., the number of pulses counted when a customer dials zero) specified by the data field of the instruction.

(3) Transfer to the program specified in the address field of the in-

struction if the two compared quantities are equal (this program will cause the customer to be connected to an operator); otherwise, continue the present program sequence.

For efficient operation, the central control should be capable of executing instructions which combine a number of the operations listed above. For example, the repertoire includes an instruction which reads the word at a temporary memory address specified by an index register, masks the word read, complements it, and then AND's the result with the accumulator register in the central control during one operational cycle.

To make efficient use of data processing time the repertoire must include a class of special instructions designed to facilitate the reading and making of logical decisions on input data and special orders for delivering output data to both the network controllers and trunk control circuitry. Therefore the central control can be described as an input-output processor superimposed on a general-purpose data processor.

Since the fundamental task of the central processor is continuous monitoring and controlling of its rapidly changing environment, consisting of wide variations of traffic submitted by customer lines and trunks, real time must be carefully considered when planning the system and writing the program. Programs to monitor and gather input data and to deliver output signals and data must be especially efficient in their use of central processor cycle time.

Input-output programs must be executed on a strict schedule. For example, the program to detect and receive dial pulses must be performed every 10 milliseconds. In order to accomplish this, an interrupt mechanism is included in the central processor. The interrupt mechanism, when activated by a source such as a clock or check circuitry, generates a new program address, thus transferring control of the system to an interrupt program sequence. When this occurs, the address of the next instruction which normally would have been executed is automatically stored. When the interrupt program completes its task, control is returned to the interrupted program.

To make the system capable of continuous operation during malfunction of circuit components, the central processor is duplicated. In order to detect and ultimately pinpoint hardware malfunctions, the central processors include:

(1) circuitry which compares the execution of instructions in both central processors,

(2) circuitry for generating a parity bit for each word stored in the temporary memory,

(3) circuitry for checking the parity of words read from temporary memory,

(4) circuitry for detecting and correcting single-bit errors in instructions read from the program memory,

(5) circuitry within the peripheral equipment for checking data received and for notifying the central processor when either data it receives or its own check circuitry indicates trouble,

(6) circuitry within the central control for verifying signals sent back by peripheral equipment, and

(7) circuitry for special-purpose instructions for controlling and interrogating stores and peripheral equipment.

III. BASIC DESCRIPTION OF CENTRAL CONTROL

In this section an order structure, including the symbolic names of the instructions, and a central control block diagram will be concurrently explained.

3.1 *Basic Facilities*

In the simplest form (see Fig. 2), the central processor consists of a central control, a program store which receives an address from central control and returns the corresponding instruction, and a call store, which receives an address and either receives data to be recorded at that location or returns the data previously stored at that location.

As a starting point, central control must contain registers for receiving program store instructions and call store data, and facilities for generating and transmitting addresses to both stores and data to the call store (see Fig. 3). The buffer order word register (BOWR) receives instructions from the program store, and the data buffer register (BR) receives data from the call store. The program store address register (PAR) is

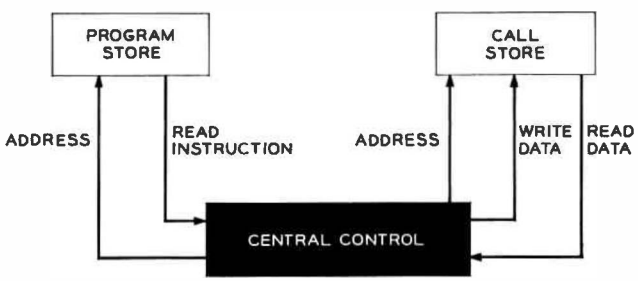


Fig. 2 — Block diagram of the data processor.

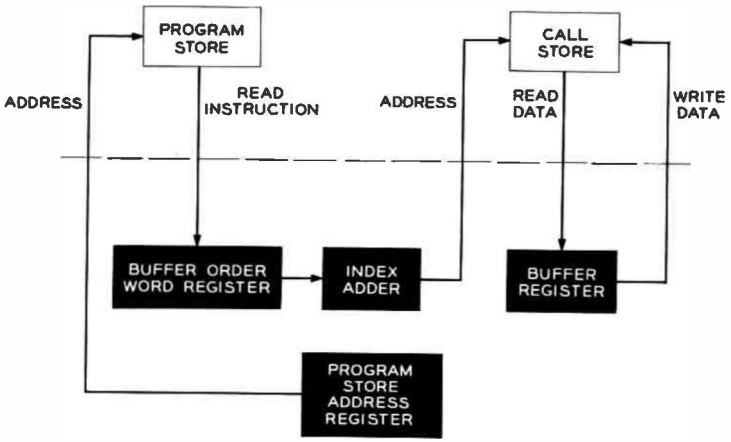


Fig. 3 — Derivation of the central control (1).

used as the source of addresses of instructions to the program store, and the index adder is used to generate addresses for the call store.

These facilities must be augmented by an instruction decoder, the buffer order word decoder (BOWD) attached to the buffer order word register (see Fig. 4). This decoder is used to control the gating of information inside central control.

Also needed to control such gating and to synchronize the decoding with the reading of information from the stores is a clock. In the No. 1 ESS central control, the clock is a synchronous 5.5-microsecond clock, with 22 distinct phases separated by 0.25 microsecond. Arbitrary-length clocking pulses are derived by setting a flip-flop circuit with one arbitrary phase, resetting it with another, and using the output of the flip-flop as the clocking pulse. Such pulses are repeated once every 5.5 microseconds.

Gates are therefore controlled by decoding the output of the buffer order word register and combining the decoded output with a suitable clock pulse.

3.2 *Index Registers*

In addition to the buffer register, there are a number of general-purpose flip-flop index registers, F, X, Y, and Z (see Fig. 4). The index registers are 23 bits long, the basic word length of the central control and the call store. (The 24th bit of the call store, a parity check bit, does not store useful information; this bit need not be carried along in central control data processing, although it must be generated anew when-

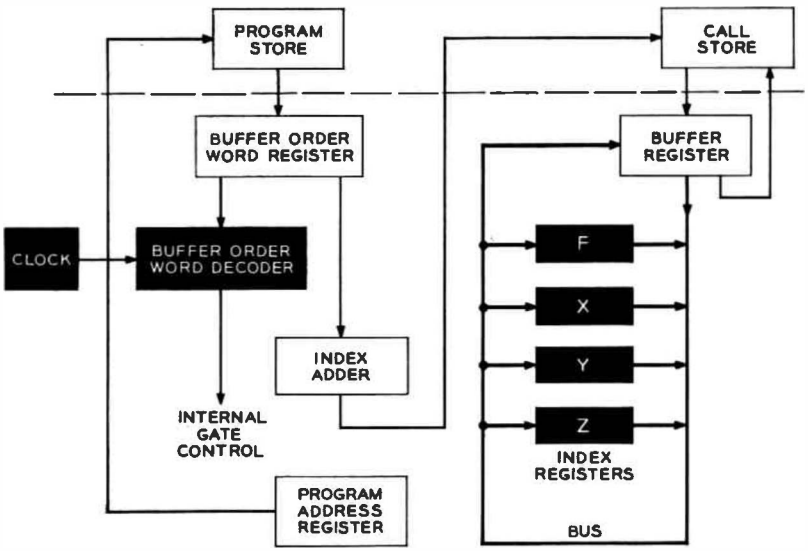


Fig. 4 — Derivation of the central control (2).

ever information is stored in the call store and checked whenever information is read from the call store.)

Indexing is useful for developing a program which is general for any telephone central office. It is the process of deriving a memory address by adding a constant from the instruction to a variable previously derived and stored in an index register. For example, the index register might contain the starting address of a block of call store words used for accumulating dialed information; the constant might be the location of a word within such a block, containing information known to be needed at a certain stage of a call. Such an instruction may be described as follows: fetch the call store reading in the third word of a block whose starting address is stored in the Y index register, and store the reading in the X index register. The description of the operation is divided into three parts (see Fig. 5): the basic operation (fetch data from memory

	OPERATION	CONSTANT	INDEX REGISTER	———
MEMORY TO X REGISTER	MX	3	Y	

Fig. 5 — Basic instruction format 1.

and store in the X register), the constant of the instruction (3), and the index register used in indexing (Y). In the basic mnemonics of the system, memory to X register is written as MX; therefore this instruction is written as MX,3,Y. Instructions which read or write in memory contain M in their mnemonic representation and are collectively designated M instructions.

3.3 *Bus*

Information is transmitted among the index registers via a bus (Fig. 4) consisting of 23 parallel information paths. The F, X, Y, and Z registers plus the buffer register, which can also be treated as an index register, all have output gates to and input gates from the bus.

3.4 *Index Adder*

In order to perform indexing, an adder (see Fig. 6) is required. The index adder receives one input (the constant of the instruction) from the buffer order word register, and the other input (the variable, i.e., the contents of the specified index register) from the bus. The output

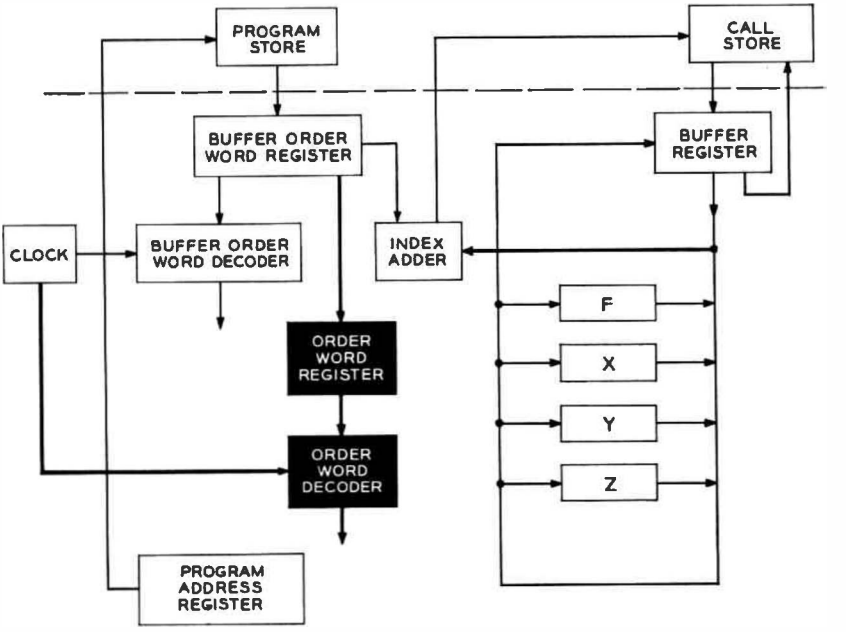


Fig. 6 — Derivation of the central control (3).

of the index adder, as previously indicated, is the source of addresses to the call store.

3.5 *Order Word Register*

The basic cycle time of the program store is 5.5 microseconds, and the maximum time from the reception of an instruction such as MX,3,Y until the specified reading from a call store is in the data buffer register is about 6.0 microseconds. When one considers the additional data processing of a call store reading after it has been received by central control, it can be seen that the execution time of an instruction occupies major fractions of two machine cycles. Thus there is overlap in the execution of two consecutive instructions. This overlap is described in Section VIII. An order word register and decoder are therefore provided to control part of the execution of an instruction. The buffer order word decoder and the order word decoder simultaneously control the execution of two consecutive instructions.

The buffer order word decoder controls the addressing of the call store. On a reading instruction, the order word decoder controls the gating of information from the call store to the data buffer register and thence, via the bus, to the destination register; on a writing instruction, the order word decoder controls the gating of data from some source register to the data buffer register and thence to the call store. The actions of the two decoders are sufficiently independent that the division of decoders into a buffer order word decoder and an order word decoder does not cost very much compared to the use of a single decoder.

3.6 *Masking: Logic Register, Unmasked Bus, Masked Bus, and Mask Circuit*

The 23-bit word length is much longer than many of the basic quantities of data. A long useful quantity of data is a 21-bit memory address. A typical short quantity is a single binary-coded decimal digit, 4 bits long; several such short quantities may be packed in a single word. In order to treat partial words efficiently, the central control has masking facilities (see Fig. 7). Since most data words pass over the bus, a single mask circuit on the bus accomplishes most of the masking functions. The mask circuit has two inputs, the unmasked bus, which is connected to the gates at the outputs of index registers, and the output of a logic register whose chief function is to control the masking function. The output of the mask circuit is called the "masked bus" and is connected to the input gates of index registers.

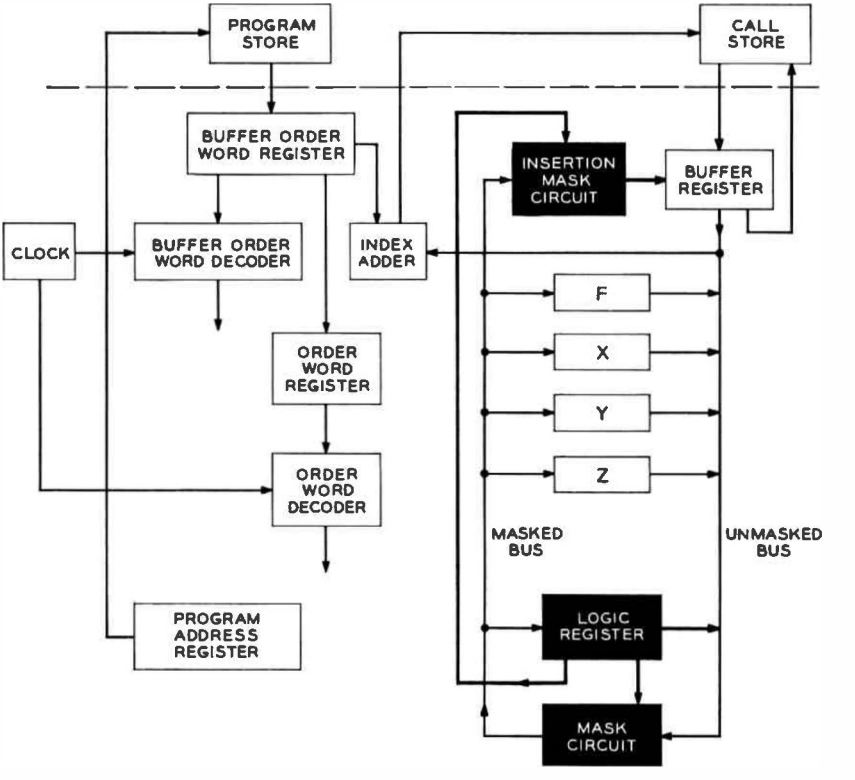


Fig. 7 — Derivation of the central control (4).

The logic register is 23 bits long; each bit controls the masking of 1 bit on the bus. The logic register is itself connected to the unmasked and masked bus so that it may be controlled and read as easily as any index register.

Masking is an option of most instructions. If in the previously described instruction (MX,3,Y) only the four least significant bits were of interest and the logic register were already set up with these four bits equal to one and the rest of the bits equal to zero, then by specifying masking only the four least significant bits would be transmitted to the X register (see Fig. 8). The rest of the bits would be transmitted as zero. This form of masking is called PL masking (P = product, L = state of the logic register; therefore, product with the state of the logic register). This instruction would then be written as MX,3,Y,PL.

	OPERATION	CONSTANT	INDEX REGISTER	MASK AND INSERTION
	MX	3	Y	PL (MASK)
X REGISTER TO MEMORY	XM	3	Y	EL (INSERTION)

PL OR EL WILL USE THE CURRENT CONTENTS OF THE LOGIC REGISTER L WHICH WAS SET BY A PREVIOUS INSTRUCTION

Fig. 8 — Basic instruction format 2.

3.7 *Insertion Masking*

Another form of masking that is used frequently is insertion masking. Insertion masking permits all but a selected group of the bits of a certain register to remain unchanged. The selected group of bits is then set up according to the instruction. Because of the requirement that certain bits remain intact, it is convenient to associate the insertion mask circuit with only one of the registers. The most logical choice is the buffer register, since insertion masking is most frequently used when only a portion of a word in the call store is to be altered. The insertion mask circuit is also controlled by the logic register, since in most cases the bits to be inserted and the position associated with these bits have been set up in the logic register for some previous masking (PL) operation. Insertion masking is indicated by specifying EL masking. (E = insertion, and L = the state of the logic register.) If, for example, the four least significant bits of the X register are to be stored in the address Y + 3* while leaving the other bits of that memory location intact, this action could be performed with the following two-step program (provided that the logic register is already set up to 1's in the four least significant bits and 0's elsewhere): MB,3,Y (read the contents of memory at the address Y + 3 into the data buffer register); XM,3,Y,EL (insert the contents of X into the BR for all bit positions of the logic register equal to one, leaving the rest of the bits of the BR intact; then write the buffer register into memory at address Y + 3). If PL, instead of EL, masking had been specified, the contents of memory would be all 0's except in the four least significant bits; by specifying EL the upper bits remain the same as they appear in the BR.

There is no circuitry available at the call store for performing the equivalent of insertion. Therefore, insertion into memory must always

* For simplicity, the following convention is used in this paper: the contents of a register, such as Y, plus a constant, such as 3, are represented by an unbracketed expression, such as Y + 3.

be a two-step operation: the first step to read the word at which information is to be inserted; the second step to insert this information and then write a complete word back. Insertion is entirely a central control function, not a store function.

3.8 *Transfer Facilities*

So far the details of addressing a program store have not been shown except for a program store address register (PAR) which transmits such an address. In the program, one of two things can happen. Normally, the program advances from one instruction to the next, so the contents of the program store address register are simply incremented by one. This is accomplished by attaching an increment circuit to the program store address register (see Fig. 9). (The program stores themselves do not have any incrementing facilities. A program store is always addressed with a complete address.) However, sometimes in a program a transfer is necessary. A transfer instruction is an instruction which causes the program to go to another set of instructions, not the immediately following instruction. The most convenient source of the address to which the program would transfer is the output of the index adder, since this is the place where the contents of the instruction are combinable with the contents of registers and thus indirectly with memory readings. A connection from the index adder to the program address register is therefore provided (see Fig. 9).

Direct transfers are transfers to an indexed address. Indirect transfers — i.e., transfers to an address stored in memory, the memory being read through the use of an indexed address — are also possible in No. 1 ESS. An indirect transfer is indicated by an M suffix in the index register field. An input (for simplicity, left out of the diagram) from the call store to the buffer order word register transmits the transfer address to the index adder, thence to the program store address register.

3.9 *Complement Option*

Another option existing in the system is the complement option. When numbers are considered numerical rather than logical data, a negative number is stored as the complement of the positive number whose absolute value is the same. The most significant bit, 22, is the sign bit of the entire quantity. This means that both a positive and a negative quantity 0 exist in the system, since the complement of all 0's (+0) is all 1's (−0). Such a system has the advantage of having very simple adder circuits, even though it does introduce occasional programming

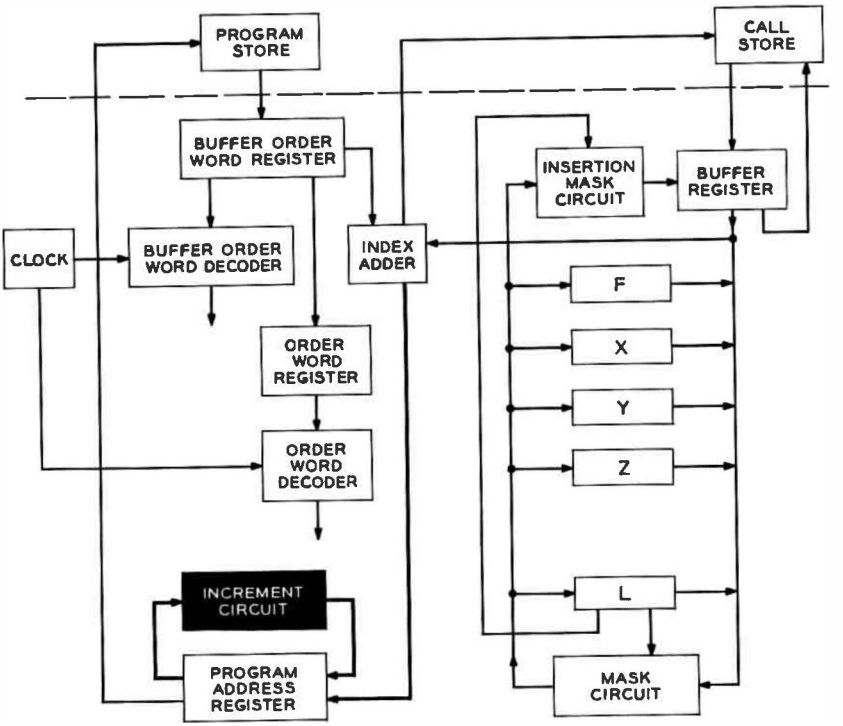


Fig. 9 — Derivation of the central control (5).

problems. The complement circuit is in series with the mask circuit (see Fig. 10); masking takes place before complementing. The complementing is specified as part of the mask field. Thus if we wish to load into X the masked and complemented contents of memory found at the address Y + 3 we may specify this instruction (see Fig. 11) by writing MX,3,Y,PLC.

3.10 *Data Instructions*

So far only instructions which deal with memory readings have been considered. There is another large class of instructions which deal with internal data manipulations and with the setting up or altering of registers by some constant (data word) within the instruction. These instructions are defined as W (for word) instructions. (It is important to bear in mind the fundamental property that instructions in this machine are not variable. A constant in an instruction is truly a constant until such a time as the program itself is altered, which can be done only by

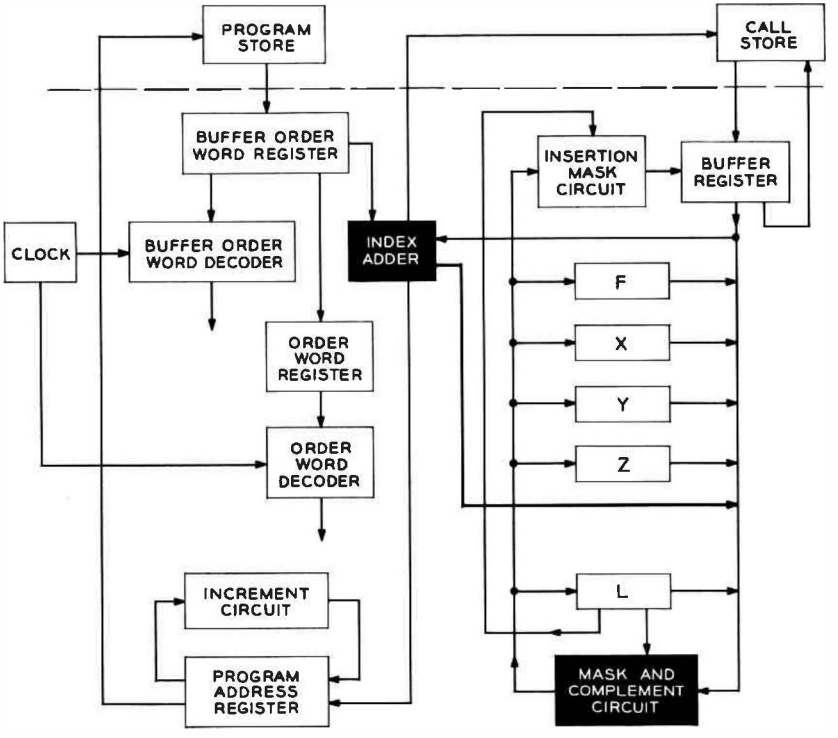


Fig. 10 — Derivation of the central control (6).

writing new permanent magnet twistor cards.) A natural channel for performing such data manipulations is via the index adder. Therefore the index adder has an output onto the unmasked bus (see Fig. 10). Thus, for example, we can increment register X by a constant by gating the X register to the index adder, adding the constant of the instruction and gating the output of the index adder via the bus back into the X register. All these operations are performed by the instruction WX (W equals indexed word, i.e., output of the index adder). This instruction is executed by circuit actions equivalent to generating a mem-

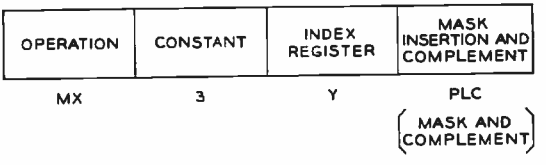


Fig. 11 — Basic instruction format 3.

ory address, except that the address is gated to the unmasked bus instead of the memory.

W instructions are also maskable, since the output of the index adder has to go through the mask circuit before it arrives at the destination register. Thus the instruction WX,3,Y,PL takes the Y register, increments it by 3, and places the result in the X register after first masking it according to the present contents of the logic register.

3.11 *Accumulator*

The central control must perform many additions and logical combinations of two quantities. It is convenient to use one register as an accumulator (K) and to permit this register to be combined readily with masked memory and W-type data. The accumulator adder (see Fig. 12)

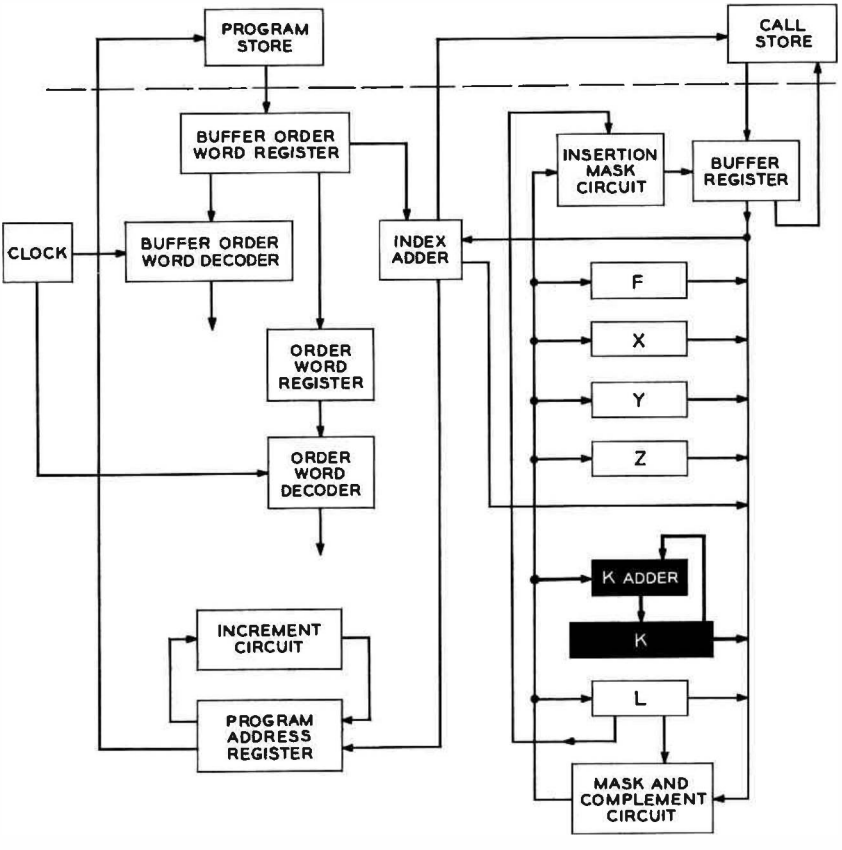


Fig. 12 — Derivation of the central control (7).

is capable of combining the present contents of the accumulator with indexed data or memory readings, both optionally masked, by adding, ANDing, ORing, or EXCLUSIVE-ORing the two. The timing problems are sufficiently severe that a single adder system cannot serve both as an index adder and an accumulator adder for combining two data operands. Accordingly, central control contains two adder systems to handle both operations concurrently.

Data in the accumulator can also be shifted and rotated. The shifting is not usually done for multiplication purposes, but to line up two items of information found in different positions within two data words to a position where the two items may be logically combined or treated in some other standard manner. For example, a pulse count for a decimal digit may always be accumulated in bit positions 0 through 3 (see Fig. 13). However, it may have to be stored in positions 4 through 7, or 8 through 11, or 12 through 15, according to which digit of a number it represents. To get data accumulated in positions 0 through 3 to positions 4 through 7, a shifting operation is necessary.

The rotation operation is similar to the shift operation except that for a left rotate the contents of the most significant bit, instead of being shifted out, are shifted back into the least significant bit, and vice versa for a right rotate. A special-purpose rotation within 16 positions of K is also available in central control. This rotation is extensively used in the network path hunt program.

Shifting is also performed very frequently when a number is composed of two parts, the first part indicating the location of an appropriate table of information and the second part indicating the location within that table.[11] For example, a line equipment number consists of a line link network and line switch frame indication, which is used to find a table, each table containing line translation information for one line switch frame and a position within that line switch frame which is used to find the line translation information within such a table. Without the ability to split such numbers into parts, it would be very diffi-

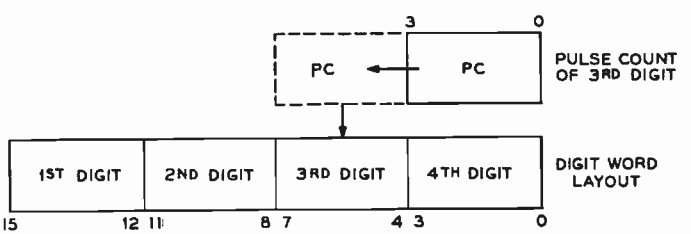


Fig. 13 — Use of shifting.

cult to organize the memory layout of the system for both rapid access and compact storage.

3.12 *Data from the Program Store*

So far the simplifying assumption has been made that data always come from a call store and instructions always come from a program store. In practice, however, much of the data is stored in the program store — specifically, the translation data. The process of reading a program store for data is a complicated one, especially in view of the overlap operation that is used.

The same types of instructions are used to read data from the program store and from the call store. This helps programming, since it does not fix a memory location at the time the program is written and helps to relieve the programmer from the burden of considering two different types of memories. A memory address decoder (see Fig. 14) connected to the output of the index adder recognizes when the output of this adder specifies an address that is not in the call store but is in program store.* It triggers a sequencer (see Fig. 14) which takes care of a special group of operations to be described below.

A sequencer is necessary to prevent the data that are coming from the program store from being incorrectly interpreted as an instruction. This sequencer must cause the program store to be read at the address specified by the output of the index adder and must then go on to the next instruction.

Fig. 15 shows the contents of the buffer order word register, order word register and program store address register during the processing of an instruction for reading data from the program store. The instruction is MX,BB,Y.† Y + BB specifies an address which happens to be in the program store. This instruction is located at the address AA. At time 1 the buffer order word register contains the instruction at address AA, the order word register contains the instruction at address AA − 1 and the program store address register contains the number AA + 1 in preparation for the reading of the next instruction. At time 2 the buffer order word register has the instruction at address AA + 1 but it cannot execute this instruction because the data word called for by the previous instruction has not yet been fetched. The order word register, in the

* Blocks of memory addresses assigned to program stores and call stores are fixed for all No. 1 ESS installations; the wiring pattern of the memory address decoder is therefore the same in all installations.

† BB is a symbolic representation of some constant; AA represents a program address.

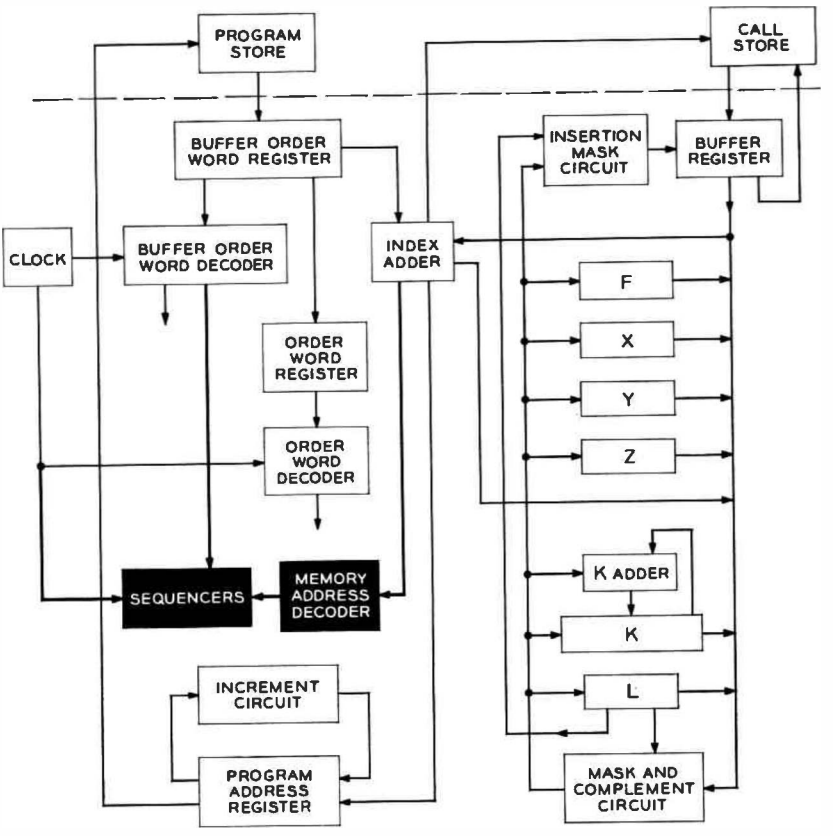


Fig. 14 — Derivation of the central control (8).

meantime, has the instruction at address AA, while the program store address register has received a data address from the index adder. This data address Y + BB is now being used to read the program store. At time 3 the buffer order word register contains the information located at address Y + BB, and the order word register continues to hold the instruction at address AA, while the program store address register has now been incremented to the value AA + 1 to prepare for the reading of the next instruction. At time slot 4 this instruction has been read into the buffer order word register, the output of the buffer order word register has gone via the index adder to the appropriate destination (the X register) under the control of the order word register, and the program address register is preparing to read the instruction at address

TIME SLOT	BUFFER ORDER WORD REGISTER	ORDER WORD REGISTER	PROGRAM ADDRESS REGISTER
1	(AA)	(AA − 1)	AA + 1
2*	[illegible]	(AA)	BB + Y
3*	(BB + Y)	(AA)	AA + 1
4	(AA + 1)	(AA)	AA + 2
5	(AA + 2)	(AA + 1)	AA + 3

(—)→ SYMBOL MEANING WORD STORED AT THIS ADDRESS OR IN THIS REGISTER.
INSTRUCTION AT ADDRESS AA IS MX, BB, Y;
BB + Y IS AN ADDRESS OF DATA IN THE PROGRAM STORE.

*THESE ACTIONS ARE CONTROLLED BY CENTRAL CONTROL INTERNAL SEQUENCE CIRCUITS SINCE THE OPERATION COVERS MORE THAN ONE CENTRAL CONTROL CYCLE TIME.

Fig. 15 — Time sequence of words passing through BOWR, OWR, and PAR when reading data from program store.

AA + 2. At time 5 this instruction is in the buffer order word register and the order word register has the instruction located at address AA + 1, while the program store address register has been incremented to AA + 3. Since the order word decoder is strictly a combinational circuit, the sequencer must be used to control the actions of fetching the data; otherwise the order word register would simply control the execution of the instruction at address AA three consecutive times. Note that the instruction for reading data from the program store consumes three cycles: one basic cycle, one cycle to read the data from the program store, and one cycle to reread the next instruction.

3.13 *Conditional Transfers*

A very important part of any data processing machine instruction repertoire is the set of conditional transfer instructions. These instructions cause a transfer of program control to a specified address if some data word or bit of data appearing in central control is some predetermined value. If the word or bit does not have that value, the transfer is not made, and the instruction immediately following the transfer instruction is executed.

Eight transfer instructions are provided to interrogate the contents of the accumulator for the following values: positive, negative, arithmetic zero,* all but arithmetic zero, logical zero (+0 only), all but logical zero, less than or equal to arithmetic zero, and greater than or equal to arithmetic zero.

* Arithmetic zero includes +0 (all zeros) and −0 (all ones). In both cases all 23 bits are alike, and arithmetic zero is therefore also referred to as "homogeneity."

A pair of control (C) flip-flops connected to the masked bus (see Fig. 16) store the homogeneity and sign of data words read from memory as the words appear on the masked bus. Another eight transfer orders test the C flip-flops for the same combinations of values available for testing the accumulator register.

Normally, the conditional transfer instructions follow immediately after the condition is registered, either in the accumulator or the C flip-flops. The usual instructions for gating information into registers may set the accumulator or the C flip-flops. In addition, there is a set of compare instructions (see Fig. 17) which do not alter any register but

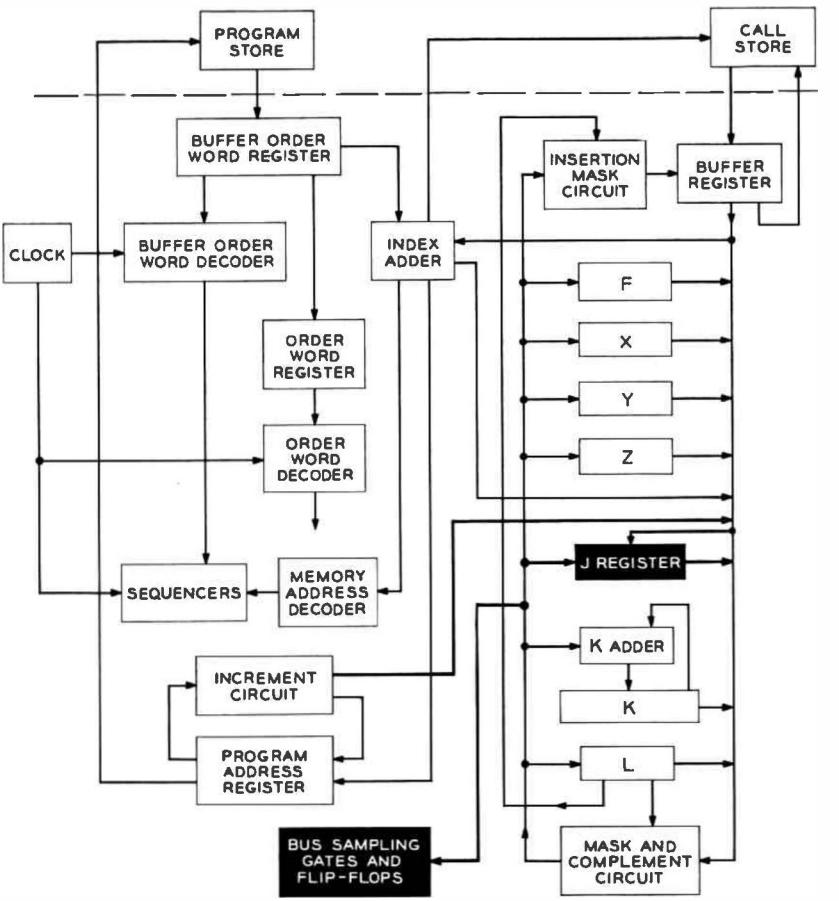


Fig. 16 — Derivation of central control (9).

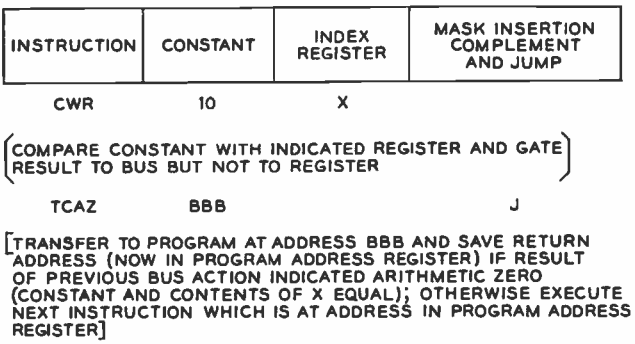


Fig. 17 — Basic instruction format 4.

which set the C flip-flops according to the result of the comparison. For example, the instruction CWR,10,X compares 10 with X and places the result in the C flip-flops. The comparison is performed by subtracting 10 from X, placing the result on the bus, and gating it only to the C flip-flops. One way to check whether the X register was actually equal to 10 is to follow the first instruction with an instruction TCAZ, BBB: transfer to the address BBB if the C flip-flops are equal to arithmetic 0; if not, advance to the next instruction.

3.14 *J Option and Register*

Associated with transfer instructions is a return address (jump) option (see Fig. 17). Unconditional or conditional transfer instructions occur frequently in the middle of a program and are used to transfer to a subroutine to do a common task; subsequently the subroutine returns control to the original program. Since the subroutine must know where to return, a J register (see Fig. 16) has been provided which may be set up at the discretion of the programmer whenever a transfer is executed. If the transfer is actually executed, the J register is set to the address of the instruction immediately following the transfer instruction. To set up the J register, a path must be provided from the output of the increment circuit to the unmasked bus and thence to the J register (see Fig. 16).

3.15 *Index Register Modification Options*

Index register modification options are available in the No. 1 ESS order structure. If some task is being performed on a number of successive memory locations, it is sometimes convenient to set an index

register to the value of the first memory address, then to increment the register by +1 as successive words are read from memory. This incrementing can be performed as an option on most reading instructions. For example, the instruction MX,3,Y simply gates the contents of memory at the address Y + 3 into the X register. The instruction MX,3,YA (see Fig. 18) gates the contents of memory at the address Y + 3 into the X register and increments Y by 1. The incrementing is performed by connecting the increment circuit input to the unmasked bus (see Fig. 19); this allows the index register to be gated into the increment circuit; the output of the increment circuit may later be gated to the index register via an output connection to the masked bus.

Two other index register modification options exist which change the indicated index register to the indexed quantity. Thus, the instruction MX,3,YW reads the memory at address Y + 3 and gates this into the X register and also changes Y to the value W, the indexed quantity, which is Y + 3. Another index register modification option is the setup index register modification. For example, MX,30000,SY would read the contents of memory at address 30,000 into the X register and would place the quantity 30,000 into the Y register (SY = set up Y).

3.16 *Logic Register Setup Options*

The logic register is changed very frequently in the course of a typical program. Furthermore, many of the programs that are encountered include effectively indirectly addressed readings,* i.e., one instruction is used to read a quantity, and the reading is then used as the address of the quantity which is actually desired. In such a case, the register being used for the second reading usually contains the complete address so that no additional data are required as the constant of the instruction. Consequently, it is desirable to be able to use the constant of the second instruction to set up the logic register instead of performing unnecessary indexing. A direct path has therefore been provided between the buffer order word register and the logic register (see Fig. 19). If setup masking is specified, then the constant of the instruction is used to set up the logic register and is not used in indexing.

The interplay of index register modification and setup masking options can be illustrated by the following programming problem: complement the 3 least significant bits of memory at the address Y + 3.

* Indirect addressing was not considered worthwhile for data operations, because it never saves time and rarely saves instructions in this system.

INSTRUCTION	CONSTANT	INDEX REGISTER AND MODIFICATION	MASK INSERTION COMPLEMENT AND JUMP
MX	3	YA	

[READ CONTENTS OF MEMORY AT ADDRESS Y+3 INTO X REGISTER; THEN INCREMENT Y BY +1]

Fig. 18 — Basic instruction format 5.

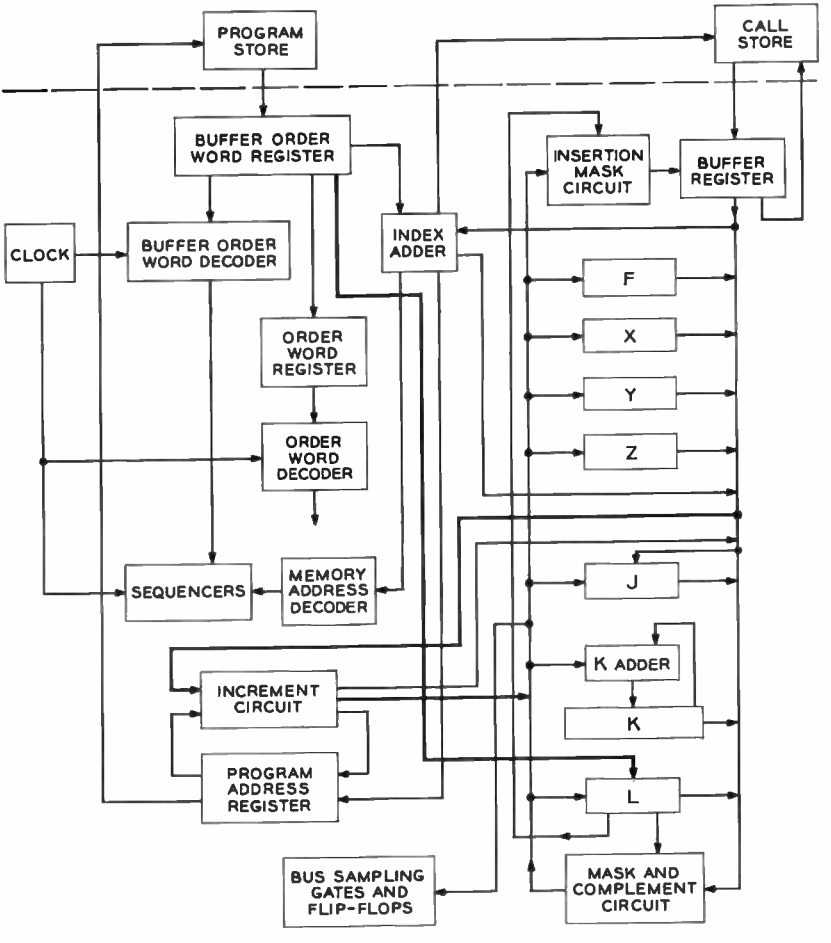


Fig. 19 — Derivation of the central control (10).

Using straightforward programming, this can be done by the following three-step program: MX,3,Y (read memory into the X register); WL,7 (set up the logic register to the value seven, i.e., 1's in the 3 least significant bits, 0's elsewhere); XM,3,Y,ELC (store X in memory using insertion masking and complementing at the address Y + 3). This program will result in 3 least significant bits of X — i.e., the bits selected by the logic register — being complemented (C option) and inserted (EL option) into, first the buffer register, thence the memory, at the location Y + 3. A two-step program for performing the same task is the following: MX,3,YW (read from memory at address Y + 3 into the X register and set up Y to the value Y + 3); XM,7,Y,ESC (set up the logic register to the quantity 7 as indicated by the S in the ESC mask option; complement, as indicated by C, the X register, and insert, as indicated by E, into the buffer register the 3 least significant bits as selected by the L register; and gate this information to memory at the address which is now in Y, and which is 3 greater than the original value of Y). Note that the first program repeated the constant 3 twice, whereas the second program used it only once. The index register modification option permitted the constant of the instruction to be remembered for subsequent instructions without using any extra steps. Similarly, the constant 7 for setting up the mask was useful in the second instruction of the modified program because, since no constant was necessary for addressing the memory, a constant could be used for setting up the logic register.

It is important to remember that these options not only conserve memory space for instructions but save the time necessary to execute additional instructions. In this system, each instruction takes one cycle whether it be a memory instruction (M) or a register setup instruction (W).

3.17 *Rightmost One Function*

One function that occurs frequently in the type of data processing work that constitutes the call processing program of the No. 1 ESS is that of detecting and identifying a one in the midst of a group of zeros in a word. The one might signify a request for action, the zeros, inactivity; the position of the one would represent which member of the group requires the action. By concentrating always on the least significant one, successively all requests will eventually be handled.

It is important to have some instruction which identifies the position within a word of the rightmost one, because this operation is performed often and is awkward to do using more conventional instructions. The

word to be examined is placed in the accumulator; a rightmost one detect circuit connected to the accumulator gates the binary position of the rightmost one onto the unmasked bus. This information is then transmitted via the buses to the F (for "first-one") register. A circuit to reset this bit in the accumulator then receives its selection information from the F register.

Two instructions exist for the first-one function: TZRFZ transfers if the accumulator is zero; otherwise, it gates the position of the rightmost one to the F register and resets that bit; TZRFU performs the same actions, except that the bit in K is not reset. The programmer specifies the transfer address information in the same way that it is specified for any conditional transfer instruction.

3.18 *Buffer Bus Registers*

In addition to the index registers described above, a number of other flip-flops in the central control are under the control of a programmer. They include the bulk of the maintenance control and match flip-flops. These flip-flops are in groups and are set up and read by buses connected to the B register (see Fig. 20). The flip-flop groups are then examined and controlled as if they were word locations in memory. Each flip-flop group is assigned a distinct address; when this address is generated, the memory address decoder operates the gates to or from this flip-flop group. Thus the control of registers does not consume instruction code space.

3.19 *Interrupt Facilities*

The No. 1 ESS central control has interrupt facilities. These facilities permit a signal to come in at any time and:

(1) cause the program currently being executed to be interrupted,

(2) permit the state of central control to be stored in memory,

(3) allow an interrupt program to be executed,

(4) allow the state of the central control to be restored to its pre-interrupt state, and

(5) allow the interrupted program to be resumed.

In effect, the programmer need not be concerned about the possibility of an interrupt occurring at any time, since the interrupt will not interfere with the execution of his program.

The interrupt is used for two purposes: first, dial pulse scanning and similar functions which must be performed every five or ten milliseconds are carried out by an interrupt program, triggered by a five-millisecond

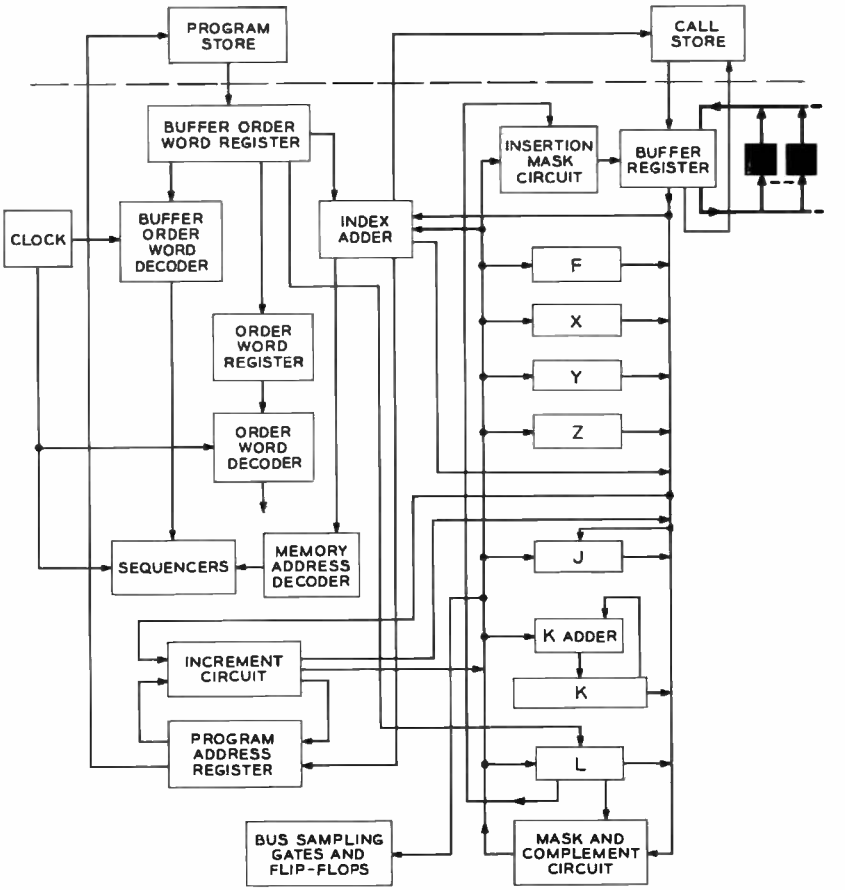


Fig. 20 — Derivation of the central control (11).

clock within the central control. Second, any trouble indication leads to an immediate interrupt to analyze the trouble and take corrective action before the trouble source causes errors in calls.

Call processing programs are carried out at the base level, which may be interrupted by any of the interrupt sources. Several levels of interrupt exist. A signal from a higher level has the ability to interrupt a program initiated by a lower-level interrupt. Most of the special interrupt facilities are obtained through the use of the interrupt sequencer.

In general, there is no problem of interaction among different interrupt programs and between the base-level program and interrupt programs, *provided they do not both write in the same sections of memory.*

Sometimes, this is unavoidable; for example, the program that detects an incoming trunk service request (an interrupt-level program) and the program that seizes an outgoing trunk on a terminating call (a base-level program) both write into the same busy-idle bits if the trunk is a two-way trunk used for both incoming and outgoing calls. Interaction problems may occur if a bit is being inserted in memory and the interrupt occurs between the reading and writing steps that constitute an insertion into memory. Two instructions have been created to solve this problem: MCII and MKII. These instructions are the normal MC (memory to the buffer and also the C flip-flops) and MK, except that all but the maintenance interrupts are barred until the immediately following instruction has been executed. By using one of these instructions as the first step of an insertion, a programmer guarantees that no interfering interrupt will occur while he is inserting the desired information.

3.20 *Mixed Indexing*

Sometimes, a couplet of instructions such as

MX,0,Y

MZ,0,X

occurs. The second of these instructions uses the value of X that was set up by the first instruction. Since the call store reading of the first instruction comes back at the same time that indexing is performed for the second instruction, a timing problem exists. This is handled by recognizing such situations (mixed indexing) and gating the data on the masked bus, i.e., the data going into the X register, to the index adder (see Fig. 20), instead of gating data from the unmasked bus, i.e., the present contents of the X register. The circuit for recognizing this situation must examine the buffer order word register and the order word register to check for this condition. Note that if an interrupt takes place between these two instructions, then the X register will have been set up to the new value, and no mixed indexing takes place; in effect, the second member of the couplet has been preceded by a vacant or no-op instruction.

3.21 *Early Transfer Instructions*

For certain highly repetitive programs, especially those involving scanning, it is desirable to have a conditional transfer instruction which

will consume additional cycles only if the transfer is made. This is accomplished by coupling a normal read or write instruction with an indication that a transfer to a preset address is to be made if the C flip-flops or the K register records a particular state. The instruction is called an "early transfer" instruction because, if the transfer is made, the reading or writing action is inhibited; the transfer decision must be made sufficiently early so that the unwanted action is prevented.

Two of the early transfer instructions are TCMMF and TAULM. TCMMF will transfer to the address (previously set up) stored in the J register if the C sign flip-flop shows a 1 (or minus); otherwise, a normal MF (memory to the F register) instruction is executed. All options normally available for an MF instruction can be specified, since the TCMMF operation itself completely specifies the conditional transfer instruction and does so without permitting any options or any choice of the source of the transfer address. TAULM will transfer to the address (previously set up) in the Z register if the C flip-flops show a nonzero quantity; otherwise, a normal LM (logic register to memory) instruction is executed. Again, all options normally available for an LM instruction can be specified; the TAULM operation completely specifies the conditional transfer instruction and transfer address source.

The advantage of the early transfer instructions is that the transfer address need be set up only once for a large number of loops of a subroutine, or that the transfer address may have been previously set up in the course of executing another part of the program; if no transfer takes place, no cycles have been wasted on making the decision.

3.22 *Logical Combinations of Registers with Memory or Data*

Instructions are available which permit indexed data (W) or the contents of memory found at an indexed location (M) to be logically combined with the contents of the X, Y, or Z register. This is accomplished by first gating the X, Y, or Z register to the L register via the buses, then using the mask and complement circuit to logically combine L with either M or W and gate the result back into X, Y, or Z. AND, OR, AND complemented, and OR complemented are the logical expressions that may be specified; naturally, this means that the mask circuit must be able to OR as well as AND. Since these instructions use the L register, no masking may be specified.

IV. PERIPHERAL SYSTEM FACILITIES IN CENTRAL CONTROL

In addition to communicating with the stores, the central control also communicates with the peripheral system.[12] This system contains three

main bus systems (Fig. 21). The first of these is used in communicating with the central pulse distributor.[3] This central pulse distributor either operates flip-flops which drive relays directly or is used for selecting the particular unit which is to be addressed via the second bus system, the peripheral address bus. Responses from peripheral units come via the scanner answer bus.

Peripheral actions are not generally performed in the middle of a complicated data processing problem. Therefore, some of the general purpose index registers of central control are used for driving these buses (Fig. 22). The F register is used to drive the translator which controls the central pulse distributor. The logic register is used to receive answers from the scanner response bus. This makes it easy to combine scanner answers with memory information, since the memory information can be read into the buffer register and can be combined logically with the contents of the logic register in the mask and complement circuit.

Ideally, it would be most reasonable to drive the peripheral address bus from the data buffer register. However, the timing of the waves of information leaving central control to address the peripheral system is such that the buffer register would not be available for a reading on the subsequent cycle; peripheral actions start after a considerable amount of preliminary processing, frequently including a call store reading. For this reason the contents of the buffer register are transmitted to the accumulator addend register, which is known to be available at this time. The accumulator addend register is used to drive the translators connected to the peripheral addressing bus.

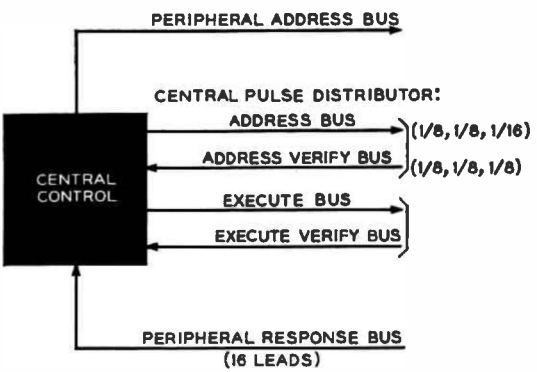


Fig. 21 — Block diagram of central control communication with peripheral equipment.

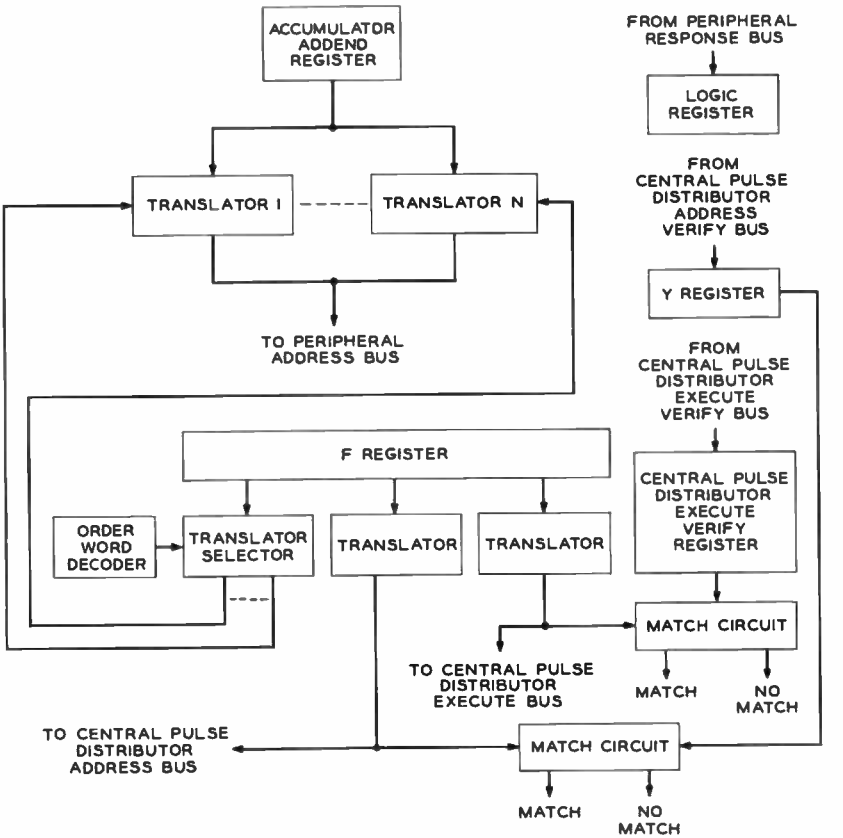


Fig. 22 — Detached internal central control blocks for communication with peripheral equipment.

A large majority of the units driven by the peripheral system are network units,[4] signal distributors,[3] and scanners.[3] For economic design of the many peripheral controllers, a coded address is used for controlling these units. For example, a 1024-point scanner requiring a 1-out-of-64 row selection is addressed by two 1-out-of-8 signals, not by 6 binary signals. Furthermore, the address code for each of the various network frames and the signal distributors is different. Central control has built within it translators to convert the binary information, convenient for data processing, to the particular code used for controlling a specific unit.

Since many peripheral instructions are sent out from a section of call store called the "peripheral order buffer," it is desirable that the actual

peripheral instructions be independent of which particular type of frame is being addressed. Since the enable address for a particular frame is always required along with the peripheral instruction, it is convenient to store the type of frame along with this enable address so that the proper translation of the binary addressing information will be made. A peripheral action is therefore preceded by setting up the F register to the enable address necessary for selecting the particular unit and a code which will select the proper translation option of the binary information to be sent to the peripheral bus. The actual peripheral instruction is then set out via an instruction, such as MA, which assumes that the F register has been preset. MA first sends out an enable signal, then takes the contents of memory and gates them via the buffer register and accumulator addend register to the peripheral address bus, translating the binary information according to the code stored in the F register.

When a peripheral (scanner) response is to be returned to central control, it is sent to the logic register. Therefore the instructions for controlling scanners are different from the instructions for simply sending an instruction to a peripheral unit which does not give a response. The basic general-purpose peripheral instructions are therefore MA and MAS (MAS being used when a response is expected) and the counterparts, WA and WAS. The MAS instruction and WAS instruction also reset the logic register and open gates to this register from the peripheral response bus.

If a central pulse distributor is addressed only for the purpose of operating or resetting a flip-flop, a special instruction, MD or WD, is used. With these instructions it is only necessary to set up the F register, since the peripheral addressing bus is not required.

For addressing the scanner, only 6 bits of information are necessary to select a row. There is enough information in a 23-bit word to set up an enable address and to actually address the scanner. The MSF and WSF instructions, by simultaneously setting up the F register and the accumulator addend and then performing a peripheral operation, accomplish this in one setup cycle. (The actual execution of the instruction overlaps into the next cycle so that two scanner readings cannot be made in consecutive cycles.)

For purposes of verifying peripheral operations, a verify response signal comes back from the central pulse distributor. This response is gated into the Y register and is then matched against the output of the translator that is connected to the F register. (No response is returned when executing MD or WD instructions.)

For driving such units as the tape unit,[5] which will accept straight binary information, and for sending test signals to peripheral units, it is also possible to bypass the translator, simply letting the outputs of the accumulator addend register go directly to the addressing bus. Since the peripheral addressing bus is 36 bits wide, it is possible, if testing a unit which requires more than 23 input leads, to take 13 bits from the accumulator as well as 23 bits from the accumulator addend.

It is important to remember that in the case of network frames the peripheral addressing bus contains instructions as well as addressing data.[4]

Another check made on peripheral operations is that the proper central pulse distributor has been selected. When a central pulse distributor is selected, it sends back an echo signal which goes to a flip-flop register on the buffer bus. This is matched against the output of a translator attached to the F register.

V. SUMMARY OF ESS ORDER STRUCTURE WITH OPTIONS

This section is a summary of No. 1 ESS instructions with their available options, as shown in Table I. As has been previously indicated, every instruction has three main modifiers: the data field; the RM field, which includes index register modification and indirect addressing for transfers; and the LCJ field, which includes masking options, complementing, and the setting up of the J register on transfers. In addition, although this is not specified in the actual writing of each instruction, many instructions set up the C flip-flops, which can then be examined on a subsequent TC conditional transfer instruction.

The constant in the data field may serve one of three purposes: it may be directly used data that is part of the instruction; it may be part of the address used for finding such data; or it may be the mask that is to be used in the instruction. In Table I, the symbol S in the data or address section indicates that data or an address may be specified unless an S occurs in the LCJ field, in which case no data or address constant may be specified, since the constant of the instruction must be used for setting up the mask.

The R subfield usually contains the identity of the indexing register; in a few instructions, the R subfield specifies a register indicated by the instruction. The latter include the instructions for adding the contents of two registers (since only one of the registers can be specified as part of the instruction, the other register must be specified as an option); the CWR instruction, which is used to compare a register with a constant in the order; and the TR family of instructions which sense the

TABLE I — SUMMARY OF BASIC ORDERS AND AVAILABLE OPTIONS

Symbolic Order Fields	DA		RM						LCJ						
Subfields	D	A	R*					M	Lt				CJ		
General-purpose operation codes	Data	Address	Identity of R in OP Code	Indexing Register	A (R + 1 → R)	W (DA + R → R)	S (DA → R)	Indirect Addressing	PL Masking	EL Masking	PS Masking	ES Masking	C Complementing	J (RA → J)	Set C Flip-Flops
WK, AWK, SWK, PWK, UWK, XWK	S			√	√				√		√		√		
WF, WJ, WX, WY, WZ, CWK, CWKU	S			√	√				√		√		√		√
WL, PWX, PWY, PWZ, UWX, UWY, UWZ	√			√	√								√		√
WB	S			√	√				√	√	√	√	√		√
H, HC, Q, QC	√			√	√										
MC, MCII		S		√	√	S	S		√		√		√		√
CWR	S		√						√		√				√
MB		S		√	√	S	S				√				
BM		S		√	√	S	S					√			
ABR, AFR, AJR, AKR, ALR, AXR, AYR, AZR	L		√						√		√		√		√
SBR, SFR, SJR, SKR, SLR, SXR, SYR, SZR	L		√						√		√		√		√
FM, JM, KM, XM, YM, ZM,		S		√	√	S	S		√	√	√	√	√		
MK, AMK, SMK, PMK, UMK, XMK, MKII		S		√	√	S	S		√		√		√		
MF, MJ, MX, MY, MZ, CMK		S		√	√	S	S		√		√		√		√
LM		S		√	√	S	S			√	√	√	√		
ML, PMX, PMY, PMZ, UMX, UMY, UMZ		√		√	√	√	√						√		√

TABLE I — *Continued*

Symbolic Order Fields	DA		RM						LCJ						
Subfields	D	A	R*					M	L†				CJ		
General-purpose operation codes	Data	Address	Identity of R in OP Code	Indexing Register	A (R + 1 → R)	W (DA + R → R)	S (DA → R)	Indirect Addressing	PL Masking	EL Masking	PS Marking	ES Masking	C Complementing	J (RA → J)	Set C Flip-Flops
T, TK... TC...		√		√	C	C	C	√						√	
TR...		√	√		C		C	√	√				√		
Input-output operation codes															
MA, MAS, MSF, MD		S		√	√	S	S		√		√		√		
WA, WAS, WSF, WD	S			√	√				√		√		√		
Combined operation codes															
TZRFU, TZRFZ		√		√	√	√		√						√	
TAULM, TCMMF		+ S		+ √	+ √	+ S	+ S		+ √		+ √		+ √		

* Of options A, S, and W, only one may occur in any one instruction.
† Not more than one of the four L options can be used in an instruction.
Key to symbols:
√ — indicated use of field is available
S — √ unless S appears in L subfield
C — conditional and late: occurs only if transfer occurs and after register is used (or, in the case of TR... orders, after the register is tested)
L — √ only if S appears in L subfield
\+ — action of options occurs only if transfer does not occur.

contents of some register and transfer accordingly. Three index register modifications are available, of which only one, the A option, is available if setup masking is used. (This is a direct result of the fact that only one constant may be specified in any instruction; if this constant is used to set up the mask, then it cannot be used to modify an index register.)

Some general observations may be made concerning the types of options available with various instructions. W instructions do not have W or S index register modification options. While these index modification options are meaningful, they would have a relatively low utility and require a great deal of code space. The W and S options are much more useful on memory instructions for setting up a register to a full

address, so that on a subsequent instruction, the constant in the data field will be available for masking.

Certain instructions, such as PWX, UMY, WL, ML, etc., do not permit setup masking, since they either set up the mask directly as part of the instruction (WL), or the logic register is used in carrying out the instructions (UMX, etc.).

On MB instructions, no masking or complementing is possible, since the contents of the buffer register are fed directly by the memory and do not pass over the bus. The PS option is included and allows a programmer to set up the logic register for a subsequent instruction even though he does not use it in this instruction.

Insertion masking is used only on WB instructions and on the instructions which write the contents of some register to memory, since the insertion mask can only be associated with the B register.

The C flip-flops in general are set up on all compares and on all orders which gate W or M to some register other than K.

As can be seen, the rules for checking on which options are allowed on any particular instruction are not too simple. Therefore, even though they are summarized in Table I, the No. 1 ESS program compiler[13] checks for violations of the allowed options. There are, of course, a number of other restrictions that the compiler can check for. The chief restriction within No. 1 ESS is the fact that the accumulator may not be used as the indexing register on the instruction immediately following AMK, PMK, UMK, XMK, or SMK instructions. Following all peripheral instructions, there are a number of very complicated restrictions; in general, the Y and F registers cannot be altered on the next instruction, and the L register cannot be changed following MAS, WAS, MSF, and WSF.

VI. ENCODING OF THE ORDER STRUCTURE

Each instruction or program order word obtained by central control contains 37 information bits designated 36, 35, $\cdots$, 0. Each such order consists of an order field and a data-address field. The order field includes an order selection subfield, an index register subfield, and order option subfields. When the order includes a data word, the data-address field contains the data word in bits 22 through 0, and bits 36 through 23 compose the order field. When the order word contains an address in the data-address field, the address occupies only bits 20 through 0 of the word, and a larger order field appears in bits 36 through 21 of the program order word.

The 21-bit address field permits full access by memory reading and

writing orders to all locations in the program stores and call stores, as well as to a group of flip-flop registers in central control. When an order option subfield indicates the setting up of the logic register (e.g., PS or ES masking), the data-address field contains the 23-bit word of data; in such instances, the order field is restricted to bits 36 through 23, even when the order selection subfield indicates a memory reading or writing order.

The encoding of the instruction repertoire or order structure represents a compromise between: (1) attempting to provide an order structure with the maximum flexibility that can be represented by the available binary combinations and (2) decomposing the combinations of order selection, index register selection, and option selection into simple subfields. Counting all meaningful combinations of order selection, index register selection, and nonconflicting order option values, the encoding provides over 12,000 distinct combinations in the 14- and 16-bit order word.

The index register subfield is always encoded in bit positions 34 through 32. Bit position 35 is reserved for the complement option for all orders except regular transfer orders; bit position 35 serves as a J option subfield for transfer orders. A complete decomposition is not possible for index register modification options or masking options without an excessive waste of code space. However, the encoding includes a grouping of the classes of related orders, this grouping being represented by relatively simple bit combinations; within each grouping the index register modification and masking options are each grouped into one-, two-, or three-bit subfields according to the number of meaningful and useful combinations.

The encoding is shown as the Karnaugh maps in Figs. 23 through 26. These maps represent the four binary combinations of bits 31 and 30; this division of the encoding is representative of the grouping of several large classes of orders. For example, the binary combination 31-30 = 01 is assigned exclusively to orders reading memory; all such orders are encoded within this combination and its corresponding map in Fig. 23. The combination 31-30 = 11 is assigned to W orders in which the destination register is given explicitly by the mnemonic code and not by the index register field ($R_D \neq R_I$); the encoding of this class is shown in Fig. 24. Related orders, such as MA and WA, occupy corresponding positions within the two maps. This correspondence provides for economy in designing the gating functions which carry out the same steps for related orders. For example, a single destination register selector can be activated by either a memory reading order or its corresponding W order.

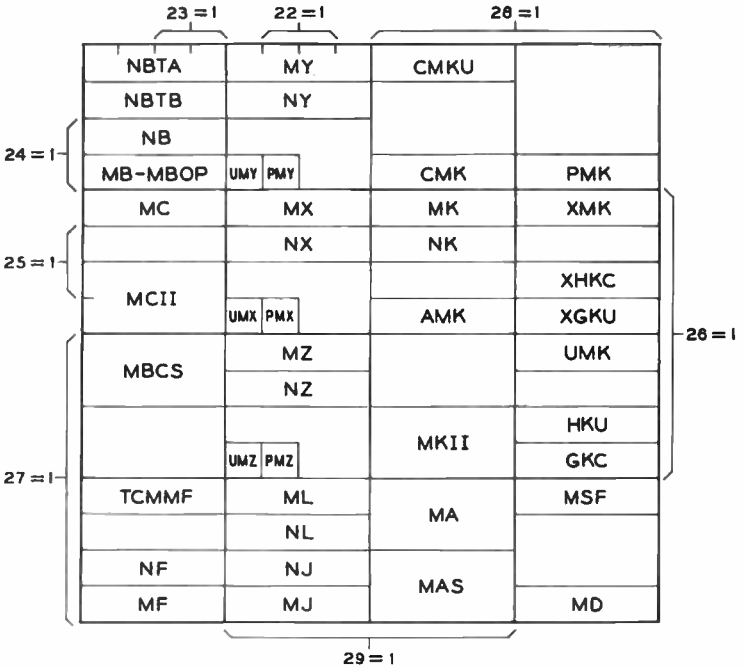


Fig. 23 — Memory reading orders (31-30 = 01).

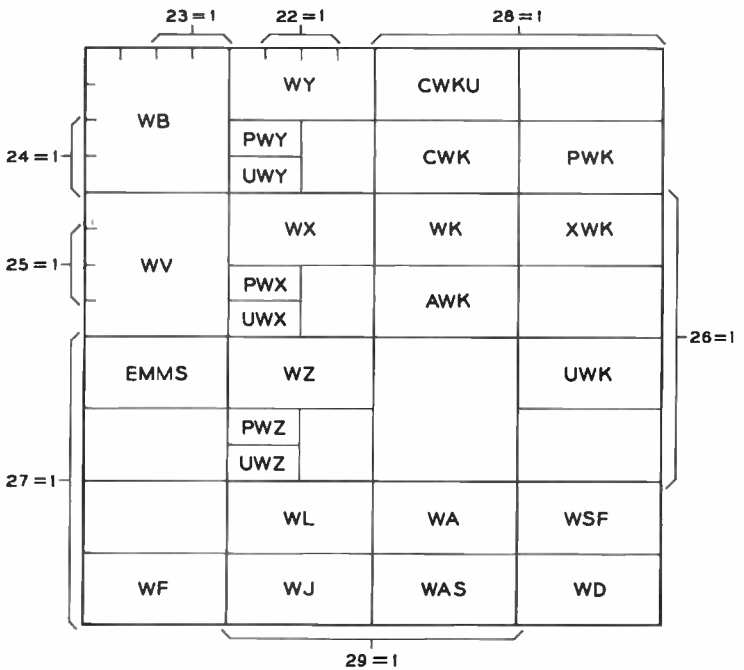


Fig. 24 — Word orders ($R_D \neq R_I$); (31-30 = 11).

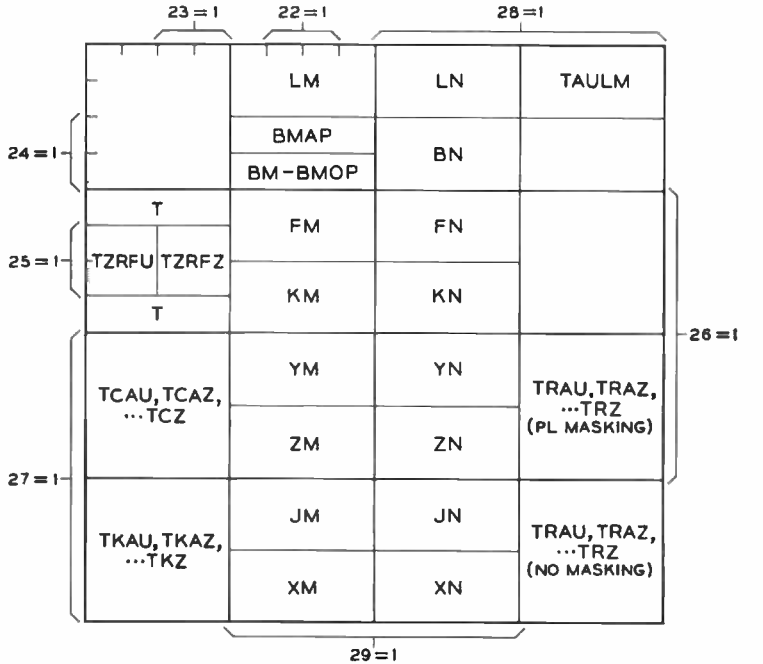


Fig. 25 — Writing orders and regular transfer orders (31-30 = 00).

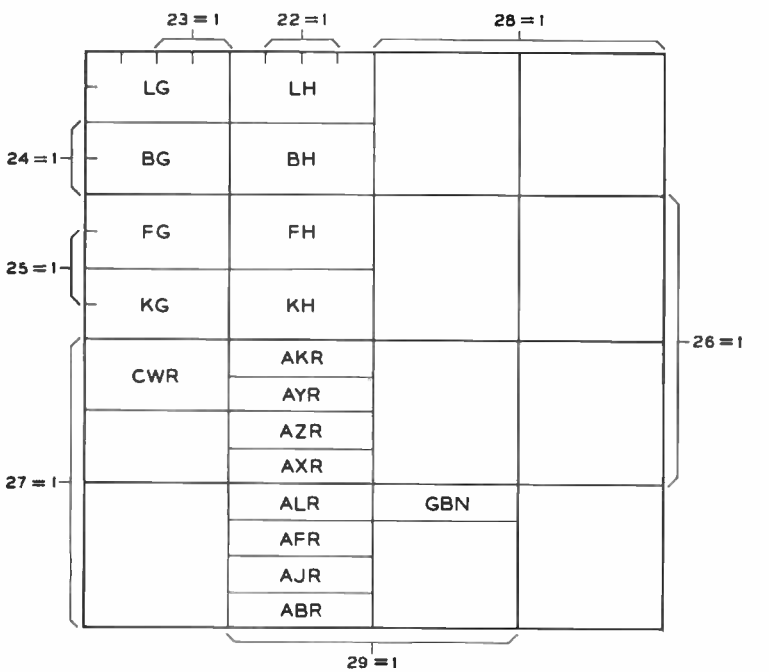


Fig. 26 — Maintenance writing orders and word orders ($R_D = R_I$); (31-30 = 10).

The remaining classes of orders require less coding space than the two classes just described and consequently occupy only portions of the remaining maps in Figs. 25 and 26. The combination 31-30 = 00 includes all regular transfer orders, all but a special class of memory writing orders, and a small class of miscellaneous orders including rotate and shift orders. Fig. 26 represents the binary combination 31-30 = 10 and includes the class of maintenance writing orders and the word orders ($R_D = R_I$). Figs. 23 through 26 include special classes or orders described below. The maintenance writing orders include control mode memory orders (represented by the letter N in the mnemonic equivalents), G-mode memory orders, and H-mode memory orders (see Section 7.7).

The early transfer orders are encoded according to the data processing actions to be taken whenever the decision is made not to transfer. Thus the order TCMMF is encoded as a memory reading order (31-30 = 01), and the order TAULM is encoded as a memory writing order.

With the encoding just shown, the decoders were designed to optimize whatever decompositions are made available for both the optional data processing gates and in the selectors and gates common to related orders. In addition to the buffer order word decoder and the order word decoder, two classes of data processing functions must be included to complete the above summary. The memory address decoder controls the generation, transmission, and central control response of each program store command and each call store command; the data processing for memory reading orders, memory writing orders, and indirect transfer orders includes the use of the memory address decoder. The decoders described here carry out those gating actions necessary to obtain and process single-cycle orders. Many classes of orders cannot be executed in a single machine cycle; the additional gating actions for such orders are handled by a group of sequencers; these actions may include automatic retrial and/or correction of program order words and data words.

VII. MAINTENANCE OBJECTIVES AND CIRCUITS

The No. 1 ESS must be able to provide continuing service to customer lines in the face of occasional and random occurrences of circuit troubles.[14] Duplication of subsystem units or portions of such units provides a set of potential replacement parts. Whenever a circuit trouble occurs within a subsystem, the detection of that trouble is followed by the required replacement; this replacement is made at electronic speeds to minimize the interfering effect of the trouble.

Each subsystem includes maintenance circuits which: (1) serve in detecting symptoms of circuit troubles as they appear and (2) aid in determining the location of such trouble to facilitate repairs

The detection of circuit troubles leads to the execution of maintenance programs which first determine whether a fault exists and if so whether the circuit trouble occurred within the active (controlling) switching system or within a standby duplicate unit. If a fault has occurred in an active subsystem unit, the next step is the necessary switching of associated active and standby units. The system is therefore quickly restored to an operable state and returned to the normal business of processing telephone calls; the subsystem unit in trouble is placed in an out-of-service state; and finally, special program sequences are interleaved with call processing programs to determine the faulty circuit element. The maintenance actions last described constitute a diagnosis of the out-of-service unit by the switching system; the results of this diagnosis appear as a printout on a special teletypewriter unit. Corresponding to each such printout is an entry in a specially prepared dictionary[14] which the maintenance man consults; the "definitions" in this dictionary are a listing of plug-in circuit packs to be replaced.

The entire procedure just described, from the detection of a circuit fault to the replacement of associated circuit packs, takes place in a matter of a few minutes; on completion of the repair of the out-of-service subsystem unit it is returned to the standby state for protection against future occurrences of circuit troubles. The maintenance circuits and associated program sequences serve in meeting a primary maintenance objective: essentially continuous telephone service with a minimum degradation in the quality of service in the presence of occasional circuit troubles.

Certain of the maintenance actions operate continuously and independently of the program sequences being executed in the central processor; other actions are obtained with the maintenance circuits and special program sequences. The integration of such program sequences and maintenance circuits is described elsewhere in this issue.[14]

7.1 *Circuit Checks of Communication Channels between Central Control and Connecting Subsystem Units*

Communication of information between central control and the remaining subsystems comprises the transmission of commands and addresses to one or more such units; each command specifies the required circuit response, and each address specifies the location or locations

which are to respond to the command. These responses include setting (resetting) flip-flops, scanning a group of ferrods, reading or writing a 24-bit word of call store data, etc. Accordingly, the circuit responses in some instances include the return of information to the subsystem unit generating the command and address. Circuit check signals accompany redundantly encoded commands, addresses and responses transmitted between the central control and (1) the program stores, (2) the call stores, (3) the central pulse distributors, and (4) peripheral units such as scanners, network controllers, and signal distributors. Maintenance circuits in both central control and the connecting units provide a continual check of communication and serve as a safeguard against noise and circuit troubles in the communication channels. Since the rate of communication between central control and its connecting units is quite high, most occurrences of circuit troubles within (1) the circuit generating the redundancy in the commands, addresses, and responses; (2) the registers for transmitting and receiving these commands, addresses, and responses; and (3) the associated check circuits, will quickly result in the detection of a check failure. For example, the check of communication between central control and the program store is briefly outlined in Fig. 27. The decoder combines clock signals from the microsecond clock with dc inputs to generate synchronizing, command and mode signals. These are transmitted along with a 16-bit address and four-bit code to select the appropriate program store information block via cable drivers connecting a twisted-pair cable leading to the program stores. The selected program store responds with synchronizing and check signals and a 44-bit reading from the twistor memory which is returned to the buffer order word register of central control. As this response is being returned, the contents of the program address register are transmitted to an auxiliary storage register so that the next information block code and address may be gated to the program address register. This last gating action permits the simultaneous check of one program store response and the generation and transmission of a following program store command. Under control of timing signals derived from the decoders, the check circuits carry out single-error and double-error checks of the information contained in the auxiliary storage register (corresponding to the address of the word obtained)[15] and the buffer order word register. The check circuit also verifies that the program store response included a check (all-seems-well) signal from the responding program store. When one or more of these checks fail, signals on the corresponding check-fail conductors lead to the required remedial circuit action.

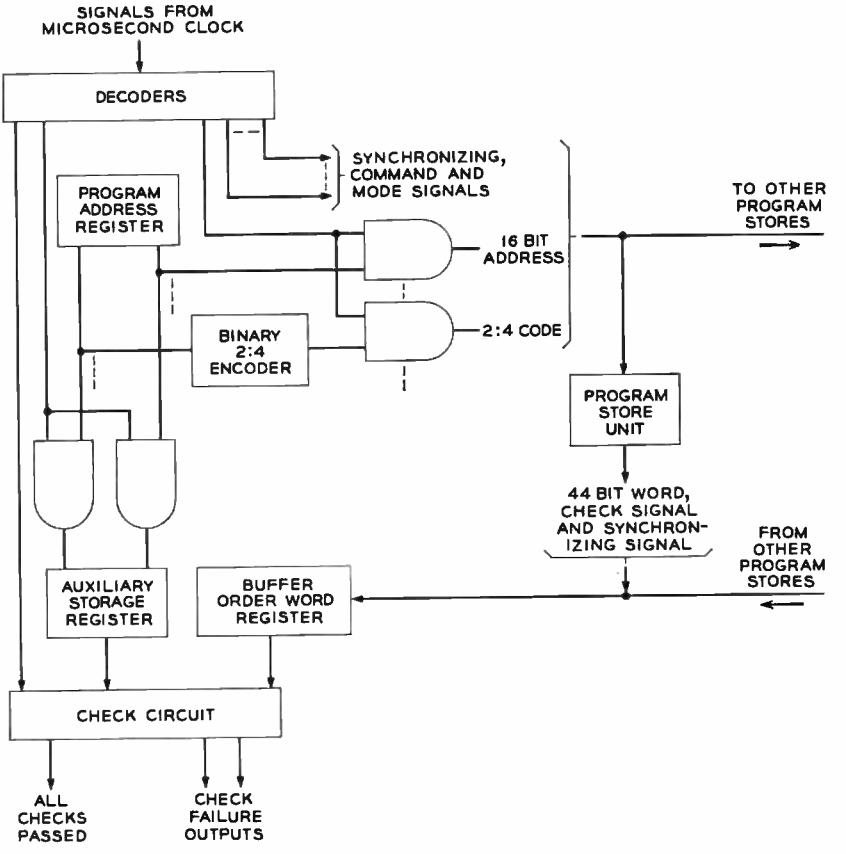


Fig. 27 — Check of central control-program store communications.

7.2 *Interrupt Actions*

A simplified block diagram of the interrupt system is shown in Fig. 28; it includes the three flip-flop registers in central control that are part of the buffer register bus system. This access permits single-cycle reading or writing access to these registers similar to that available to call store memory locations.

The interrupt source register comprises a number of interrupt source flip-flops; input signals to this register arrive from the millisecond clock and various check circuits within central control. The interrupt-level activity register serves to record the level interrupt corresponding to the program sequences being executed in central control. That is, cor-

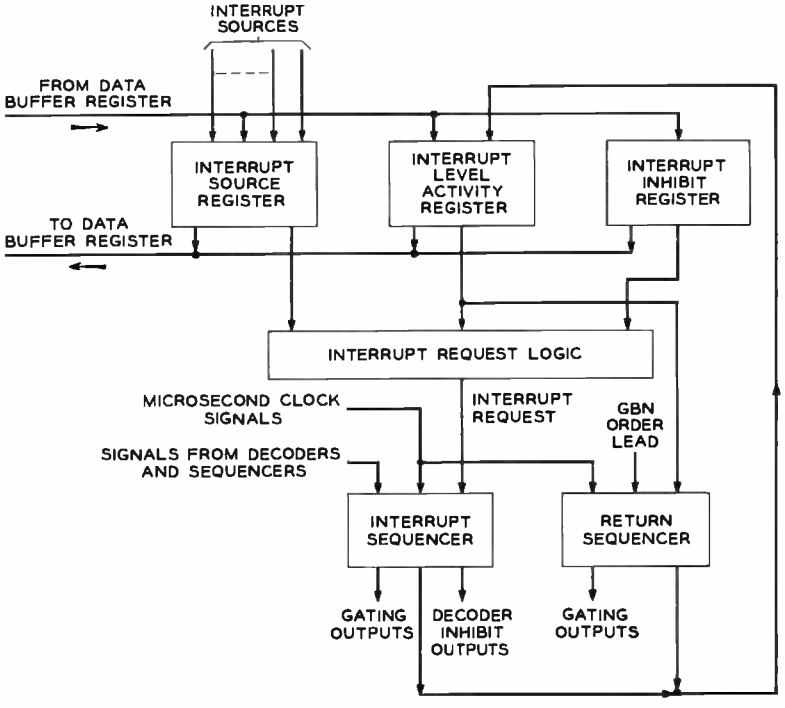


Fig. 28 — Interrupt system.

responding to each of the priority classes is a flip-flop within the interrupt-level activity register; whenever the interrupt system responds to an interrupt request, the wired transfer of program control is accompanied by the setting of the corresponding flip-flop in the interrupt-level activity register. The setting of flip-flops within the interrupt-level activity register also serves to inhibit the interrupt system from honoring interrupt requests from the level just served and all lower levels.

Assuming first that all flip-flops in the inhibit interrupt register of Fig. 28 are reset, the sequence of actions associated with each interrupt may now be described. When a given interrupt program sequence is required, a signal appearing on the corresponding interrupt source signal lead sets the corresponding interrupt source flip-flop. This signal propagates through the interrupt request logic and enables the interrupt request output conductor. The enabling of the interrupt request conductor is combined with clock signals and signals from the decoders and other sequencers to initiate the interrupt sequence. This assures that the interrupt allows any multicycle order or order following an

MCII or MKII instruction to go to completion before generating the wired transfer of program control. This consideration simplifies the hardware and program design for returning to interrupted program sequences.

Once activated, the interrupt sequencer carries out a number of functions extending over a period of several machine cycles; accordingly, the interrupt sequencer inhibits the order word decoder and buffer order word decoder outputs and generates independent gating signals to carry out a sequence of actions which include the following: (1) update the interrupt level activity register by setting the flip-flop in that register corresponding to the level interrupt currently being served (the interrupt request logic will then respond only to interrupt requests which may have a preassigned priority over the first interrupt program), (2) generate a transfer address corresponding to the entry point of the interrupt program sequence corresponding to the class of interrupt being served and gate this address to the program address register to effect this entry, (3) store the contents of the data buffer register in a first reserved location in the call store (this location depending upon the level of interrupt being served), and store the address of the instruction immediately following the last instruction executed prior to the interrupt.

Having completed these tasks, the interrupt sequencer returns to the inactive state, and the interrupt system is then responsive to further interrupt requests at higher levels. At this time, the entry to the corresponding interrupt program is made; this program begins the further storing of index registers, the logic register, etc., to complete the construction of the central control image in a set of reserved call store locations.

Upon completion of the required interrupt work functions, a program sequence restores the image of central control from the block of reserved call store locations. The interrupt program sequence then ends with a special return order (GBN), which activates another sequencer which completes the reconstruction of the central control image and transfers back to the interrupted program in an efficient three machine cycle sequence. This sequencer (also shown in Fig. 28) utilizes the interrupt-level activity register to complete the restoration of central control to the state occurring at the time of the interrupt. The sequencer must: (1) reinitialize the program address register to reenter the interrupted program at the proper point, (2) restore the data buffer register, and (3) reset the flip-flop in the interrupt-level activity register associated with the interrupt level from which the return is being made. Having completed these actions, the return sequencer advances to the

inactive state and is thereby made available for subsequent returns from other interrupt program sequences.

The inhibit interrupt register shown in Fig. 28, as its name implies, serves to selectively inhibit the response of the interrupt system to selected interrupt sources. The inhibit interrupt register is also a buffer bus register to which reading and writing access are provided. This register is used to selectively inhibit the interrupt sequencer response to interrupt signals during the execution of special test program sequences which, as part of their normal execution, cause the generation of interrupt source signals. The interrupt inhibit register also serves to inhibit interrupts due to repeated signals from defective subsystem units.

7.3 *Matching Circuits and Match Control Decoder*

In normal operation the duplicate central controls are executing identical orders within the same program sequences, and since the microsecond clock in both central control units is driven from one of two crystal oscillators,* the individual data processing steps for each order are closely synchronized in the two central control units. Normally, the two central controls are started by placing the same entry address in the program address register to simultaneously obtain and execute the same first program order. Each central control then continues in step with the other; the same data are read from memory or the scanners, the same data processing steps are performed on this data, and the outcome of each decision order is identical. Furthermore, certain trouble signals are cross connected between central control units so that any additional cycles inserted in one central control unit for remedial actions are accompanied by the insertion of the same number of cycles in the other unit.

The mode of operation just described is designated as the "in-step mode" and is utilized with the matching circuits to provide a continuing hardware check of the operation of the two central controls. This check consists of repeated comparisons of like information processing points in both central control units to obtain rapid detection of trouble conditions within either unit. The repeated comparisons are made with the matching circuits under the control of the match control decoder. A simplified block diagram of these matching circuits, decoders, and cross-connecting buses is shown in Fig. 29.

Within each central control are two internal match buses which pro-

* A flip-flop in each central control defines one unit as the active central control and the other unit as the standby central control. The crystal oscillator in the active central control drives the microsecond clocks in both units.

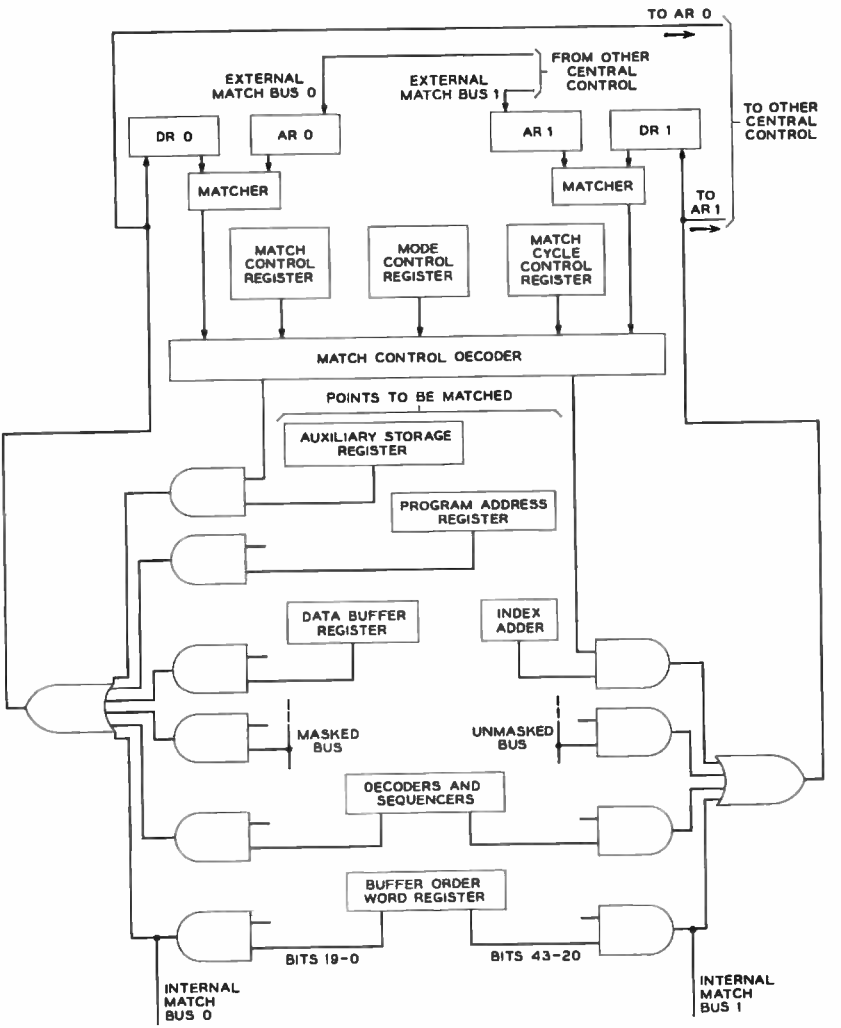


Fig. 29 — Central control matching circuits.

vide access to selected information processing points; these are labeled internal match bus 0 and internal match bus 1. Under control of the match control decoder, information from selected points is transmitted to these internal match buses, and from there other gates are enabled to place this information into internal match registers DR0 and DR1 and simultaneously transmit the sampled information via external match buses 0 and 1, respectively, to the other central control unit. The

match control decoders in both central controls are normally operating in step so that the information transmitted from the first central control to the second is stored in external match registers AR0 and AR1, respectively, in synchronization with the previously described gating actions. Two match circuits serve to compare the contents of AR0 with DR0 and AR1 with DR1; according to the state of the mode control register and the presence or absence of the match condition, the match control decoder generates the corresponding output signals. For example, when the matching circuits are employed in the routine matching mode, a selected sequence of common match points in each central control is matched at the rate of 2 matches per cycle; the detection of a mismatch condition generates a maintenance interrupt signal and further matching is automatically halted.

Since the matching circuits are limited to a maximum rate of two matches per machine cycle, the routine matching mode selects the specific sequences of internal points to be matched; the points to be matched depend upon the program and hardware actions being taken in central control to strategically examine those points most pertinent to the data processing steps that are occurring during a given machine cycle. Signals from the decoders and sequencers within central control are transmitted to the match cycle control register shown in Fig. 29. These signals set and reset specific flip-flops, which in turn direct the match control decoder and the selection of internal points for matching during each machine cycle.

The selection of points to be matched provides the detection of hardware troubles developing within central control as soon as the effect of that trouble would be communicated to other units in the switching system. It should be noted that the routine four-cycle match does not include the matching of all points to which the internal match buses have access. These additional points serve in other match modes described below.

7.4 *Maintenance Matching Modes*

A number of maintenance matching modes provide program-controlled access to the array of register buses and test points connected to internal match bus 0 and internal match bus 1. During the performance of certain maintenance programs, the routine matching mode is inhibited, and instead one of a number of maintenance matching modes may be selected to use the matching circuits to monitor test points within the standby central control or use the matching circuits and

connecting buses to communicate selected data from certain of these points from one central control unit to the other.

The maintenance matching mode to be executed and optional gating actions ensuing the detection of mismatch (or match) conditions are selected by the information placed in the mode control register. Certain of the maintenance matching modes match selected points at selected time intervals; information placed in the match control register determines these selections.

The selection of the routine matching mode or one of the maintenance matching modes is made by writing the selected word into the mode control register and the match control register shown in Fig. 29. To switch from one matching mode to another, a special flip-flop in both central controls is reset by a central pulse distributor command; this inhibits the response of the match control decoder to the matching mode currently specified. This step is followed by updating the mode control register and match control register to the new matching mode desired; following these actions, the CPD-controlled flip-flop is again set and the match control decoder is responsive to the new mode. A description of each of these modes and its use appears in a companion paper.[14]

7.5 *Emergency-Action Sequencer*

The preceding sections describe a number of hardware checks and allude to both hardware and program remedial actions in response to circuit troubles detected in the central processor. These remedial actions include program sequences which are calculated to isolate circuit troubles to a particular unit and to control any required switching of units to obtain a working system. The execution of these sequences is in itself possible only if the active central processor includes an operable combination of the central control, certain program stores and the interconnecting bus system. That is, if the fault itself lies in one of these units, the central processor may be incapable of performing the necessary rearrangements. The emergency-action sequencer serves as a hardware checking and corrective means to handle this problem.

A block diagram of the interconnections between the emergency-action sequencer and its inputs and outputs is shown in Fig. 30. The inputs consist of four hardware checks; the failure of any one or more of these checks generates trouble signals to activate the emergency-action sequencer. These signals are de-connected to the inputs of a monostable pulse circuit; the output of this monostable pulse circuit is connected to a series of monostable pulse amplifiers to provide a sequence of output

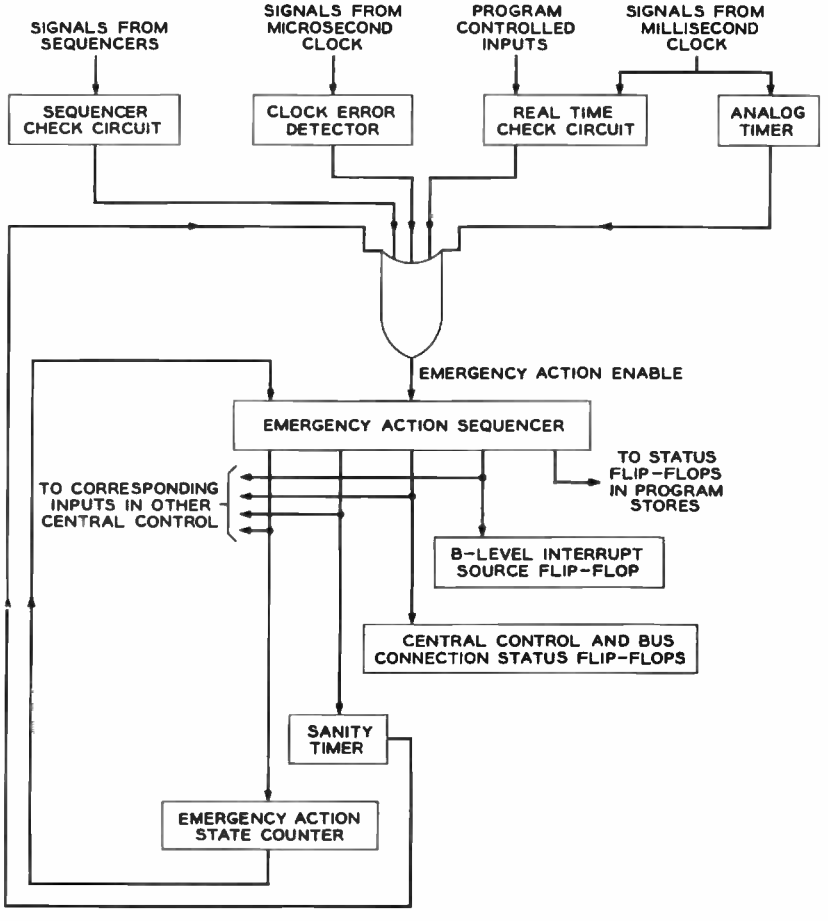


Fig. 30 — Central control emergency-action system.

pulses. The emergency-action sequencer is therefore operable in the presence of circuit troubles in the microsecond clock.

Each sequence of output pulses generated by the pulser and delay line defines an emergency action. Each emergency action increments an emergency-action state counter, initiates a B-level interrupt, and sets and resets selected status flip-flops in the duplicate central control. Switching these flip-flops results in a corresponding rearrangement of active and standby central controls, program store input and output connections, and selection of connecting buses. The rearrangements made during each emergency action depend on the internal state of the

emergency-action state counter. It is quite possible that a given rearrangement of duplicated units in the central processor will not result in an operable combination of subsystems; the result will be the reactivation of the emergency-action sequencer which, under control of the emergency-action state counter, forms a new arrangement.

When an operable combination of units have been connected together to become the active central processor, the execution of a test program (initiated with a B-level interrupt) can be completed in approximately 100 machine cycles. If the active central processor contains faulty equipment which inhibits the proper execution of the test program sequences, the sanity timer will recycle in approximately 128 machine cycles. Such recycling reactivates the emergency-action sequencer.

A more detailed description of sequencer actions and the additional hardware and program checks performed with the emergency-action sequencer are covered in a companion paper.[14]

7.6 *Maintenance Orders*

Included in the order structure of the central processor are classes of orders designed specifically for use in maintenance program sequences to obtain test results or to place test signals within central controls, call stores, and program stores. These orders perform special gating actions that are either inconvenient or impossible to obtain with combinations of other orders described above. The maintenance orders include the following classes: (1) G- and H-mode memory reading and writing orders, (2) control mode memory reading and writing orders, and (3) miscellaneous test and test signal orders.

7.7 *G- and H-Mode Memory Orders*

As previously noted, all program and data information is duplicated; each word stored in one program store unit or call store unit is also stored in another store unit. There are many situations where initial installation requirements or growth in an office require an amount of duplicated semipermanent and/or temporary memory that could be satisfied with an odd number of store units. The duplication scheme employed in the No. 1 ESS central processor permits the use of an odd number of units to gain economy over a system using only an even number of stores.[14] In this duplication scheme each store unit is divided into two blocks of memory. Each block or half-store is assigned a different code. One block in the memory unit is designated the H block; the remaining block is labeled the G block of memory. All the information appearing in the H block of one store unit is duplicated in the G block of another store unit.

Commands to read or write in memory include a code and address. Each code and address corresponds to one word of information in memory, but since each word is duplicated in the memory, the code and address correspond to two memory locations, one appearing in the G block of one store unit and the other location appearing in the H block of a different store unit.

The usual communication of data and program words between the central control and connecting stores is accomplished with normal mode commands. Each normal mode command is capable of generating a simultaneous response in the store units containing the G and H duplicate memory locations; however, no store unit simultaneously receives two commands to which it is to respond, and no central control unit simultaneously receives both of the duplicate responses.

When troubles are detected in the communication of information between the stores and the central controls, the remedial actions include program sequences to examine individual store units; it is desirable to read or write test information specifying the store unit which is to respond to these commands. Consequently, the order structure includes G and H memory reading and writing order words or instructions that specify which duplicate location is to be read or written. These memory reading and writing orders are processed in central control like the normal mode reading and writing orders: the codes and addresses are generated in the same fashion, but the command includes G- or H-mode signals. For example, a G-mode memory reading order will obtain information from only the duplicate unit which contains the G image of the memory location specified by the code and address. H-mode memory reading and writing commands make a similar distinction.

The G- and H-mode memory reading orders include additional features when applied to the reading of data in program stores. Certain of these orders call for the reading of data from the G or H locations, respectively; when the data is obtained from a program store, the readings are accepted and processed as valid data without carrying out any rereading or correcting steps as indicated by the check circuit. Other G and H memory reading commands, when applied to program store memory locations, permit the correction of data as required, but no rereadings may take place. The G and H memory reading and writing instructions just described are given in Table II.

7.8 *Control Mode Orders*

Control orders resemble the normal mode memory reading and writing orders in that central control carries out the reading or writing of data in

TABLE II — G AND H MEMORY ORDERS

Mnemonic Representation	Order	Available Options
HKU	Read H image of specified memory location; no remedial actions for invalid responses; data reading replaces contents of accumulator	Indexing; index register modification options, including incrementing, W, and register S options; product (PL and PS) masking and complementing of data reading
GKC	Read G image of specified memory location; remedial action limited to single-error correction of program store data readings; data reading replaces contents of accumulator	
XGKU	Read G image of specified memory location; no remedial actions for invalid responses; EXCLUSIVE-OR of data reading and accumulator contents placed in accumulator	
XHKC	Read H image of specified memory location; remedial action limited to single-error correction of program store data readings; EXCLUSIVE-OR of data reading and accumulator contents placed in accumulator	
BG, BH	Place contents of data buffer register in specified G (H) call store memory location; no remedial actions for invalid responses	Indexing; index register modifications listed above
FG, FH KG, KH, LG, LH	Place contents of index register F (accumulator K, logic register L) in specified G (H) call store memory location; no remedial actions for invalid responses	Indexing; index register modification options listed above; product (PL and PS) masking; complementing and insertion (EL and ES) masking of data to be stored

such units as the program store and the call store. These orders differ from normal mode memory order in that: (1) when commands are transmitted, the mode signals which appear as part of these commands indicate the control mode, and (2) control mode memory writing orders may be used also to treat flip-flop registers in the stand-by central control as memory locations in a call store.

Control orders are designed to provide the convenient setting, resetting, and reading of status flip-flops and other test points in the program stores, call stores, and standby central control.

The control mode orders are listed in Table III; this table includes comments regarding the specific application of certain control mode

TABLE III — CONTROL MODE ORDERS

Mnemonic Representation	Order	Available Options
NB	Read specified control location; place reading in data buffer register	Indexing; index register modification options, including incrementing, W, and register S options
NF, NJ, NK, NL, NX, NY, NZ	Read specified control location; place reading in index register F (J, K, L, X, Y, Z)	Indexing and index register modification options: product (PL and PS) masking and complementing of data
NBTA, NBTB	Special control reading orders to test call store address translators; translated address placed in data buffer register	Indexing and index register modification options
BN	Place contents of data buffer register in specified control location	
FN, JN, KN, LN, XN, YN, ZN	Place contents of index register F (J, K, L, X, Y, Z) in specified control location	Indexing and index register modification options: product (PL and PS) masking, complementing, and insertion (EL and ES) masking of data to be stored
WNPS	Transmit control command (specifying) location and data for control flip-flops in a specified program store	Indexing

orders. For example, the order WNPS is designed specifically for writing information into duplication status and test flip-flops within the program stores. The program stores are the semipermanent memory of the system. No on-line writing of information within the twistor memory is possible and therefore none of the previously described writing orders has access to the program store. The WNPS order is executed by sending a command on the program store bus which indicates a control mode writing operation, and part of the address transmitted is treated by the responding program store as the data to be placed in control flip-flop registers within that store.

7.9 *Miscellaneous Maintenance Instructions*

In addition to the two classes of special memory reading and writing instructions described above, there are a number of specific orders, including some normal mode memory reading and writing orders designed

specifically to create trouble conditions not encountered in other normal memory commands or to initiate special tests.

Executing the order WV causes a data word to be transmitted to a special V register in central control; the outputs of this register are then transmitted as half-microsecond pulses to special points within the central control and via cable drivers and connecting twisted-pair cables to the other central control unit. This order is used, for example, for transmitting signals from the active central control to standby central control to:

(1) start or stop data processing in the standby central control;

(2) reset certain registers in the standby central control unit such as the buffer order word register and the order word register, or

(3) generate maintenance interrupt signals in the standby central control.

A mismatch sampling order, EMMS, concurrently carries out a number of information processing steps to initiate the mismatch sampling mode as described in a companion paper.[14]

The remaining miscellaneous maintenance orders (BMAP, BMOP, MBOP, and MBCS) comprise normal memory reading or writing orders which are specifically designed for exercising or examining the parity generation and check circuits in both central control and the call store.

VIII. TIMING CONSIDERATIONS

The central processor is a synchronous data processing system; a microsecond clock in central control provides clock pulses defining a machine cycle and intervals or phases within that cycle. A significant aspect of both the logical organization and detailed circuit specifications of the central processor is the integration of the response times of program store and call store systems and the multiphase data processing steps of central control in response to each of a diversity of program orders. This integration began with the preliminary design of a central control and its order structure.

The classes of orders considered for inclusion in the order structure consist primarily of different combinations of meaningful data processing operations or steps such as indexing, index register modification, and the placing of a data word obtained from memory into a specific index register in central control. Each major data processing step is assigned to one or more clock phases; the minimum time for a clock phase is fixed according to the maximum propagation time of information through the longest logic chains corresponding to the data processing

steps assigned to that phase. These phases are then fitted into a machine cycle which approximately equals the minimum cycle times of communication between the central control and both the call store and program store; the relative placement of each of the phases is dictated by the time of appearance of related data processed in central control to form commands, addresses and data for communications between central control and its stores.

In the design of classes of order words and the determination of the number of clock phases, an extra clock phase is provided by overlapping operational steps of successive orders. To maximize the average data processing rate, most of the orders are designed to be executed at the rate of one order per machine cycle. Certain orders, such as transfer orders and orders to read data words from the program store, require more time and are designed to fit into a timing framework of two or more machine cycles.

Detailed timing studies based on min-max component tolerances served to provide a "paper simulation" of the central processor. This simulation revealed that a number of logic chains and communication paths are limiting in fitting all the operations steps into a 5.5-microsecond clock cycle; accordingly, an improvement of any one logic chain or subsystem could not materially increase the central processor's data processing speed capabilities.

8.1 *Single-Cycle Call Store Memory Orders*

The central processor is designed to execute most of its sequences of program orders at the rate of one order per machine cycle; such orders are referred to as "single-cycle" orders. Multicycle orders require additional time (2 to 4 machine cycles, depending on the order) and are executed with the aid of special sequential control circuits described later in this article. The data processing time of the single-cycle orders determines the machine cycle time and the allocation of phase intervals within this cycle; included in this class are memory reading and writing orders which receive and transmit data from memory locations in the call stores.

Call store memory reading and writing orders comprise an estimated 60 to 70 per cent of the instructions executed in call processing program sequences; the ability to execute these orders at the rate of one order per machine cycle defines a machine cycle, which may be limited by one or more of three considerations:

(1) the maximum repetition rate of obtaining program order words from a program store system including up to 6 program stores;

(2) the maximum repetition rate of a call store system of up to 37 call stores in response to a random sequence of reading and writing commands generated in the execution of a sequence of call store memory orders; and

(3) the maximum turn-around time at which a data reading can be obtained from a call store and combined in an index adder to generate an address for use in a call store reading or writing command.

An additional requirement in allocating clock phase intervals is noted in fitting call store writing orders into the machine cycle. An upper bound is placed on the interval from the beginning of the clock phase which initiates indexing (to obtain a call store address) to the beginning of the clock phase which moves the data to be stored from an index register to the data buffer register.

There is no difficulty in fitting the data processing steps of the non-memory data processing orders into this framework; the only orders employed in significant numbers in the program sequences requiring more than one machine cycle for their execution are orders to transfer program control and memory reading orders which obtain data words from the program store.

8.2 *Phases of the Machine Cycle*

The overlapped execution of sequential program orders combined with 3 basic data processing phases per machine cycle provides four data processing intervals per instruction. This is a sufficient number of intervals for all the data processing steps required for transmitting information via one or both of the unmasked bus and masked bus from one register to another through logic combining circuits such as the index adder or the mask and complement circuit.

Fig. 31 shows how call store memory orders are fitted into the overlapped execution of single-cycle orders. An order is obtained from a program store during phase 1 of cycle 1 in response to a program store command transmitted from the central control in the preceding machine cycle. At the beginning of this phase, the buffer order word register is reset to erase the preceding order word, and the register inputs are connected to the program store response bus. Depending on the response time of the program store unit and its distance from central control, the order word arrives at some time during phase 1. In this interval, the command and address for the first succeeding program order are transmitted to the program store.

Phase 2 and 3 clock pulses are applied to the buffer word decoder to control indexing and index register modification during cycle 1. The indexing addition is completed towards the end of phase 2 and the sum

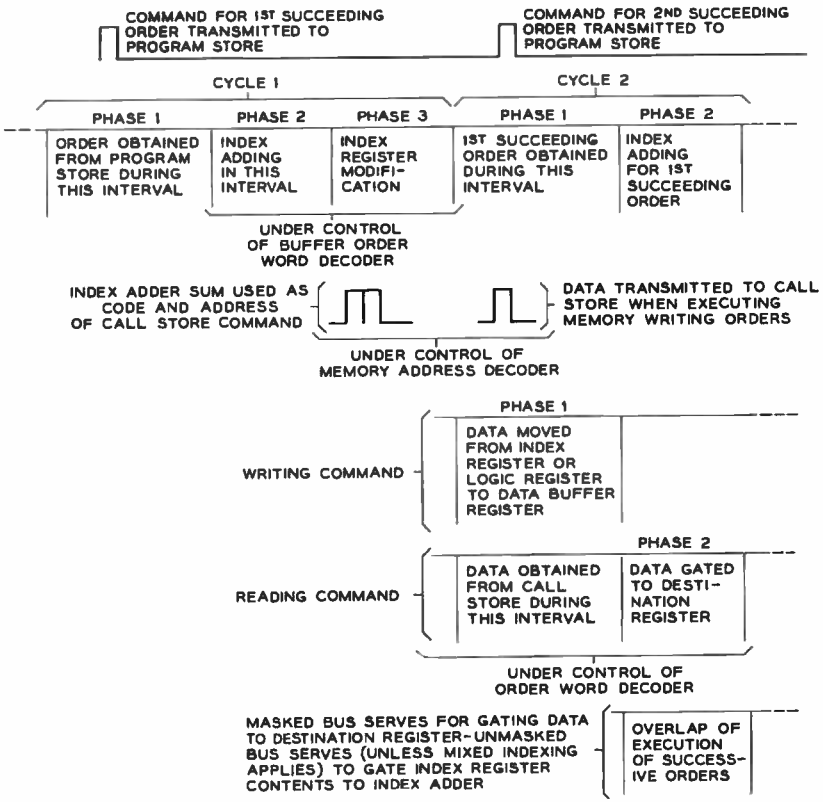


Fig. 31 — Timing of execution of single-cycle memory orders.

is gated to an index adder output register (contained in the index adder). This register serves as an address register for call store memory orders. Index register modification is performed during phase 3, and the call store command and address are concurrently transmitted under control of the memory address decoder to the call store.

At the beginning of phase 3 of cycle 1 the call store memory order depicted in Fig. 31 is transmitted from the buffer order word register to the order word register. Phase 1 and phase 2 clock pulses applied to the order word decoder and memory address decoder carry out the remaining steps for the order. Two sequences of these remaining steps are shown: the first sequence is that of a call store memory writing order, and the second corresponds to a call store memory reading order.

Call store memory writing orders select the contents of one of the

seven index registers or the logic register to be transmitted as data to the call store. The data buffer register in central control is connected to the call store data transmission and call store response buses; accordingly, the contents of the selected register are transmitted via the unmasked bus, the mask and complement circuit, the masked bus and the insertion mask to the data buffer register. A parity generator connected to the outputs of the data buffer register produces a parity bit which is then transmitted simultaneously with the 23-bit data word to the call store to complete the call store memory writing order.

Call store memory reading orders employ phase 1 of cycle 2 to accept data words by connecting the inputs of the data buffer register to the call store response bus. The data buffer register is reset at the beginning of phase 1 and the data word is received at some time during phase 1, depending on the response time of the call store unit and the distance between that unit and the central control.

During phase 2 of cycle 2, the data reading is transmitted to the destination register selected by the memory reading order by gating the reading through the data buffer register and the mask and complement circuit onto the masked bus.

Concurrently with the completion of the memory reading order, the first succeeding order begins its indexing. That is, phase 2 of cycle 2 includes the transmission of the contents of a selected index register to the index adder via the unmasked bus. Certain pairs of orders occur where the first order reads data from a call store and the second order is a call store memory order which selects as its index register the destination register of the first order. (These pairs of orders require the previously described mixed indexing.) To execute such pairs correctly, the lower bound on the machine cycle time must equal or exceed the maximum round-trip time of call from call store command to response, including the indexing addition in central control.

Indexing and index register modification of all other orders are also performed during phase 2 and phase 3 of the first cycle, as shown for memory orders in Fig. 31. W class combining orders move the data word to a specified destination register during phase 1 of the second cycle; shift and rotate orders are also completed in the same interval. The earlier completion of these orders (as compared to memory reading orders) permits the accumulator to be selected as the index register in the succeeding order; such a selection is not feasible when the first order is a memory reading order moving data to the accumulator, since the accumulator will not contain the correct result in time for indexing during phase 2.

The data processing steps of orders such as UWX, PMZ include moving the contents of an index register (X,Z) to the logic register; this step is performed using the unmasked bus during phase 3 of the first cycle. The unmasked bus is available, since the index register modification step requires the use of only the masked bus; accordingly, the two steps may occur concurrently, as required by the order.

Whenever an order specifies that the data word is to be transmitted to the logic register (PS or ES masking) this step is accomplished during phase 3 of the first cycle under control of the buffer order word decoder.

Decisions are made by the order word decoder for regular transfer orders at the beginning of the second cycle; decisions for early transfer orders are made at the beginning of phase 3 of the first cycle. In both cases, the decision to transfer is implemented by activating a sequencer at the stated times.

8.3 *Basis of Timing Specifications*

Figs. 32 and 33 show the response times of the program stores and call stores in terms of minimum and maximum calculated response times. To provide sufficient spacing for large numbers of program stores and call stores, bus lengths from a few feet to a maximum of one hundred feet were assumed for each bus. The buses are all twisted-pair cables having a delay constant of approximately 2 nanoseconds per foot.[12] A single program store command bus system connects both central control units to all program stores; each store has a lightly coupled set of pulse receivers connected in series with the bus system, and the bus ends with a terminating resistor. These receivers add a small fixed delay to the bus system, 1.5 nanoseconds per receiver, which is insignificant in most timing considerations. Similarly, a single program store bus system connects all program store memory outputs (cable drivers are shunt connected) to inputs in both central controls. A similar arrangement of shunt and series connections of cable drivers and pulse receivers connects the central controls and the call stores. These connections include a call store command bus system, a call store response bus system, and a call store data transmission bus system.

To predict the delays of logic chains in central control, the characteristic minimum and maximum delay times of the low-level logic (LLL) AND-NOT gate[6] and the medium-current logic (MCL) AND-NOT gate were determined, as tabulated in Fig. 34. These limits include allowance for lead capacitance, and wiring rules restrict the actual capacitances to less than this allowance.[6] The characteristics of the cable driver (CD), cable pulse receiver (CPR) and clock pulse output amplifier (CPOA)

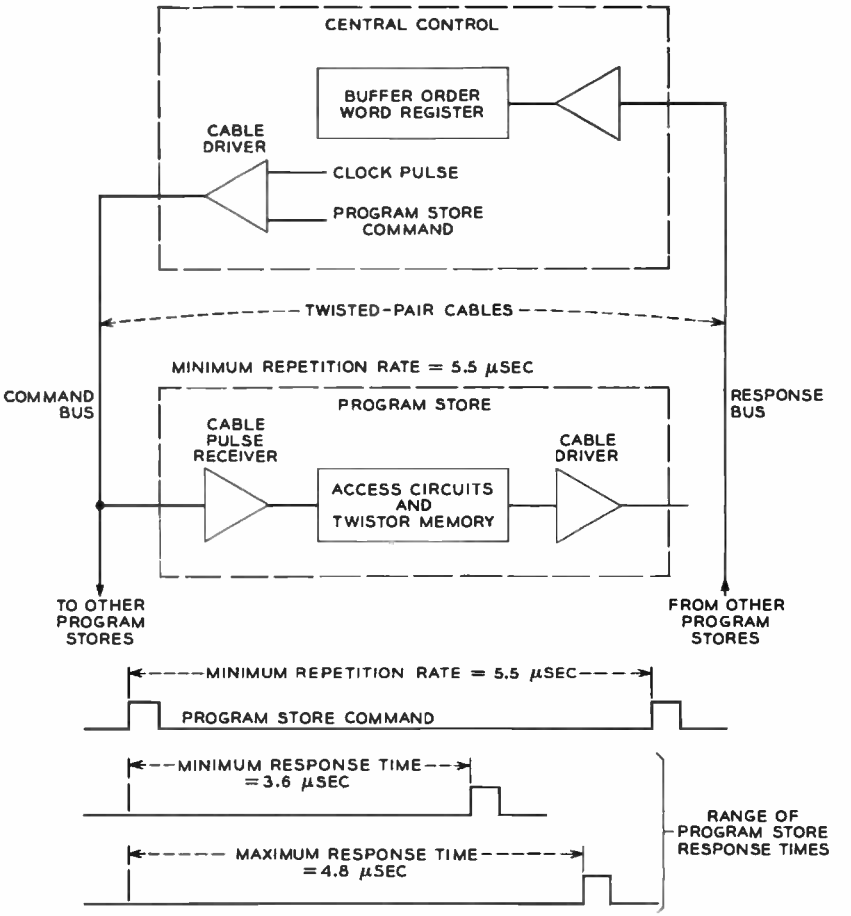


Fig. 32 — Program store timing tolerance limits.

gates complete the information needed for calculating propagation delays of central control.

In certain timing chains, minimum pulse widths become a part of the design considerations, and therefore a pulse-shortening characteristic for each class of gate is included in Fig. 34. It is assumed that the turn-on time of an LLL gate or an MCL gate cannot exceed its turn-off time; a consequence is that reduction in the width of a pulse propagating through a number of stages of logic gates due to differentials in turn-on and turn-off times is calculated to occur only in alternate stages which have negative-going pulse inputs.

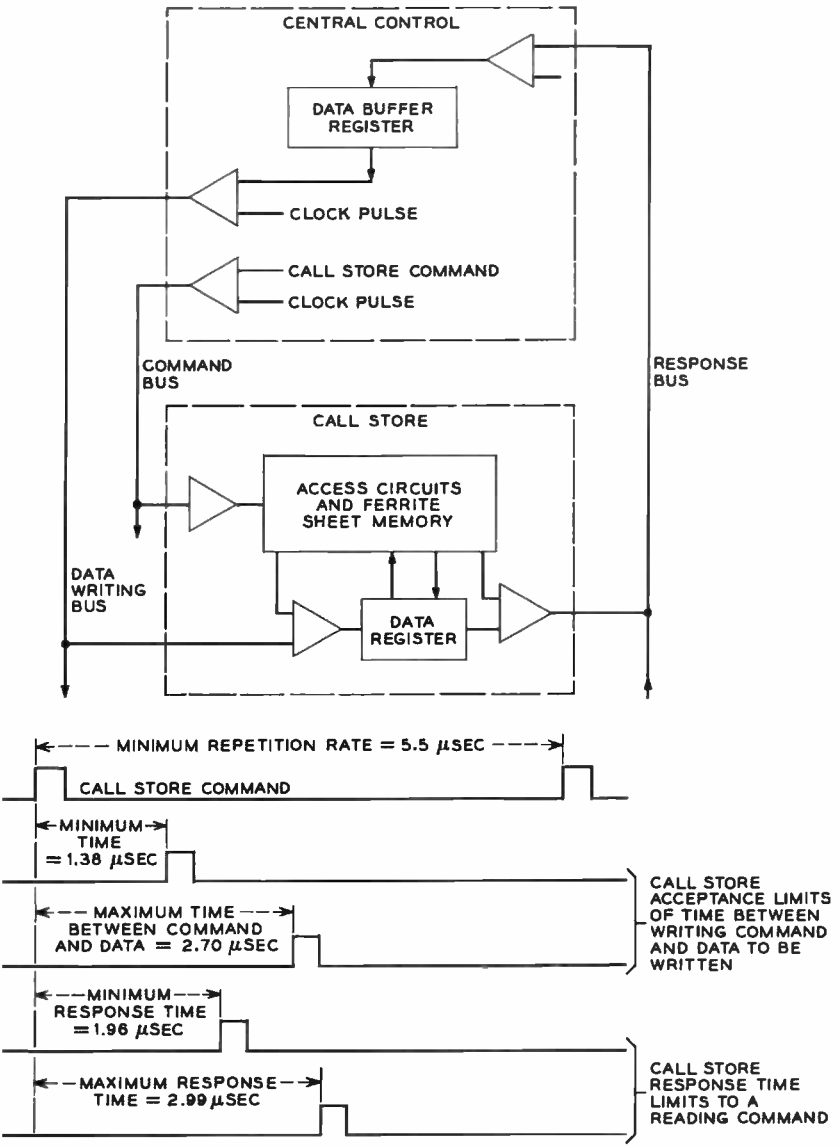


Fig. 33 — Call store timing tolerance limits.

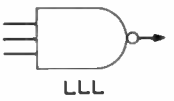
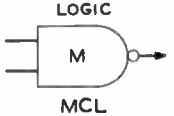
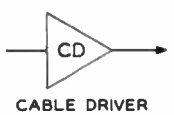
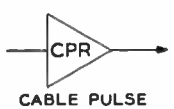
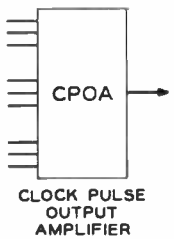
	TURN-ON TIME IN NANOSECONDS	TURN-OFF TIME IN NANOSECONDS	MAXIMUM SHRINKAGE OF PULSE WIDTH
	10-65 NS	10-65 NS	55 NS – FOR NEGATIVE-GOING INPUT PULSES ONLY
	10-75 NS	20-85 NS	75 NS – FOR NEGATIVE-GOING INPUT PULSES ONLY
	10-75 NS	10 NS-UNSPECIFIED	NONE
	10-65 NS	10 NS-UNSPECIFIED	NONE
	10-75 NS	10-75 NS	20 NS

Fig. 34 — Logic package timing characteristics.

8.4 *Microsecond Clock Characteristics*

The multiphase clock in central control defines the machine cycle of 5.5 microseconds and the three phases within the machine cycle. In addition, a number of shorter pulses are provided by the clock to carry out the gating of commands and addresses from central control through cable drivers onto twisted-pair cables connecting to stores, central pulse distributors and peripheral units such as scanner and network controllers. Other short pulses are required for resetting registers and gating informa-

tion from one flip-flop register to another at times other than the three principal phases; data processing results are thereby transmitted as they are completed. Half-microsecond pulses spaced at quarter-microsecond intervals over the entire machine cycle are provided to permit maximum flexibility in the logic design; timing chains were easily modified as additions and alterations were made in the development of the logical organization of the central processor. This large array of pulses thus satisfies the need for (1) selecting optimum gating times for internal data processing transmission and (2) obtaining a useful minimum pulse width (where reasonable clock pulse tolerances are a consideration) for (a) gating information from one register to another, (b) resetting a register, and (c) transmitting a pulse of sufficient width over twisted-pair bus systems to the most distant stores, central pulse distributors, and peripheral units.

The microsecond clock for central control includes a crystal-controlled 2-megacycle oscillator driving an 11-stage counter. The outputs of the counter drive logic chains which translate the states of the counter into the array of half-microsecond pulses and the basic phase pulses are shown in Fig. 35. The machine cycle is divided into 22 quarter-microsecond

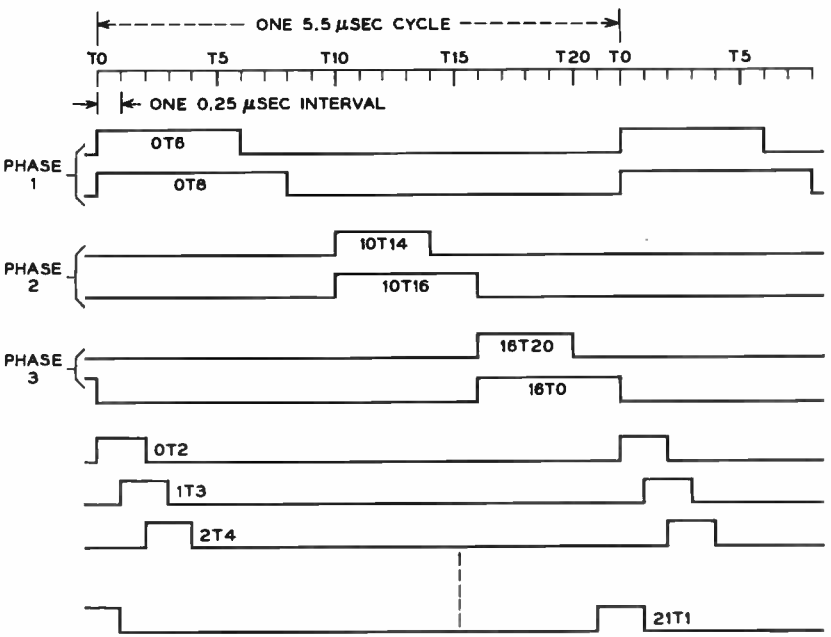


Fig. 35 — Phases of central control microsecond clock.

intervals and the beginning of each interval is denoted as T0, T1, · · · , T21. Each clock pulse begins and ends at one of the times T0, T1, · · · , T21, so that each clock pulse is labeled ATB, where A is a number corresponding to the time of the leading edge of the pulse and B corresponds to the trailing edge time.

Both central controls contain a complete clock, but the oscillator in the active central control drives the 11-stage counter in both central control units to keep the units closely synchronized. To keep the counters in step, a phasing signal is transmitted from the active central control counter to the standby unit once every machine cycle.

The tolerances on the clock outputs in each central control and the cross-connection tolerances are tabulated in Fig. 36. Fig. 35 depicts an ideal set of clock pulses and the table indicates additional delays from that ideal. That is, the figure represents all minimum delay conditions in the clocks, and only positive tolerances appear in Fig. 36. This approach simplified the many calculations to be made in the design. Only minimum/maximum values were substituted into delay equations, rather than nominal values plus or minus a tolerance figure. Examples of this technique follow.

8.5 *Sample Timing Calculations*

The half-microsecond pulses are employed in transmitting multibit commands and addresses to units connecting to the central control. To meet high fan-out requirements, a clock output amplifier is usually interposed between the clock pulse output and the array of cable drivers to be pulsed. This connection is exemplified in Fig. 37 using the pulse 0T2 for purposes of illustration.

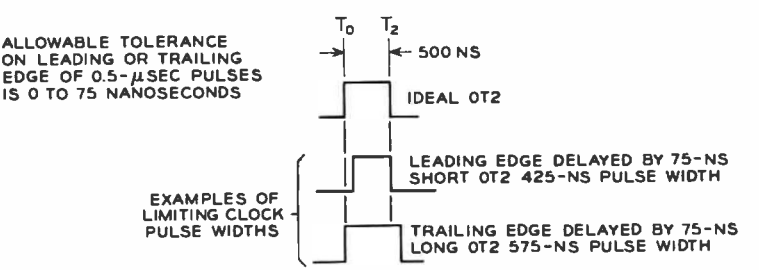


Fig. 36 — Microsecond clock tolerance limits. Allowable tolerance on leading or trailing edges of bus phase pulses (0T6, 0T8, 10T14, 10T16, 10T20, 10T22) is 0 to 150 nanoseconds. Allowable tolerance of propagation of oscillator signal from active to standby central control is such that idealized phases of Fig. 35 in standby central control lag by 0 to 75 nanoseconds.

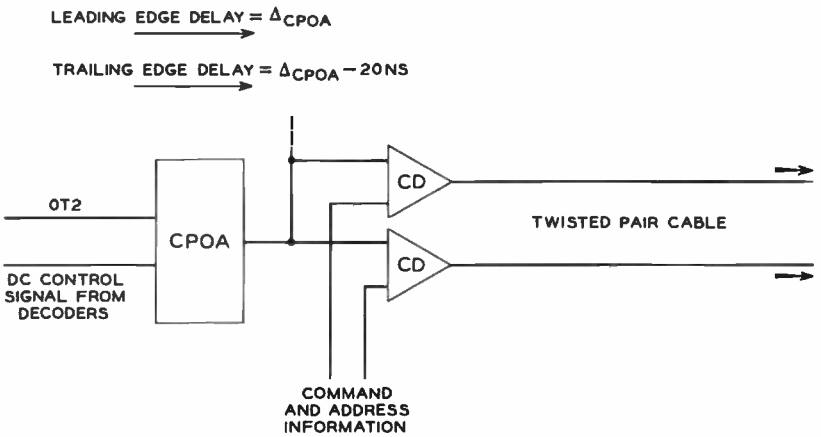


Fig. 37 — Clock pulse transmission chains.

A timing calculation was made to determine if such an arrangement would always provide command and address pulses of sufficient width (250 nanoseconds minimum) to a receiving unit located at the maximum allowable distance from central control. To determine this minimum pulse width, one first assumes the shortest allowable clock pulse and then calculates the maximum pulse shortening in logic chains and cables.

Referring to Fig. 36, the latest leading edge of the pulse would occur at T0 + 75 nanoseconds, and the earliest trailing edge would appear at T2, yielding a minimum clock pulse width of 425 nanoseconds. According to Fig. 34, the clock pulse output amplifier may shorten this pulse as much as 20 nanoseconds, and the cable driver would contribute no pulse shortening. Thus the minimum pulse width at the cable driver output is 405 nanoseconds.

Empirical equations for the shrinkage of the width of current pulses on a twisted-pair cable with an arbitrary number of series-connected cable receiver transformers as a function of that number (N) and the length of the twisted-pair cable (L in feet) have been derived.[16]

$$\text{Shrinkage in nanoseconds} = 50 \ln \left(\frac{950}{950 - L - 5N} \right). \qquad (1)$$

The bus system connecting central control to the peripheral units may be required to serve a large number (N) of such units and the bus length be correspondingly large. With limits of $N = 50$ and $L = 450$ feet the maximum shrinkage would be approximately 70 nanoseconds. Thus the

minimum pulse width at the input of the most distant peripheral unit would equal or exceed (405 − 70) = 335 nanoseconds, which meets the peripheral unit minimum pulse requirements.

The previously described round trip of one call store reading to be used in generating a call store command is outlined in Fig. 38 and results in a round-trip time of 5.36 microseconds, which fits into a 5.5-microsecond cycle. The calculated round-trip time does not allow for a slight delay in the loop deriving from clock pulse gating between the index adder and the index adder output register. Further, it should be noted that the late clock pulse arriving at point Y cannot be considered, since the memory reading may be obtained in response to a call store command generated in the active central control and utilized in the standby central control for generating a call store address. The 10 LLL delays in the index adder represent the maximum delay of carry propagation signals. Other adder designs with shorter delays were considered; however, smaller delays

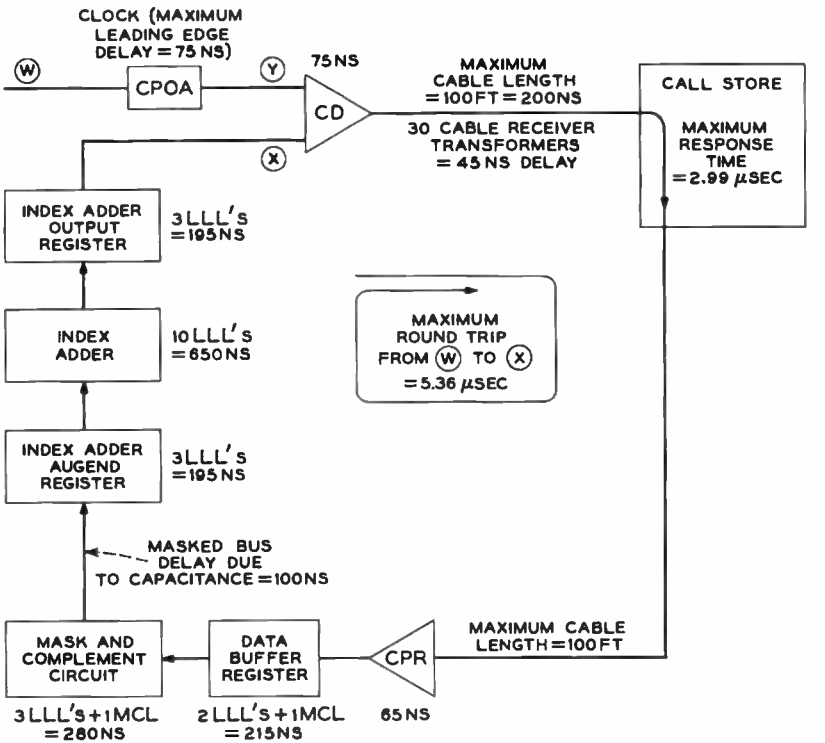


Fig. 38 — Call store round-trip time.

and a slightly shorter machine cycle could be achieved only with much larger adding circuits.

Other timing chains not described here also work with the 5.5-microsecond cycle but could be made to work with a shorter cycle only with additional logic circuits. Other limiting cases not described in detail here include:

(1) communicating commands to peripheral units once every two machine cycles to establish a maximum rate for supervisory scanning of customer lines,

(2) round-trip times of call store memory reading orders including the cross connection of associated check failure signals between central control units, and

(3) various combinations of related sequences of data processing steps in the execution of consecutive orders.

Fig. 39 illustrates some of the constraints that govern the subdivision of the clock phases within the machine cycle. The longer of the pair of pulses for each phase must not overlap, but time intervals between phases is allowed.

Phase two data processing includes the last step of a memory reading order where the contents of the data buffer register are transmitted via the mask and complement circuit and the masked bus to a selected destination register. This step requires a one-microsecond interval to complete the required propagation of information, and therefore phase two comprises the one-microsecond bus sampling interval 10T14 and a corresponding source-to-bus interval 10T16. A similar propagation delay requirement for index register modification options applies to phase

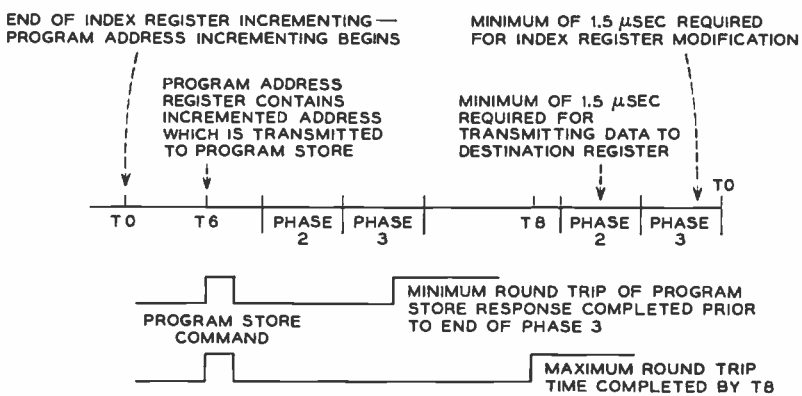


Fig. 39 — Fitting of clock phases into a machine cycle.

three, where the contents of the index adder output register are transmitted via the mask and complement circuit and the masked bus to the selected index register. Accordingly, clock pulses 16T20 and 16T22 define phase three.

Phase two and phase three require 3.0 microseconds of the machine cycle, and the incrementing circuit is employed throughout this interval for index register incrementing. The end of this interval is represented as T22 or T0, the beginning of the next machine cycle. Starting at this time, the contents of the program address register are transmitted to the incrementing circuit; the incremented quantity is then returned to the program address register and transmitted as part of a program store command to obtain the next program order word in sequence. The maximum time for the above steps is 1.5 microseconds, so that the program command is transmitted during 6T8 to the program store command bus. The maximum round trip returns the program word by T8 of the following cycle. This leaves a reasonable margin of 500 nanoseconds for signals to propagate through the buffer order word decoder before beginning gating actions starting at T10.

Calculations indicate that the minimum round-trip time to the program store returns the next program word before the buffer order word decoder has completed gating actions on the immediately preceding order. To circumvent a conflict on the use of the buffer order word register, an auxiliary register is placed between the program store response bus and the buffer order word register. For phase three gating actions, the operation field of the order in the buffer order word register must be retained, but not the accompanying 21-bit data field portions and 7-bit Hamming and parity portions of that program word. Accordingly, only the 28 corresponding cells of the buffer order word register are reset and connected to the program store response bus system during phase three; the auxiliary register serves to receive and retain the 16-bit operation field of the succeeding order until the buffer word decoder has completed phase three data processing steps for the current order. The contents of the auxiliary register are then transmitted to the buffer order word register during phase one of the following cycle in preparation for data processing steps beginning at phase two.

The above sample calculations are representative of those made as part of determining the realizability of the logic organization and detailed circuit design of the central processor for No. 1 ESS. These calculations served not only as a check of feasibility of proposed circuit design but also as guides in determining specifications for:

(a) tolerance limits on duplicate multiphase clocks driven from one of two crystal-controlled oscillators,

(b) maximum lengths of twisted-pair cable connecting central control to its communicating units, and

(c) tolerances between individual pairs and between buses of twisted-pair cables.

IX. DESCRIPTION OF SEQUENCER FUNCTIONS AND DESIGN

As indicated in Fig. 31, the buffer order word decoder, mixed decoder, and order word decoder serve to execute sequences of single-cycle overlap orders at the rate of one order per 5.5-microsecond machine cycle. The overlap of data processing occupies only one phase of the machine cycle, but when the generation of commands to obtain program order words is considered an additional degree of overlap is evident. The execution of certain orders (as well as hardware remedial actions such as the automatic rereading of information from the program stores and call stores) requires either the insertion of additional cycles of data processing or an extension of overlap to provide the necessary time. In providing these additional gating actions in central control a number of sequencers are provided; each sequencer carries out a specific class of data processing or remedial actions. Those sequencers which insert extra machine cycles inhibit the decoders to momentarily halt the flow of instructions obtained from the program store, and in this interval additional gating actions are carried out. Other sequencers do not insert cycles but extend the overlap by operating concurrently with the decoders; the decoders continue processing sequences of orders without regard to sequencer gating actions; in this latter instance, programming restrictions are applied to prevent conflicts in the overlapping flow of data processing. A brief description of the functions associated with some typical sequencers follows.

9.1 *Data Reading Sequencer*

All of the memory reading orders previously described (including G- and H-mode memory reading orders and control mode memory reading orders) may obtain data not only from the call stores but from any location within one of the program stores. Each of these memory reading orders generates a code and address during its indexing cycle, and the code so generated determines the memory location to be read and hence whether the data word is to be obtained from a call store or a program store. If the code and address refer to a call store location, then the central control carries out the single-cycle memory reading operation previously described. Whenever the code address refers to a program store

location, the data reading sequencer is enabled to obtain these data. Once this has been accomplished the data reading is in the same position as data obtained from the call store (i.e., placed in the data buffer register), and the data reading sequencer returns to the inactive state. The order word decoder then proceeds to complete the moving of the data from the data buffer register to the destination register specified in the memory reading order.

As the data reading obtained is returning from the program store the data reading sequencer simultaneously returns the address of the next instruction in sequence to the program address register. Consequently, as the order word decoder is completing the processing of the data reading the next order word in sequence is returned and data processing from that order begins under control of the buffer word order decoder.

9.2 *Transfer Sequencer*

Because of the degree of overlap in the central processor a transfer sequencer is enabled when a transfer is to be executed. The transfer sequencer stops the flow of instructions for at least one cycle, until: the transfer address has been placed in the program address register, transmitted to the program store, and the first program order word of the new sequence of instructions has been returned to the central control. In addition, the transfer sequencer controls gating actions required for indirect transfers, and in such instances one or two additional machine cycles are inserted, depending on whether the transfer address is to be obtained from a call store or program store location.

A class of early transfer orders is included in the order structure; the term "early" alludes to the enabling of the transfer sequencer at a time in the machine cycle earlier than that performed for regular transfer orders. The early enablement of the transfer sequencer serves to inhibit the decoders, so as not to carry out the reading or writing operation when the decision is to execute the transfer of program control.

9.3 *Program Store-Correct Reread Sequencer*

When program or data words are read from the program store, checking circuits in central control examine the 44-bit word received and the previously transmitted program address which has been retained in central control for such checking purposes. Output signals from the checking circuit indicate either:

(1) that all checks are passed and central control continues the processing of that word,

(2) a single error is detected in the 44-bit word received, or

(3) an error is detected in the address transmitted, or an even multiple error is detected, or the program store response did not include an all-seems-well signal. The program store correct-reread sequencer responds to conditions (2) and (3) to correct or to reread and recheck the program store response. The program store correct-reread sequencer increments one of two binary counters for each word corrected or reread. The counters are periodically interrogated to determine the rate at which central control is receiving program store responses containing single or multiple errors.

The failure of the program store correct-reread sequencer to succeed on a retry is designated a "reread failure condition," which requires maintenance action, since the repeated trouble condition is not assumed to be due to a transient error condition. The program store correct-reread sequencer carries out several hardware steps leading to these maintenance actions. First, a maintenance interrupt source signal is generated to switch control to the E-level maintenance interrupt program sequences. Second, the program store correct-reread sequencer stores in a special register in central control the information indicating the trouble condition associated with the rereading failure. The contents of this program store error summary register indicate whether the detected trouble condition was due to a double-error condition, an error in the transmitted address, or a failure of the program store to return its hardware check signal.

Each of the sequencers described here consists of an individual counter and gating circuits within central control; however, the actions taken by one sequencer may serve as part of the data processing steps of other sequencers in central control. For example, the program store correct-reread sequencer is activated to correct or reread program order words, but it is also responsive to data readings obtained from the program store by the data reading sequencer and to indirect transfer addresses obtained from the program store by the transfer sequencer. Whenever the data reading sequencer obtains a data word from the program store, the previously described checks are made, and when correction or rereading is required the program store correct-reread sequencer is activated. In such instances the program store correct-reread sequencer inhibits the gating actions of the data reading sequencer and prevents the counter of the data reading sequencer from advancing to the next internal state. The program store correct-reread sequencer then proceeds to carry out the required correction or rereading, and upon conclusion of its remedial actions returns to the inactive state. This last action automatically releases the data reading sequencer, which continues its handling of the

corrected or reread data word. Similar interactions occur between the program store correct-reread sequencer and the transfer sequencer.

9.4 *Accumulator Sequencer*

This sequencer provides gating actions necessary to the completion of a special class of memory reading orders. The orders are those which perform a memory reading and transmit this memory reading to the accumulator system to become one of the operands in an arithmetic or logical combining operation to be completed in the accumulator. Since the logic combining operation requires additional data processing time, the last gating action (the gating of the resulting combination into the accumulator register) cannot be completed in the time framework for the single-cycle instruction shown in Fig. 31. Accordingly, the accumulator sequencer is enabled whenever an order of this class is executed, and it completes its gating action in one additional clock phase. This sequencer differs from the previously described sequencers in that it shares control of gating actions with the decoders and extends the degree of overlap, rather than inserting additional machine cycles. Such additional time is required to process a memory reading when it is transmitted to the accumulator system, and since extended overlap rather than inserted machine cycles is used in this instance, a memory reading order which uses the accumulator as the destination register may not be followed by an order which uses the accumulator as the index register.

9.5 *Peripheral Sequencer*

The peripheral sequencer is similar to the accumulator sequencer just described in that it carries forward the data processing and instructions on an extended overlap basis rather than within inserted machine cycles. It differs from the accumulator sequencer in that its actions extend over a period of nearly two machine cycles to carry out the transmission of commands and addresses to the central pulse distributor and (as required) to all of the peripheral units such as scanners, signal distributors, network controllers, and so on. The sequencer continues to remain active in order to gate check signals and scanner responses from the addressed units, and to generate trouble signals if any of the appropriate checks fail.

The peripheral orders are used in repetitive scanning operations of lines, trunk circuits, and junctors. To achieve efficient use of real time, the maximum scanning rate of one scan for every two machine cy-

cles is provided. In such instances, the peripheral sequencer remains active during the execution of such a sequence of orders and does not return to the inactive state until (1) the scanning sequence has been completed, (2) an error in the response of the central pulse distributor or peripheral unit is detected, or (3) an interrupt intervenes in the execution of the scanning sequence.

9.6 *Sequencer Design*

A total of ten sequencers is contained in central control. The remainder of the sequencers serve in multicycle operations — for retrial of call store reading and writing operations, for special circuit actions and subsequent return from interrupted programs, for stopping and restarting data processing in the standby central control, and for obtaining special maintenance program sequences from the call store.

The ten sequencers perform different data processing steps as required in the execution of program sequences by central control. The sequencers are designed as a collection of individual counter circuits rather than as one large counter. As noted, one sequencer may "call" a second as required to carry out a class of data processing steps associated with the second sequencer. Thus the decomposition of a sequencer counter results in relatively efficient use of the number of internal states required of the counters. Perhaps more importantly, the flexibility of design gained by this decomposition has proved helpful in developing circuit specifications in the face of simultaneously evolving requirements for each of these circuits.

The heart of each sequencer is a synchronous counter. This synchronous counter responds to input signals from the decoders and check circuits and selected phases of the microsecond clock. It is enabled and advanced through various active states at definite times within a sequence of machine cycles. Although the counters are synchronous in the sense just described, the counters are designed as asynchronous counters and do not require delay lines in their feedback loops. This simplification is achieved by requiring that at least two changes of state must occur in any 5.5-microsecond interval of time (unless of course the sequencer is inactive or being inhibited by another sequencer).

An example of a sequencer counter is shown in Fig. 40 along with the time diagram showing the advancing of the sequencer through a succession of its active states.

Assuming the inactive state corresponds to both flip-flops being reset and assuming that an enable signal appears by T0 of a given machine

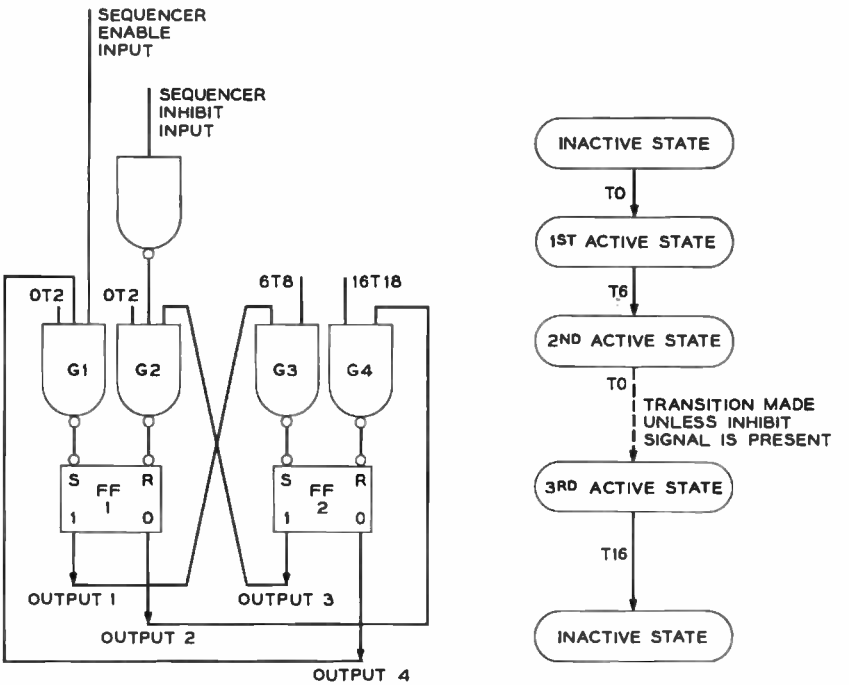


Fig. 40 — Sequencer counter.

cycle, this sequencer is enabled by activating gate G_1 at T0 to set flip-flop 1. This in turn enables output 1 so that gate G_3 is enabled at T6, and the sequencer advances from its first active state to the second active state at this time. Accordingly, by T0 of the following cycle a signal appearing on output 4 will disappear so that gate G_1 cannot be again activated in the immediately succeeding machine cycle. Instead, gate G_2 will be enabled to reset flip-flop 1 unless a signal appears on the inhibit input at this time. If an inhibit signal appears (e.g., the output of another sequencer), the sequencer will remain in the second active state until the inhibit signal disappears and a new machine cycle begins with the reappearance of the clock pulse 0T2.

The example in Fig. 40 illustrates a relatively simple counter design. Some of the more complicated counters, such as the transfer sequencer counter, include alternate sets of active states according to the type of transfer order being processed; the different sets of states serve to carry out different gating actions over different numbers of machine cycles.

The selection of clock pulses for the various counters is made to span

selected phases of various machine cycles to provide the simplest translation of counter state outputs into gating control signals. Where possible, these transitions were placed at such a time as to make sequential lockout of one sequencer to another possible; the extra connection of inhibit signals required in simultaneous lockout schemes is thereby avoided.

Although sequencers seize and release control of each of the decoders at different times (according to the function being performed), the grouping of decoder leads into four classes according to decoders and clock phase suffices for all but a few of the decoder-controlled gates. Accordingly, the outputs of the sequencers are combined to generate four sets of inhibit signals to selectively seize and release control of central control gates for all situations in which the sequencers insert additional machine cycles. The gates that could not be so grouped consist primarily of those which control the flow of program order words from the program store to central control. These had to be handled separately to provide sequencer control gating actions that performed the necessary functions with a minimum number of inserted machine cycles.

X. SUMMARY

This article has described some aspects of the logic design of the central processor, including (1) design considerations, (2) a development of the program order structure, (3) a development of the logic blocks and their interconnections to implement the order structure, (4) a description of order encoding, (5) the circuits and program orders needed to meet maintenance objectives, (6) a discussion of timing requirements, and (7) a description of sequencing circuits required for multicycle data processing functions.

Table IV summarizes the number of components in each of the functional units in one central control unit.[17] One such unit occupies four standard No. 1 ESS bays[6] and requires approximately 2800 printed wiring boards of various logic packages.

This design of the central control represents a balanced compromise between the data processing capability, economy, and reliability of operation in a telephone switching system. Furthermore, the system is able to grow through the addition of modular memory and input-output equipment without extensive wiring changes. The use of overlap operation and the inclusion of special orders to carry out several steps of highly repetitive input-output functions simultaneously assist in obtaining high data processing capability.

TABLE IV — NUMBER OF TRANSISTOR GATES IN FUNCTIONAL UNITS IN ONE CENTRAL CONTROL UNIT

Functional Unit	No. Transistor Gates	Percentage of CC Total
Program store bus circuits	260	2.1
Call store bus circuits	290	2.3
Instruction registers	390	3.1
Decoders (including memory address decoder)	1160	9.3
Error-checking and -correcting circuits and parity generators	640	5.1
Sequencers and sequencer-controlled gates	870	7.0
Index adder system	810	6.5
Program store register, increment circuit, and auxiliary storage register	720	5.8
Mask and complement circuit and insertion mask	280	2.3
Peripheral communication circuits	1140	9.2
Index registers and logic register	1140	9.2
Accumulator system, including shifting and find-rightmost-one circuit	1050	8.4
Masked bus sampling gates (C flip-flops and logic)	80	0.6
Matching circuits	1860	15.0
Emergency-action circuit	170	1.4
Clock circuits	330	2.7
Miscellaneous buffer bus registers, including interrupt sources, central processor status and error summary registers	1010	8.1
Miscellaneous circuits, including power control and maintenance scanning access	240	1.9
Total	12,440 gates	100.0%

XI. ACKNOWLEDGMENTS

Many of our colleagues contributed materially to central control organization. Mrs. E. S. Hoover and I. D. Nehama performed many of the early and fundamental systems studies which led to the present plan, A. H. Doblmaier contributed to the over-all organization of the central control, M. P. Fabisch worked out much of the encoding scheme, R. W. Downing specified most of the maintenance facilities, J. S. Nowak specified the emergency-action circuit, E. Graeve designed the clock circuits, and R. B. Smith and Miss V. R. Smith created the mnemonic representation of orders and the programmer's manual.

REFERENCES

1. Keister, W., Ketchledge, R. W., and Vaughan, H. E., No. 1 ESS: System Organization and Objectives, B.S.T.J., this issue, p. 1831.
2. Biddulph, R., Budlong, A. H., Casterline, R. C., Funk, D. L., and Goeller, L. F., Line, Trunk, Junctor, and Service Circuits for No. 1 ESS, B.S.T.J., this issue, Part 2.
3. Freimanis, L., Guercio, A. M., and May, H. F., No. 1 ESS Scanner, Signal Distributor, and Central Pulse Distributor, B.S.T.J., this issue, Part 2.

4. Danielson, D., Dunlap, K. S., and Hofmann, H. R., No. 1 ESS Switching Network Frames and Circuits, B.S.T.J., this issue, Part 2.
5. Dougherty, H. J., Raag, H., Ridinger, P. G., and Stockert, A. A., No. 1 ESS Master Control Center, B.S.T.J., this issue, Part 2.
6. Cagle, W. B., Menne, R. S., Skinner, R. S., Staehler, R. E., and Underwood, M. D., No. 1 ESS Logic Circuits and Their Application to the Design of the Central Control, B.S.T.J., this issue, p. 2055.
7. Harr, J. A., Hoover, Mrs. E. S., and Smith, R. B., Organization of the No. 1 ESS Stored Program, B.S.T.J., this issue, p. 1923.
8. Carbaugh, D. H., Drew, G. G., Ghiron, H., and Hoover, Mrs. E. S., No. 1 ESS Call Processing, B.S.T.J., this issue, Part 2.
9. Ault, C. F., Gallaher, L. E., Greenwood, T. S., and Koehler, D. C., No. 1 ESS Program Store, B.S.T.J., this issue, p. 2097.
10. Genke, R. M., Harding, P. A., and Staehler, R. E., No. 1 ESS Call Store — A 0.2-Megabit Ferrite Sheet Memory, B.S.T.J., this issue, p. 2147.
11. Ulrich, W., and Vellenzer, Mrs. H. M., Translations in No. 1 ESS, B.S.T.J., this issue, Part 2.
12. Connell, J. B., Hussey, L. W., and Ketchledge, R. W., No. 1 ESS Bus System, B.S.T.J., this issue, p. 2021.
13. Martellotto, N. A., Oehring, H., and Paull, M. C., Process III, A Compiler-Assembler for No. 1 ESS, B.S.T.J., this issue, Part 2.
14. Downing, R. W., Nowak, J. S., and Tuomenoksa, L. S., No. 1 ESS Maintenance Plan, B.S.T.J., this issue, p. 1961.
15. Tuomenoksa, L. S., and Ulrich, W., Coding and Information Identification, IEEE Trans., **67**, 1963, p. 403.
16. Butler, T. T., unpublished work.
17. Menne, R. S., unpublished work.

Organization of the No. 1 ESS Stored Program

By J. A. HARR, MRS. E. S. HOOVER and R. B. SMITH

(Manuscript received January 22, 1964)

The stored program of the No. 1 ESS must perform switching functions reliably and promptly. Its design must be economical of memory and execution time, and must accommodate office growth easily. The program is organized so that an interrupt system initiates the input-output programs that must be performed to accurate timing tolerances. The data collected by these input-output programs are passed to the call processing programs which decide what course of action the calls shall follow. The program also assigns an appropriate share of central control time for maintenance and administrative functions. The program is generic in the sense that specific quantities which change from office to office are looked up in tables and are not embedded in the program itself. This article describes the basic program structure and illustrates it with some typical programs.

I. INTRODUCTION

The program for the No. 1 electronic switching system (ESS)[1] constitutes the operating intelligence of the system. In fulfilling this function it must meet the stringent operating requirements of a telephone switching office. This article describes the most important of these requirements and outlines the implementation selected as a result.

There are six main classes of requirements. First, the system must respond appropriately in real time to demands for service. Second, the system must perform a large variety of actions to provide the many services which are offered and to work compatibly with a wide range of connecting systems. Third, the system must be extremely reliable. In forty years of continuous operation, the total time the central processor may be out of service is measured in minutes. Fourth, the system must work in wire centers which are growing in the amounts of equipment and in the scope of features offered. Fifth, although there will be a large number of installations which may differ from each other in the kind

and quantities of equipment and in the services which they offer, costs of compiling programs must be kept at a minimum. Finally, the program should be designed to keep system costs low whenever possible, principally the cost of program storage, the cost of call storage, and the cost of the required number of central processors needed for the total market.

1.1 *Requirements on Processing Capability*

Telephone customers are used to receiving prompt service whenever they request it. As a result, much of the call processing work performed by a switching system cannot be postponed for long. For example, when a customer answers a ringing telephone, ringing stops in less than a quarter of a second and the two customers may begin conversation.

A particular wire center also receives signals from other wire centers as well as from customers directly. The size and extent of the total investment in switching equipment makes it necessary for new equipment to communicate with older systems. For example, because step-by-step switching trains are driven by the customer's own dial pulses, the step-by-step office will spill digits into another office without checking to see if the receiving office is ready. Therefore, if calls are to be processed correctly, the No. 1 ESS must be ready to receive dial pulses from a step-by-step office in a few hundredths of a second from the time the trunk is seized.

A third source of demand for prompt action comes from the character of the switching system which the program must control. The relay circuitry has been much simplified compared to that in previous common control systems. However, relays are still used both in the network controllers and in trunk, junctor, and service circuits. Simplification has been possible because the program has taken on the job of triggering the operation of relays in proper timing and sequence. In the case of the dial pulse receiver, for example, pulse detection, counting, and memory functions are performed by the program. In order to make sure that the pulses are detected under the worst cases of pulse distortion, the program must look every 11.5 milliseconds for a change in the line current. The higher the repetition rate of scanning, the larger the percentage of time which the central processor must spend in scanning. Because the duration of the scanning program itself varies with the number of pulses detected, there is variation in the actual rate of scanning a given receiver. Study has shown that when a group of receivers is scheduled to be examined every 10 milliseconds, the interval between inspections of any particular receiver will almost never exceed

11.5 milliseconds. Thus, certain functions are performed by the program on fairly tight timing tolerances in order to simplify circuitry in the rest of the switching system. The need to perform a number of tasks with different tolerances is inherent in the nature of the switching function. Failure to organize the program properly would result in inefficient use of the central processor, which in turn would mean a larger investment in data processing equipment.

1.2 *Versatility*

A fundamental reason for using a stored program in the No. 1 ESS is the need for versatility. The No. 1 ESS must be compatible with other switching equipment and must provide the ever growing list of special services which the telephone industry offers to its customers. By using a stored program, these objectives can be attained economically. Even though much development work may go into designing programs for new features, the program, once written, can be easily installed. Only the contents of memory must be changed and not a large number of wired connections. Because the program must perform so many functions, it is very large. To keep it manageable, it is divided into functional blocks which are utilized in different ways to perform a large number of complicated tasks. When a new service is needed, much of the required program may be already available. The manner of dividing the program is discussed in the paper on call processing.[2]

1.3 *Reliability*

The ability to process calls is the principal function of the central processor and its program. Of almost equal importance is the ability to do so reliably. A complete failure of a central office for even a period of 15 minutes is an event so rare that it generally is reported in newspapers. Yet the No. 1 ESS has so centralized the intelligence of the central office that no call can proceed without the attention of the central processor. Therefore, when a failure or malfunction is detected in the central processor, the faulty unit must be switched out and a working system put together from the duplicated units. The faulty unit must then be diagnosed so that plant personnel may quickly repair it and put it back into service.

About half of the total program instructions are devoted to fault recognition, diagnosis, and routine maintenance. The detection of a fault, the analysis of which unit is faulty, and the actions taken to assemble a working central processor are assigned the highest levels of

priority. Once a working system has been put together, the detailed diagnosis of the faulty unit can proceed at a more leisurely pace than that of normal call processing.

Included within the call processing programs are checks for abnormal inputs and processing errors. For example, a customer ought not to be able to generate more than ten dial pulses before there is a pause while the dial is wound up again. If, however, there is a weak power cross which has been undetected by the power cross test, a longer series of dial pulses may be generated. The input program which counts pulses checks for an excessive count and treats such a call as a partial dial.

1.4 *Ability to Accommodate Growth*

A given wire center will have a unique set of engineered equipment. With growth, the amount of equipment changes. It would be expensive to rewrite the program every time an additional frame of equipment is added to the center. Therefore the basic program is designed to treat all information about quantities of office equipment as data which can be changed as the office grows.

1.5 *Standardization of the Program*

Additional flexibility is required of the program to meet the needs of wire centers with differing features. One wire center needs to communicate with step-by-step switching equipment, another with panel switching equipment, a third offers TOUCH-TONE calling service to its customers, and so on. Producing a tailor-made program for each office would increase programming and compiling costs and entail problems in guaranteeing an error-free program. Therefore, a further requirement is that a single program contain all features needed for a wide range of applications. This standard program, called a *generic* program, may thus be used in many installations.

1.6 *Economic Balance*

The length of the program in program store, the number of call store words, and the number of machine cycles required to perform a task can all be traded off against each other. These quantities have associated costs; for example, the number of machine cycles affects the total amount of processing equipment required to serve a given market. Therefore an over-all requirement is achievement of a sound economic balance in the use of these quantities.

The following pages describe the organization and implementation of the program plan chosen to meet these requirements.

II. PROGRAM STRUCTURE

The organization of the No. 1 ESS program is strongly influenced by the fact that it must operate in real time. That is, the program must respond promptly to signals and data submitted to it by customers and other switching systems. In addition, it must respond quickly to errors detected by the many trouble detector circuits designed into the hardware to assure dependable operations within the system at all times. Whenever it fails to do so, the result may be improper handling of calls and a general degradation of service. For example, failure to detect digital signals may result in directing a call to a wrong number, or failure to outpulse digits to another office promptly will cause the other office to return overflow tone to the calling customer. Therefore it is necessary to establish a hierarchy of program tasks. Some tasks must be performed on a strict schedule; others may be delayed without significant adverse effects.

2.1 *Interrupt System*

The central processor has an interrupt mechanism[3] within it which seizes control of the system momentarily when a manual, trouble detector, or clock signal is received. The interrupt circuit causes the system to stop its present program task, store the program address at which it was interrupted, and then transfer to the appropriate fault-recognition program or clock-controlled input-output program. When the interrupt programs are completed, control is returned to the program that was interrupted.

Fig. 1 illustrates this over-all program plan. The interrupt sources and their associated programs are arranged in a hierarchy of nine interrupt levels; from highest to lowest, these levels are designated A, B, C, D, E, F, G, H, J. An interrupt source assigned to a particular level can interrupt programs of lower levels only. The majority of the programs are subject to interruption by any of the nine levels, and are therefore called *base-level* programs.

The highest interrupt source is the A level, initiated from the master control center, which allows manual selection of operating modes. Interrupt levels B through G are activated by system trouble detectors. These fault-recognition programs are discussed in the article describing maintenance[4] appearing in this issue.

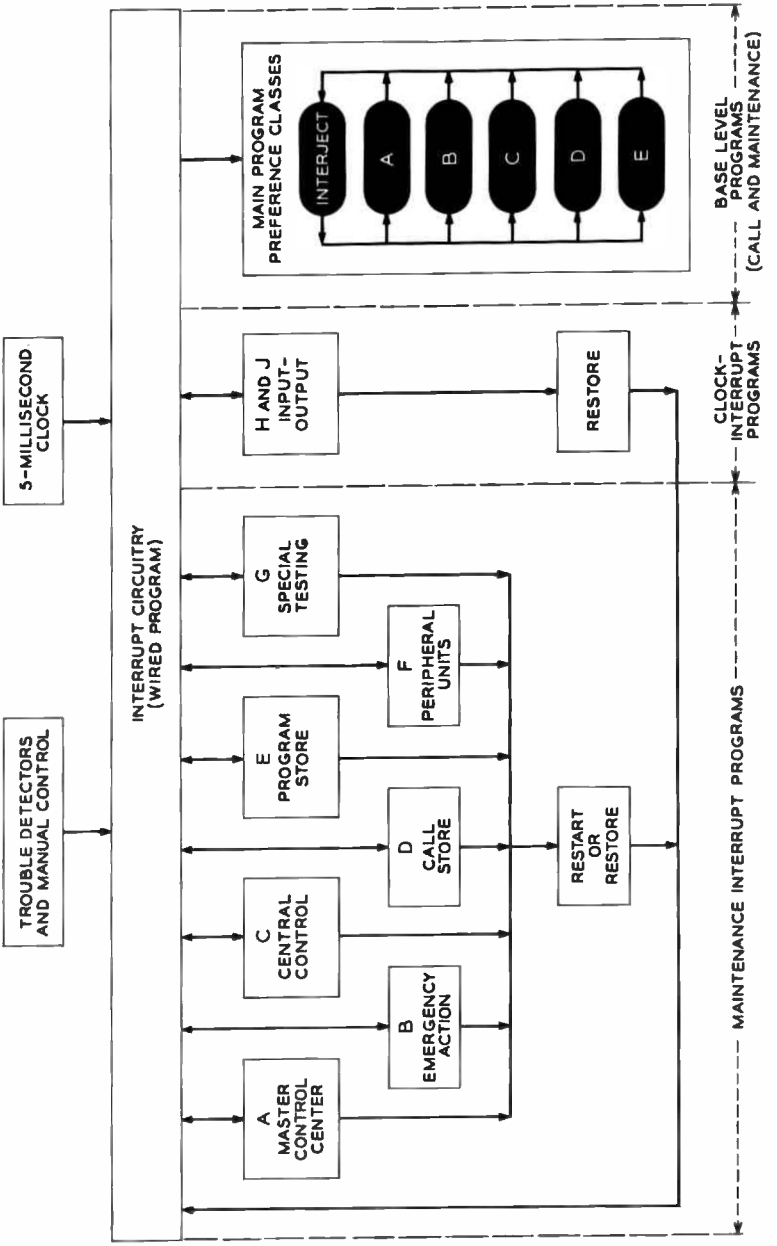


Fig. 1 — Program control plan.

Every 5 milliseconds a system clock activates interrupt level J, which in turn gives control to input-output programs. Level H is used to interrupt the level-J input-output program when tasks being performed exceed 5 milliseconds. The level-J programs are normally in control for about 0.2 to 2 milliseconds, according to the size and traffic of the office.

2.2 *Input-Output Buffering*

In order to go to the next task reasonably promptly, the individual tasks must not take too long. In particular, it is necessary to limit the amount of processing performed by interrupt-level programs. For example, the scan of the lead on a TOUCH-TONE receiver that indicates that a digit is present ends by reading the ferrods associated with the individual frequencies and placing the result in an area of call store known as a "hopper." It would be possible to carry the processing of the call further. The area in call store where the other digits dialed on this call are stored could be interrogated to determine whether dialing is finished, and if so, the network path hunt and ensuing actions required to set up ringing could be carried out. However, all this work would consume several milliseconds of continuous processing for a single call. Since the scan for a signal present on a TOUCH-TONE receiver must be performed every 10 milliseconds, an excursion of several milliseconds each time a digit is detected would frequently exceed tolerances. Hence, work on the call is terminated by buffering the digit in a hopper, from which a base-level program will later unload it and carry out the processing just described.

Fig. 2 is a schematic representation of the program control and flow which carries out the sequence of actions needed to process calls, described elsewhere in this issue.[2]

The input-output programs are shown in the upper part of this figure. The input programs are confined to scanning for and recognizing input signals and storing this information in a call store hopper. The hopper stores the input information until the base-level programs inspect it for data. When data are present, appropriate base-level programs, as shown at the bottom of Fig. 2, start or continue the processing of the call. Likewise, call store buffers are provided for the base-level programs to load with output data. At an appropriate time, these data are unloaded by an output program which delivers them to the peripheral equipment. Some examples of the hoppers and buffers used in the system are shown in the middle of Fig. 2. For example, the peripheral order buffer (POB) is used to store address and control data for network controllers and

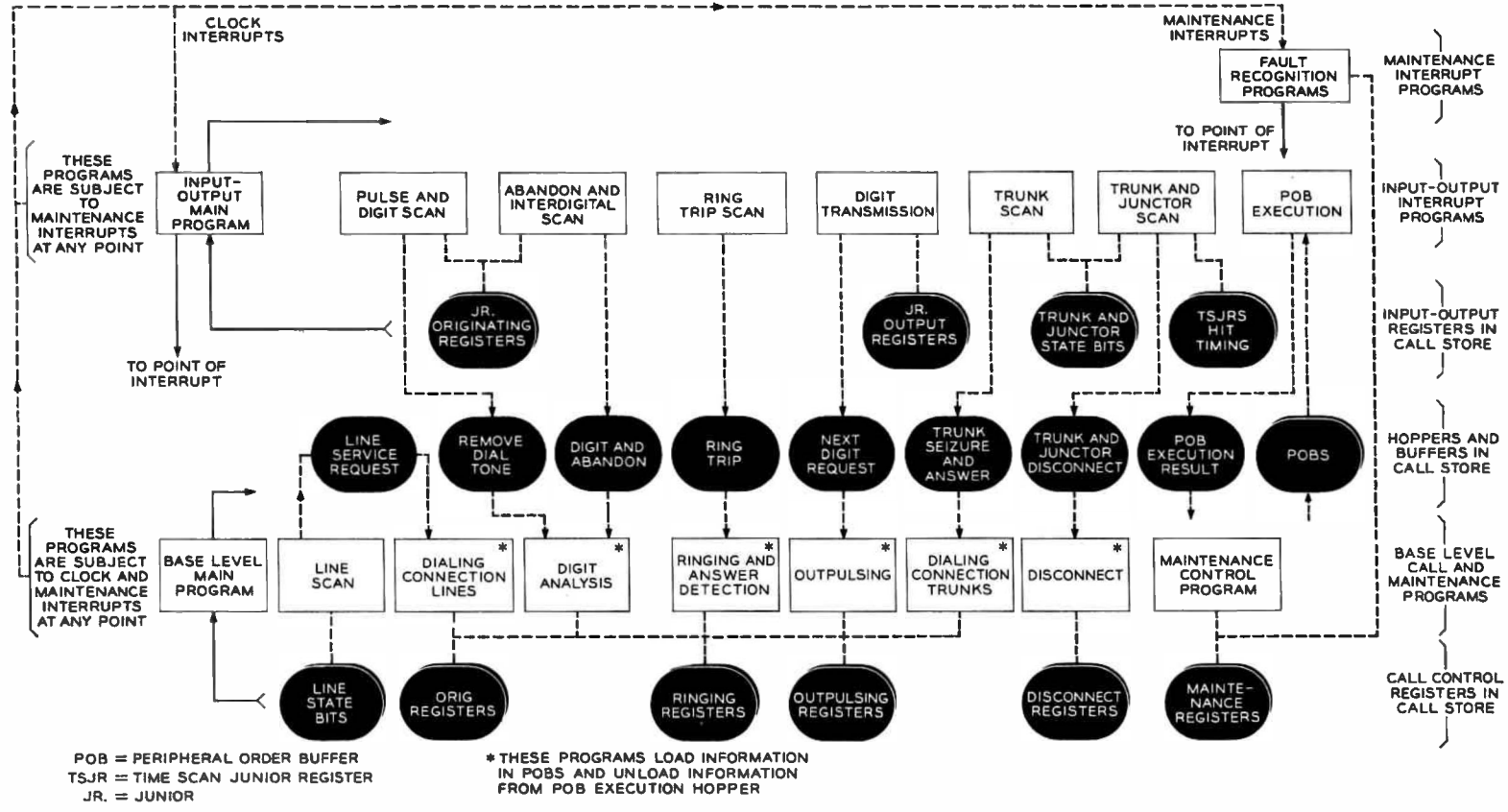


Fig. 2 — Schematic representation of program control and flow.

signal distributors. These buffers provide the means for communication between the scheduled input-output programs and the base-level call processing programs.

III. PROGRAMS PERFORMED ON THE CLOCK INTERRUPT LEVEL

The 5.5-microsecond clock pulses in the central control are counted, and every 5 (actually 5.005) milliseconds the counting circuit generates an output signal which interrupts the base-level program being performed. The interrupt circuit makes the central processor transfer to the J-level input-output main program. Some input-output tasks are performed every 5 milliseconds; others are performed at multiples of 5 milliseconds up to 120 milliseconds. The input-output programs are classified into high-priority and low-priority tasks, according to the frequency and accuracy with which they must be performed.

The low-priority tasks can be delayed for a few milliseconds without adverse effect on the operation of the system. This indeed will be the case when the coincidence of input work under a peak traffic load causes the system to take more than 5 milliseconds to complete the high- and low-priority tasks. In this event the H-level interrupt will occur and the low-priority work will be interrupted. The accumulated high-priority work will again be performed before returning to the low-priority program that was interrupted.

Accordingly, each input-output program is assigned to either the high- or low-priority timetable. A list of high-priority tasks, the frequency at which they must be executed, and the call store memory words needed for carrying out these tasks are shown in Table I. A similar list of low-priority tasks is shown in Table II.

3.1 *Input-Output Main Program*

To assist in understanding the design of the input-output main program, a description of some additional program attributes is needed.

TABLE I — HIGH-PRIORITY INPUT-OUTPUT TASKS

Task	Frequency	Call Store Memory
(1) Dial pulse and digit scan	10 ms	junior originating registers, remove dial tone and digit hoppers
(2) Abandon and inter-digital timeout	120 ms	junior originating registers, digit and permanent signal partial dial hoppers
(3) Twistor card writing	5 ms	twistor word storage register
(4) Call charge magnetic tape output	5 ms	message storage registers

TABLE II — LOW-PRIORITY INPUT-OUTPUT TASKS

Task	Frequency	Call Store Memory
(1) Teletypewriter scan	25 ms	teletypewriter buffer
(2) Peripheral order	25 ms	peripheral order buffer (POB), POB execution hopper
(3) Power cross test scan	*	POB execution hopper
(4) Ringing current test scan	†	POB execution hopper
(5) Detection of outgoing trunk wink	100 ms	outpulsing junior register, next-digit request hopper
(6) Multifrequency outpulsing	25 ms	outpulsing junior register, next-digit request hopper
(7) Disconnect scans	10 ms	timed-scan junior registers, trunk and junctor disconnect hopper
(8) Interrupt sanity test	100 ms	emergency-action register
(9) Interject timing	100 ms	interjected ordered bits buffer

* This task is performed on three consecutive 5-ms intervals after a power cross scan order is encountered during a POB execution.

† This task is performed on the first and third of the five 5-ms intervals after a ringing current test scan order is encountered during a POB execution.

3.1.1 *Characteristics of the Input-Output Main Program*

Since this program must be executed every 5 milliseconds, the time required by the central control to cycle through all of the input-output task programs is held to a minimum, even at the expense of a small increase in the total number of program words. For example, it is expedient in some cases to have a number of program blocks that perform nearly equal tasks instead of a common program capable of performing all of the tasks, since the common program would in general involve more machine operations to accommodate the small variations in each of the individual tasks.

The program plan is sufficiently flexible that the same program can provide service after changes in the system due to growth. Also, the same program must operate during and after certain changes in the features offered by an office. For example, a feature included in the generic program may require a type of trunk circuit which is not provided in all offices. These circuits may be introduced into any office without changing the program. To meet this growth requirement, the information relating to size, traffic, and features for an office is not embedded in the data field of program instructions, but instead is provided in parameter tables within the program store. In cases where the real time of the system is quite sensitive to the access time for retrieving a particular parameter, it is also stored in the call store, from which it can be read more quickly than from the program store. In particular,

two parameters of this type are the number of input-output tasks and the frequency with which they are performed.

3.1.2 *Organization of the Input-Output Main Program*

A block diagram of the input-output main program is shown in Fig. 3. When the J-level interrupt occurs, control is transferred to the input-output main program, which operates as follows:

(1) It saves the contents of the central control index registers to allow resumption of the base-level program at the point of interruption and sets the H-level inhibit flip-flop.

(2) It updates the time counter and activates, one at a time, all input-output programs that require action according to the high-

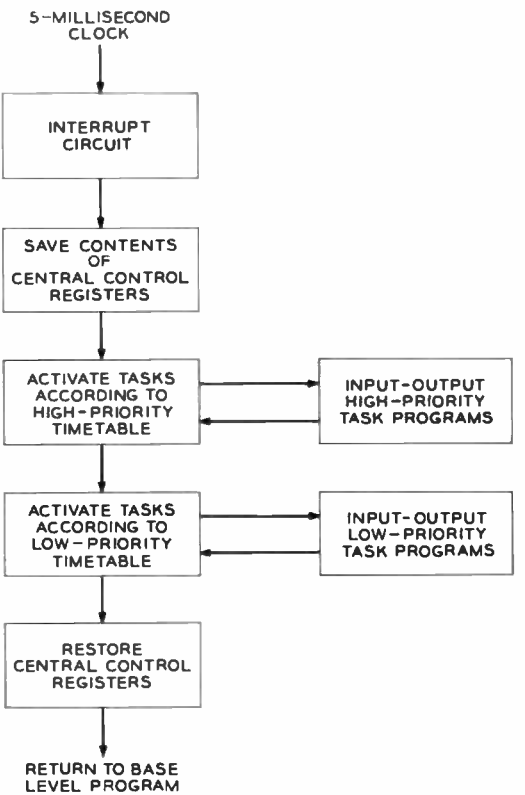


Fig. 3 — Input-output main program.

priority timetable. When all high-priority tasks are completed, it uninhibits the H-level interrupt by resetting the control flip-flop.

(3) It activates, one at a time, all input-output programs that require action according to the low-priority timetable.

(4) It then refills the central control index registers with the information saved in (1) and returns control to the interrupted base-level program.

The principal parts of the input-output main program are the high- and low-priority timetable programs. Since both programs are identical in structure, it is necessary to examine only one of them to understand the operation of the input-output main program.

3.1.3 *High-Priority Timetable Program*

Figs. 4 and 5 show the layout of the call store used to support the high-priority timetable program. The transfer table of Fig. 4 consists of 23 consecutive words. Here and in the following descriptions, symbolic designations will be used — in this case, P0 to P22 — instead of

P	0	DIAL PULSE AND DIGIT SCAN PROGRAM ADDRESS 1
	1	DIAL PULSE AND DIGIT SCAN PROGRAM ADDRESS 2
	2	
	3	
	4	
	5	
	6	ABANDON INTERDIGITAL TIMING SCAN PROGRAM ADDRESS
	7	
	8	
	9	CALL CHARGE PROGRAM ADDRESS
	10	
	11	
	12	
	13	
	14	
	15	TWISTOR CARD WRITING PROGRAM ADDRESS
	16	
	17	
	18	
	19	
	20	
	21	
P	22	ZERO CNT PROGRAM ADDRESS

Fig. 4 — Call store transfer table for high-priority timetable program.

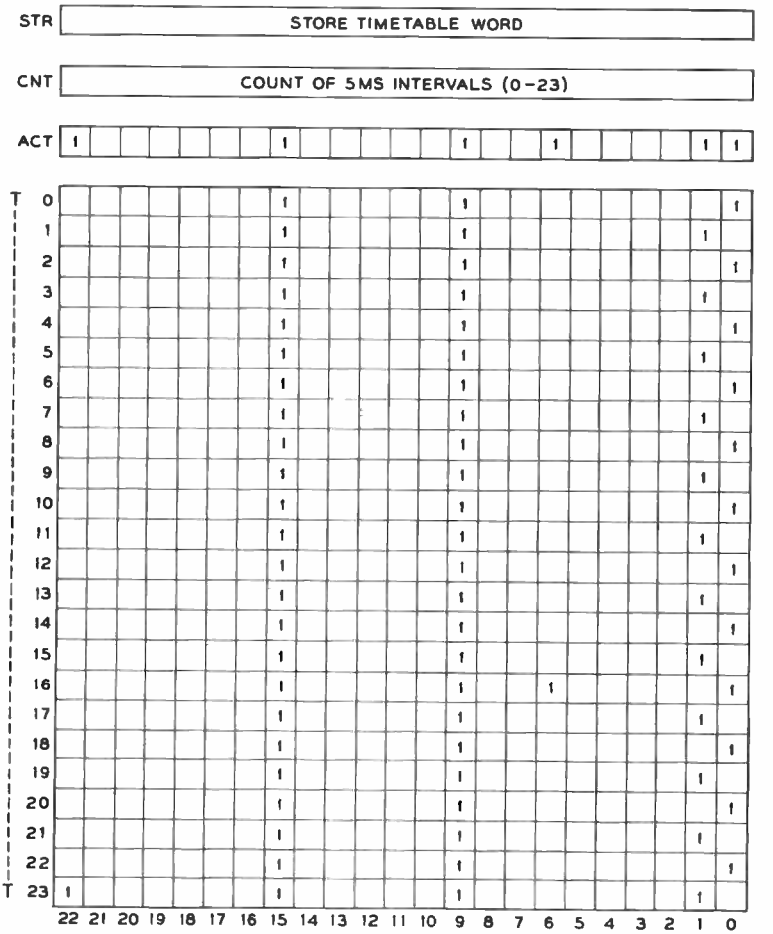


Fig. 5 — Call store timetable for high-priority timetable program.

numerical call store addresses. Each of the 23 words is used to store the starting address of an input-output program assigned to the high-priority timetable. For instance, words P0 and P1 contain the program store addresses of the dial pulse and digit scan programs.

The timetable of Fig. 5 consists of 24 consecutive words designated T0 to T23. Each word is associated with a particular 5-millisecond interval within a 120-millisecond cycle. A count kept in the CNT word is used to identify the 5-millisecond interval and the associated T word in the timetable.

Within each T word, bits 0 to 22 are associated with the input-output

programs whose starting addresses are stored in the corresponding words P0 through P22. Thus each column of the timetable is associated with an input-output program. Each column has an activity bit which is part of the ACT word. If the activity bit of a column is equal to 1, the 1's marked within the column designate the 5-millisecond intervals during which the associated input-output program is due for execution. The table shows that program P0 is activated on all even 5-millisecond intervals and that program P1 is activated on all odd 5-millisecond intervals. Program P15, when active, is due every 5 milliseconds. Program P22 is activated only once every 120 milliseconds. This program recycles the count in CNT from 23 to 0.

The word STR is used to store the pattern of 1's and 0's that indicate which input-output programs are due for execution during an interval. As these programs are executed, one at a time, the corresponding flag bits are reset to 0 until all the programs have been executed.

A step-by-step description of the program shown in Fig. 6 is given in Table III. Assume that the count of 5-millisecond intervals that is kept in CNT is initially 0.

As shown in Fig. 6, to activate the first input-output program requires eight program steps and subsequent ones only five. Note that only six of the possible 23 flag columns are presently used in this high-

LOCATION	OPERATION	ADDRESS OR DATA, REGISTER AND OPTION FIELDS
ENTER	MX	CNT
	MK	TO, XA
	XM	CNT
	PMK	ACT
	TZRFZ	LP
	KM	STR
	MY	PO, F
	T	O, Y → TO INPUT-OUTPUT TASK PROGRAM
LOOP	MK	STR ← FROM COMPLETED TASK
	TZRFZ	LP → TO LOW PRIORITY TIMETABLE PROGRAM
	KM	STR
	MY	PO, F
	T	O, Y → TO INPUT-OUTPUT TASK PROGRAM

Fig. 6 — High-priority timetable program.

TABLE III — STEP-BY-STEP DESCRIPTION OF HIGH-PRIORITY TIMETABLE PROGRAM

(1)	MX	CNT	The content of CNT is read into central control (CC) register X. (This value of the interval count is used as a pointer to select one of the 24 timetable entries.)
(2)	MK	T0,XA	The address T0 is indexed with the value of the X register to obtain the address of the timetable entry to be read into CC register K. Then the content of X is increased by 1. (In successive interrupts, this instruction results in reading the timetable entry at address T0+0, T0+1, T0+2, and so on up to T0+23.)
(3)	XM	CNT	The new value of the interval count is stored in CNT.
(4)	PMK	ACT	The timetable entry contained in K is ANDed with the activity bits in ACT; the resulting word is placed in register K. This word contains a 1 in every position in which both the timetable entry and the ACT word contain a 1. Thus, each position marked by a 1 designates a program that is active and due for execution. For instance, when T0 is read, three bits are equal to 1 in columns 0, 9, and 15.
(5)	TZRFZ	LP	If all the bits in K are 0 (if no programs need to be acted on), the program transfers to address LP where the low-priority timetable program starts. If one or more bits of K are equal to 1, the position of the rightmost 1 is stored in register F. The rightmost 1 itself is set to 0, and the program advances to the next step. Clearing the rightmost 1 allows the next 1, if any, to be recognized later as the new rightmost bit of the word in K.
(6)	KM	STR	The new binary word in register K, modified by the removal of the rightmost 1, is stored in STR.
(7)	MY	P0,F	The address P0 is indexed with the value in register F to obtain the transfer table entry to be read into register Y. This is the address of the input-output program corresponding to the rightmost 1, originally in register K.
(8)	T	0,Y	The program transfers unconditionally to the address stored in register Y as the result of step (7). When the input-output program has been completed, a transfer is made back to step (9).
(9)	MK	STR	The word stored in STR is read into register K.
(10)	TZRFZ	LP	If all the bits of the word in K are 0, the program transfers to the low-priority timetable program. If one or more bits are equal to 1, the position of the new rightmost 1 is stored in register F, that 1 is erased, and the program advances to the next step.
(11)	KM	STR	This step performs the same function as step (6); it stores any remaining flag bits in STR.
(12)	MY	P0,F	This step performs the same function as step (7); it obtains the address of the next input-output program to be executed and stores it in Y.
(13)	T	0,Y	A transfer is made to the address in register Y. When the input-output program has been completed, a transfer is made to step (9). Steps (9) to (13) are used over and over until, one by one, the flag bits are removed. The TZRFZ instruction of step (10) will then lead to the low-priority timetable program.

priority timetable. Therefore ample space for future input-output programs is provided. All that has to be done to add programs is to place appropriate time flag bits in any one of the unused columns, mark the column activity bit to 1, and place the corresponding entry address of the new input-output program in the transfer table.

This flexibility has been gained at the cost of only 17 additional call store words in the transfer table. By writing appropriate data into the call store (and a copy in the program store for reliability), an input-output program included within the generic program can be added to an office, and changes can be made in the order and frequency of execution of any input-output program. Furthermore, the input-output main program need not be changed even if a new issue of a generic program requires the addition of new programs.

An example of a specific input-output high-priority task program is given in Section 3.2.

3.2 *Dial Pulse and Digit Scan Program*

This program is used to scan the signal-present leads of dial pulse, multifrequency, and TOUCH-TONE receivers for signals every 10 milliseconds. It is activated during every 5-millisecond interval by the high-priority timetable program. However, only half the receivers are scanned in each of the 5-millisecond intervals in order to even out the work load. The functions of this program are as follows:

(1) For dial pulse receivers this program detects pulses by observing a change in the scanner reading from 0 to 1 (off-hook to on-hook condition of the handset). When such a change is found, the program adds one to the pulse count, which is located in a call store area called a "junior register" associated with each receiver. If this is the first pulse received, the address of the originating register in which digits are to be accumulated for this call is read out of the junior register and loaded into the remove dial tone hopper. In case the pulse count overflows (becomes equal to 16) the program stops incrementing the pulse count so that the permanent signal partial dial program will detect a timeout and make an entry in the permanent signal partial dial hopper.

(2) For TOUCH-TONE receivers this program detects that TOUCH-TONE signals are present by observing a change in the signal-present scan point reading from 0 to 1. Upon finding a change, the tone frequency scan points are read and their values together with the address of the originating register serving the call are stored in a TOUCH-TONE digit hopper.

(3) For multifrequency receivers this program detects the presence of multifrequency signals by observing a change in the signal-present scan point value from 0 to 1. The scanner leads associated with each frequency are read, and their values together with the address of the originating register serving the call are stored in the multifrequency digit hopper.

Observe the similarity of the program actions required to detect and report inputs from the three types of receivers. This design permits flexibility in the assignment of scan points for all three types of receiver, since the input program scans a row of scanner points for a change in reading from 0 to 1 without regard to the type of receiver being interrogated.

It is beyond the scope of this paper to discuss the complete dial pulse and digit scan program. However, a description of the core of this program reveals its basic design.

Fig. 7 shows the call store memory layout used by the core program. Each word in the scanner address table (SCA) contains a trunk scanner number of a row of digit receivers. This number addresses a word of 16 scanner outputs, among which at least one is assigned to a receiver.

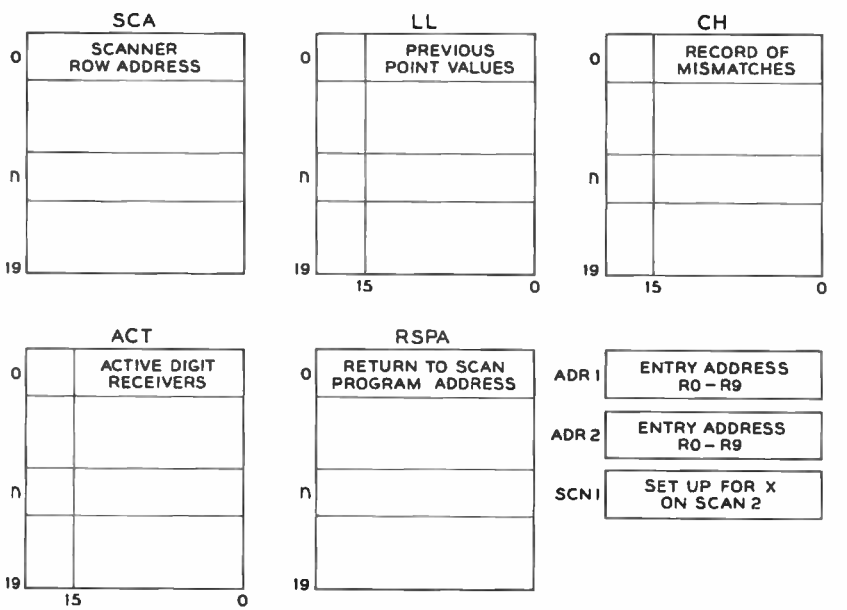


Fig. 7 — Call store memory layout for core of scan program.

The last look table (LL) stores the previous scanner outputs of the row in a memory location corresponding to the location of the row address in the scanner address table.

The change status table (CH) is used to record a mismatch between the present and previous scanner readings when detected. The corresponding change status bit for the scanner is set equal to 1. The abandon and interdigital timing scan uses this information to determine interdigital timeout or abandonment of the call by a customer.

The activity table (ACT) specifies the active receivers within each scanner row and is used by the dial pulse or digit present program to determine the active receivers reporting a change in state from 0 to 1.

The return to scan program address table (RSPA) contains the appropriate return address for use by the dial pulse or digit present program to return control to the core program after the pulse counter(s) is (are) incremented and/or the appropriate hopper(s) is (are) loaded.

Observe that all of the tables described above are blocks of 20 words each. This design was made to accommodate the maximum number of receivers required in a wide range of offices. The symbolic name of the call store address of the first word of each table is SCA, LL, CH, ACT or RSPA, respectively. The binary addresses corresponding to these names will be defined by the compiler-assembler[5] in the fixed area of the call store. That is, these tables have been assigned a fixed location in call store in all central office programs. However, for small offices only a small number of rows in each table will be used to accommodate the receivers for the office. Therefore, the core program is designed to scan only the number of rows required in a given office. Since approximately half the number of rows are to be scanned in each of the 5-millisecond intervals, three other control words are needed by the core program.

At a call store location with the symbolic name ADR1 is stored one of the entry addresses R0 through R9 according to the number of rows to be scanned during the even 5-millisecond intervals. Likewise, at address ADR2 is stored the appropriate entry address for the number of rows scanned during the odd 5-millisecond intervals. At address SCN1 is stored a number one less than the number of rows scanned in the even interval. This number is used in the odd interval scans to set an index register to the proper value for controlling the row words read and stored in memory.

The functions performed by the core program shown in Table IV are:

(1) Initialize central control registers according to whether the scan occurs in an even or odd interval.

(2) Read the row of 16 scan points represented by the first scanner

TABLE IV — DIAL PULSE AND DIGIT SCAN CORE PROGRAM

Location	Operation	Variable Fields	Comments
SCAN1	WX	−1	This is the entry point on even 5-ms intervals. Set index register X to −1.
	MSF	SCA+1,X	Send scanner address in first word of SCA table to scanners. Present state of 16 scan points returns to central control register L.
	WZ	DPDP	Store in central control register Z the address (DPDP) of entry point to dial pulse or digit present program.
	MK	CH+1,X	Move content of first CH word into central control register K.
	T	ADR1,M	Transfer program control to the address in call store location ADR1; this will be one of the addresses R0 to R9, depending on the number of rows to be scanned in this office.
SCAN2	MX	SCN1	This is the entry point on odd 5-ms intervals. Set index register to one less the number of rows scanned in the even interval.
	MSF	SCA+1,X	(Same as SCAN1+1)
	WZ	DPDP	(Same as SCAN1+2)
	MK	CH+1,X	(Same as SCAN1+3)
	T	ADR2,M	(Same as SCAN1+4, except that call store address ADR2 is used instead of ADR1.)
R0	UMKMJ	LL+1,X	R0 is the entry point from SCAN1+4 or SCAN2+4 if ten rows are to be scanned. Match (exclusive-OR) present scan point values in register L with previous scan point values read from corresponding word in LL table. Store in central control register J the 16-bit mismatch word (1's at bit positions of mismatch), and OR it with the CH word in register K, storing the result in K.
	LM	LL+1,X	Move present scan point values from register L into the LL table.
	KM	CH+1,XA	Move into table CH from K the new CH word, which now contains 1's in bit positions corresponding to scan points whose state changed since the last abandon-interdigital scan. Also add one to register X.
	JKMSF	SCA+1,X,PL	AND the mismatch word in J with the present scan point word in L and store in K this logical product, which now contains 1's only in the bit positions where a 0-to-1 change occurred in the corresponding scan points. Set a decision-control flip-flop, to be used by the following order, according to whether the product in K is zero or not. Also read the next scanner address from the SCA table and send it to the scanners. The scan point values will return to L in two machine cycles.
	TAUMK	CH+1,X	If the product in K is nonzero, transfer program control to the address in central control register Z. If zero, move into K the contents of the next word in CH table.

The next 45 steps essentially repeat the preceding five instructions nine more times to take care of a possible ten rows of scan points to be examined. The last few instructions in the tenth set of five are modified slightly because it is not necessary to prepare for a next row in that case. A concluding instruction transfers control back to address LOOP in the high-priority timetable program (Fig. 6).

row address in the SCA table if it lies in the even interval, or scanner row address one greater than the number stored in SCN1 if it lies in the odd interval.

(3) Match the 16 new scan point values with the previous scan point values stored in the corresponding row of 16 LL bits.

(4) Update the corresponding row of 16 CH bits. That is, write 1's into the CH word wherever a mismatch (change in scan point state) occurred.

(5) Update the row of LL bits. That is, read present scan point values to the corresponding row in the LL table.

(6) Determine those scan points whose value changed from 0 to 1.

(7) Transfer to the dial pulse or digit-present program if there is at least one change from 0 to 1.

(8) Repeat steps (2) through (7) for the remaining scanner rows to be examined during this 5-millisecond interval.

(9) Return program control to the high-priority timetable program.

Because of the frequency of use of this core program, an economic balance clearly required an emphasis on machine cycle minimization at some cost in total memory. Two means were used to achieve this real time efficiency. First, three special instructions (UMKMJ, JKMSF, and TAUMK) were designed, each of which accomplishes functions that would require two or three general instructions. Second, the five-word core program is repeated ten times in the program store to avoid the additional time for executing loop control orders and transferring. The program is generic, in that it is designed to handle traffic over the range of office sizes without appreciable loss of efficiency. This flexibility is attained in exchange for a moderate increase in the total call store and program store words used.

IV. BASE-LEVEL PROGRAMS

The bulk of No. 1 ESS programs, both call processing and maintenance, are executed on the base level — that is, with none of the interrupts A through J in effect. An appreciable amount of time is spent in the J level, as described above, looking for inputs and disposing of outputs, but this time is consumed in repeated execution of a relatively small number of fairly short programs. The complex work called for by the enormous variety of call situations and equipment configurations and possible malfunctions is carried out by a correspondingly large number of programs. These are executed as required, when time permits, and while no interrupts are in effect. All base-level programs

can be deferred to some extent, but the amount of delay they can tolerate varies widely. It is for this reason that a preference system and a program organization to implement it are required within the base level.

4.1 *Main Program*

This plan and its associated programs are referred to as the base-level main program, or simply the "main program." The question of precisely which associated programs should be considered a part of the main program is a rather arbitrary matter of classification. The fact is that the entire collection of base-level programs becomes in a real sense a *single* program, though subdivisions are useful for various purposes such as functional design and explanation, assignment of work to individual programmers, convenience of assembly and testing, and so forth.

Maintenance programs on the base level, as well as call processing programs, receive control from the same main program. They follow the same general rules and share many devices and techniques that are discussed below. However, since the structure of the maintenance programs is described in detail elsewhere,[4] details and examples in this paper are drawn primarily from the processing of telephone traffic.

Just as the bulk of No. 1 ESS programs are base-level, so the bulk of base-level programs are "task" programs. These are simply the programs at the end of the line of control that perform the ultimate work of the office. Most do a particular kind of work on a single call at a time; others perform various administrative functions, such as interpreting a teletypewriter input message. They differ, too, in the manner in which they receive control from the main program. Some receive it directly from a single dispenser program, while others require a series of dispensers, each triggering the next.

Within the main program complex are distinguished several kinds of "dispenser" programs. These form the links between the basic schedule of the main program and the task programs. One of the principal jobs of the dispenser programs is unpacking the input data that the H and J interrupt-level programs have buffered and distributing them to the task programs for analysis and use. Another important dispenser function is providing timed entries to programs requiring them.

All base-level work is divided into six classes. The highest priority class, called "interject," is described below. The other five are the classes A, B, C, D, and E, in descending order of frequency of examina-

tion. Specifically, the main program delivers control to these classes according to the pattern

ABACABADABACABAEABACABADABACAB

repeated endlessly. Thus class A is examined most frequently, and tasks assigned to that class will suffer shorter delays on the average than those in B, and so on. But if, for example, class E work is about to be done, the existence of waiting class A work cannot change the scheduled order of execution.

Each class consists simply of a number of dispenser programs that are executed in a fixed order, so that if $a_1, a_2, \cdots, a_5$ represent the five dispenser programs in class A, $b_1, b_2, \cdots, b_5$ the five in class B, and $c_1, c_2, \cdots, c_6$ the six in class C, the sequence of execution stated above can be expanded to:

$$a_1a_2a_3a_4a_5b_1b_2b_3b_4b_5a_1a_2a_3a_4a_5c_1c_2c_3c_4c_5c_6a_1a_2 \cdots.$$

4.2 *Dispenser Programs*

As an illustration, class B consists of the dispenser programs that administer work from the following five buffers:

class B ordered bits buffer
ring trip hopper
remove dial tone hopper
dial pulse and abandon hopper
trunk and junctor disconnect hopper.

"Buffer" is used as the general term for any memory used to store or record work to be carried further at a later time; a "hopper" is a buffer used to accumulate inputs from peripheral equipment.

The ring trip and remove dial tone hoppers are typical single-purpose hoppers. The nature of the input, in each case, is known automatically by the interrupt-level program that detects it, so it is just as easy for it to be loaded into its own hopper, and more efficient for the unloading dispenser program. The remove dial tone hopper is shown in Fig. 8. The "pointer block" of four words contains both a loading and an unloading address, so that first-in, first-out service can be given. Also, it contains the addresses of the beginning and end of the actual hopper, so that the loading and unloading programs are completely independent of the hopper's size and location. The information that is entered in this hopper requires only one word, and consists merely of the call store address of the originating register whose call is ready to have dial tone

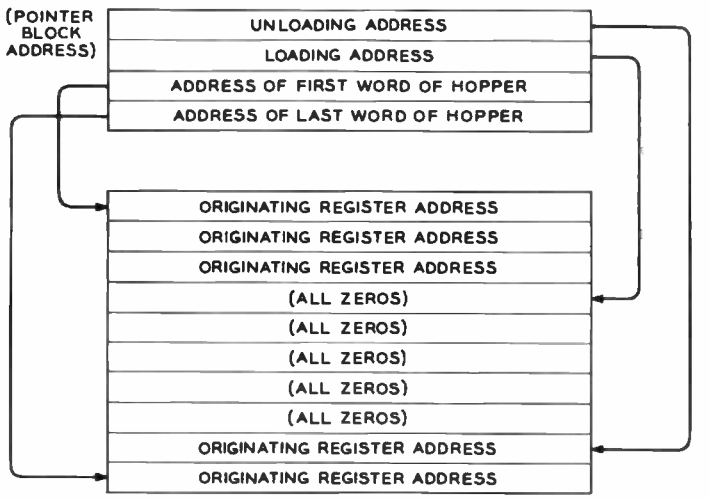


Fig. 8 — Release dial tone hopper.

removed. (A request to remove dial tone is made after detection of the first pulse, so that it will actually be removed by the time the first digit is dialed.) Many hoppers are of this same form, except for different kinds and quantities of information in each entry; some require two words per entry.

Once a dispenser program receives control, it empties its buffer of all work it finds there. Thus, if the remove dial tone dispenser finds the five entries shown in Fig. 8, it will pass control five times to the same task program, each time with central control index register X containing one of the originating register call store addresses unloaded from the hopper.

The dial pulse digit and abandon hopper (whose entry format is shown in Fig. 9) has a dual purpose. An interrupt scan program detects both types of input by the same timing; the only difference between

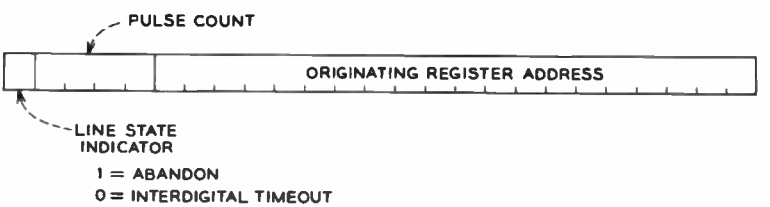


Fig. 9 — Format of entry in dial pulse digit and abandon hopper.

them is that the line is off-hook if a digit has been completed, but on-hook if the call has been abandoned. It is more efficient, therefore, for the interrupt-level program to put both types into the same hopper, with one bit of the entry indicating on-hook or off-hook, than to make the decision itself and place them in separate hoppers.

Each priority class has an ordered bits buffer; the one for class B is shown in Fig. 10. Such a buffer consists of a call store word, each bit of which serves as a flag for its associated program. The address of this program is in an accompanying table in a position corresponding to the flag's bit position in the word. Thus, in Fig. 10, the word BOBB contains two 1's. The rightmost 1, in bit position 3 (starting with 0 on the right), indicates there is work to be done by the program whose address is in word number 3 of the table BP0. This program, as seen from the

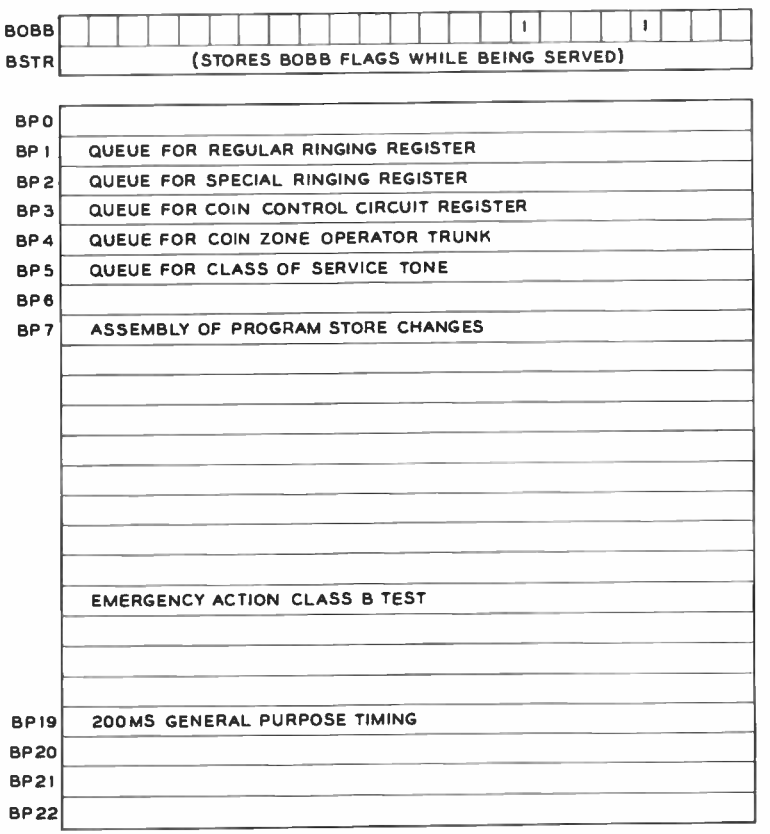


Fig. 10 — Class B ordered bits buffer.

figure, serves the coin control circuit register queue, so the flag would have been set to 1 by some program that needed such a call store register but found none available. At the same time, it would have left waiting on the corresponding queue whatever call register (perhaps an originating register) was already associated with the call. The program triggered by this flag from the ordered bits buffer will check to see if any coin control circuit registers have become available; if so, the one or more calls represented by registers on the queue will be served.

The other 1 in word BOBB of the ordered bits buffer, in bit 7, indicates that there are program store changes to be assembled preparatory to writing new cards for the translation area of the program store. This does not occur for days or weeks at a time, and the flag would have been set to 1 by a special teletypewriter message requesting this action.

The ordered bits buffer dispenser program begins by copying the contents of BOBB into the following word BSTR and zeroing BOBB. Then new flags can be set in BOBB even by interrupt-level programs while the 1's in BSTR are being served and erased from right to left, using the special TZRFZ instruction. (See Table III, step 5.)

4.3 *Interject Work*

The sixth class of base-level work is the interject class; it is not regularly scheduled at all in the ABACA ··· sequence. It has the highest priority of all, however, for a check is made after each task program in each of the other five classes for the existence of interject work, and if found it is done immediately. It is initiated by interrupt-level programs when they encounter work that cannot tolerate the delay it might suffer if put even in a class A buffer, but which should not be done in the interrupt program itself. The latter constraint might be that it is too lengthy for an interrupt level; or that it might interact harmfully with the interrupted base-level program, if both should try to change the contents of the same call store location. Therefore, it is scheduled to be done at the first natural break, by being interjected between task programs. The check for the presence of interject work is accomplished automatically as each task program returns control to the proper dispenser program by transferring to the standard program address "RETURN." At this point, the pair of instructions

MK	RETN	(move contents of "RETN" to index register K)
TKP	0, K	(if K's sign bit is positive, transfer to address in rest of K)

occur. "RETN" is the symbolic address of a call store word containing the dispenser program address to which control should be returned by each of its task programs; the first thing a task dispenser program does upon gaining control initially is to place this address in RETN. The sign bit of RETN is normally positive, but is made negative as a flag by any interrupt-level program wishing to interject a job. Thus the first two instructions of the program at RETURN read this call store word RETN and transfer to the address it contains, unless the sign bit has been made negative. In that case, program control passes to the succeeding orders, which form the interject control program. This program has certain special bookkeeping functions to perform, but essentially it transfers control to the one or more interjected jobs and then returns control to the dispenser address that was in RETN.

The interject priority class consists of a single dispenser program, the interject ordered bits buffer. At present only three kinds of jobs are definitely planned for this buffer, though more can be added if future requirements dictate. One is an emergency-action interject test, and another is the unloading of the hopper containing highest-priority reports from the network execution program. The third is the updating and distribution of information from the 100-millisecond timetable. For the first two, flags are set in the interject ordered bits buffer (and the master flag in RETN) when the relevant interrupt-level programs encounter the need either for the emergency test or the highest-priority network report. The flag for the 100-millisecond timetable is set by the 5-millisecond input-output low-priority timetable every twentieth entry.

The interject 100-millisecond timetable is central to all base-level timing. It is shown in Fig. 11, and will be described only briefly, because it is similar to the high-priority timetable for the H and J interrupt levels which has already been discussed in some detail in Section 3.1.3.

The supervisory line scan should be made every 100 milliseconds, nominally. There is no advantage in scanning excessively often, even in slack periods; routine maintenance checks and various administrative programs can usually make better use of the time. To prevent excessive scanning the line scanning program receives control from the class D ordered bits buffer when its flag bit is found to be 1. That flag is set once every 100 milliseconds from the interject 100-millisecond timetable. On the other hand, an extension of the 100-millisecond period, even if long enough to be noticeable, carries no serious penalty; in fact, since line scanning does take an appreciable amount of the processor's time, it is desirable that during overload proportionately less time be expended looking for work and more spent on work already in progress.

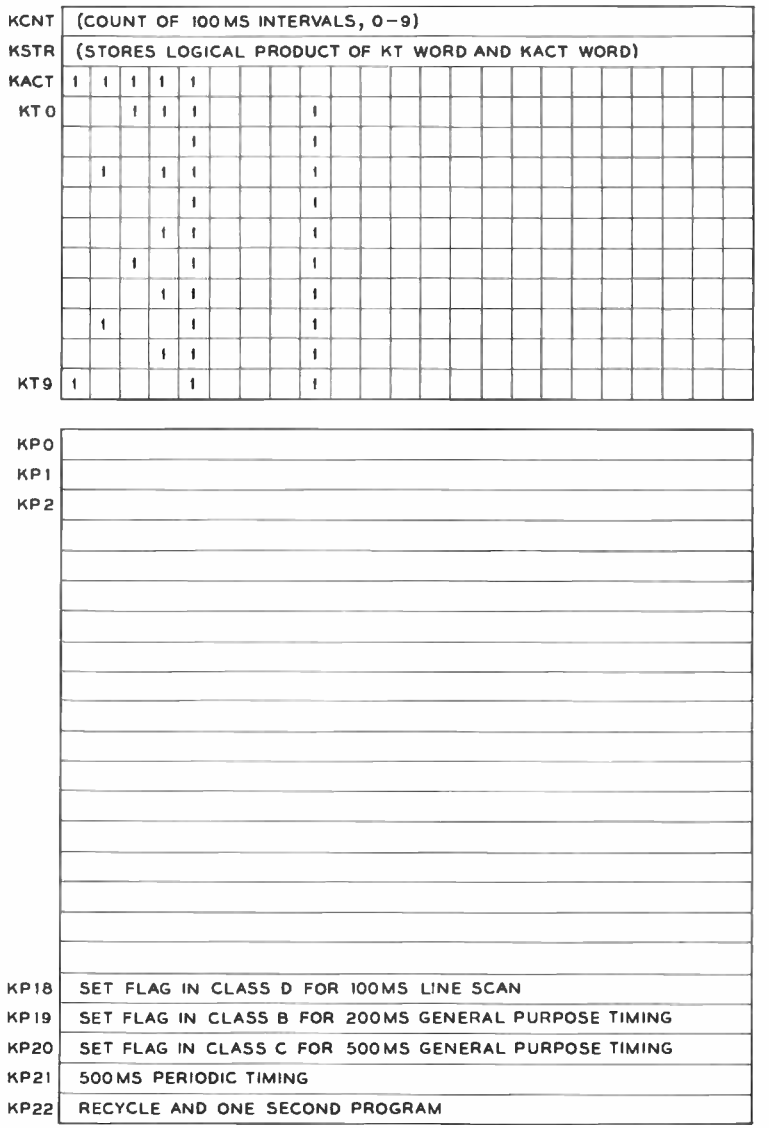


Fig. 11 — Interject 100-millisecond timetable.

The periods between executions of class D programs may exceed 100 milliseconds during periods of heavy traffic, and when they do the class D supervisory line scan flag will be set to 1 more often than it is served and made 0.

In column 19, 1's alternate with 0's, so the program whose address is

in row 19 of table KP0 will be entered every 200 milliseconds. All this program does is set a flag in the class B ordered bits buffer for 200-millisecond general-purpose timing, which will be discussed later.

All the programs entered from this table KP0 are, of course, executed as interject jobs, and as such must be kept short lest they themselves delay other interject work. The program whose address is in row 21, entered every 500 milliseconds, updates another timetable which contains 24 rows and thus is cycled through completely every 12 seconds. This timetable, however, has no associated table of transfer addresses; instead, after selecting the current row and ANDing it with the activity bit word, it ORs the result into the class C ordered bits buffer. (This divides the latter's bit positions into two kinds: those set by the 500-millisecond interject timetable; and those set by other means, like those of the class B ordered bits buffer already discussed. The ones set by other means render the corresponding columns of the 500-millisecond timetable useless, of course.)

4.4 *Timing*

As explained in earlier sections, the detection and discrimination of input signals is assigned to interrupt-level programs, and the resultant work of processing them to the base level. Somewhat analogously, time — in the form of a signal every 100 milliseconds — is an input to the base level, where it is detected and analyzed by interject programs, and the work that is found due is passed on to the lower base-level classes. When the intervals being timed become so long that the delays in performing base-level work are negligible by comparison, some of the timekeeping itself can be removed from the interject level. Thus class C has a 3-second timetable, and class E, 15-second and 15-minute timetables.

There remains an assortment of timing requirements not fulfilled by the timetables. Typically, certain kinds of call register programs may reach a point where a delay of a few hundred milliseconds or a number of seconds is required. One example is a TOUCH-TONE test trunk program, which must send out a failure signal 15 seconds after initiation of the test if a correct sequence of signals has not been keyed in by that time. General-purpose timing for such uses is provided by timing linked lists.

A hypothetical one-way 200-millisecond linked list is shown in use in Fig. 12. The only call store memory required for this list when not being used is the single word TIM200, sometimes called the "head cell"

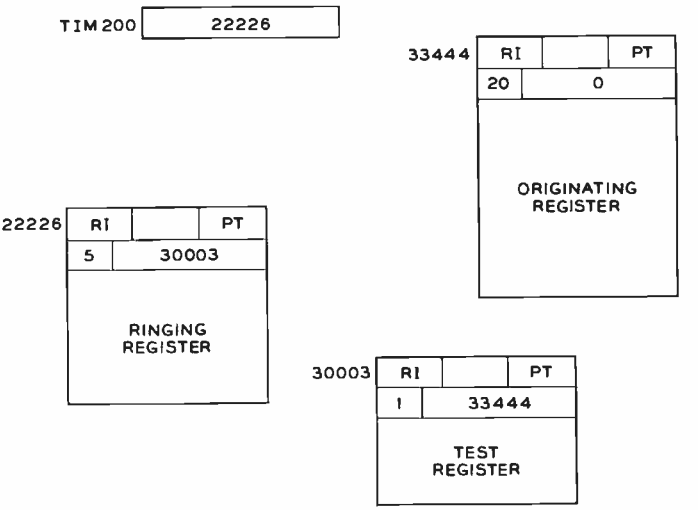


Fig. 12 — One-way 200-millisecond linked list for general-purpose timing.

of the linked list. When not in use, this word contains zero. In the figure, the ringing register was the last to be added of the three registers undergoing timing. The program which added it to the list found TIM200 containing 30003, the address of the test register. The ringing register was inserted in the list by having the "30003" stored in its second word, along with the "5" which specifies the number of 200-millisecond units of timing that are desired. Its own address, 22226, becomes the new contents of TIM200. The timing for whatever registers have been put on the list is administered by a task dispenser program given control every 200 milliseconds from the class B ordered bits buffer (Fig. 10). This program goes through the list, subtracting one from the count in the left part of the second word of each register. When the count is reduced to zero (as it will be next time for the test register, whose count is shown as "1"), the register is removed from the list (by replacing 30003 in the ringing register by 33444) and control is given to the appropriate task register program. The procedure by which this appropriate program is determined is used not only here, but in other situations in which a general program reports to a variety of other programs through their associated registers. Instead of a full program store address being stored in the call register, the RI and PT items in the register's first word are used. The RI ("register identification") selects a table of transfer addresses associated with the type of register,

and the PT ("program tag") is used to select from this table the address of the proper task program. (In this case, of course, that will be whatever program has been designed to take appropriate action when the test register times out.) This selection from the table, however, is made by altering the PT number slightly before using it as an index. This same alteration is performed by all programs reporting a certain class of information — including timing notification — to any call register. The report by some other program of a different class of information — e.g., a scan point change — would not make this alteration, and therefore could use the same PT number to select from the same table a different program. This means of distinguishing inputs eliminates many decisions within the task programs.

Note that to remove a register from a one-way linked list at an arbitrary time between scheduled inspections requires tracing through the list from the beginning. Since this could consume considerable time for a long list, two-way timing linked lists are also provided. Their use is quite similar, but each register must supply two words for the linking, one pointing forward to the next linked register and the other pointing back to the preceding. This means that any register in the chain can be directly dropped out, since it contains enough information for the linkage to be mended.

Most programs requiring general-purpose timing have an associated register to link; for those that do not, a common pool of timing registers is provided.

4.5 *Task Programs*

The variety of task programs precludes a comprehensive treatment within this paper, but at least some of this variety may be indicated. The word "program," when used to designate a subdivision of some larger program, is a static term referring to a set of instructions capable of performing a certain action or group of related actions when some or all of these instructions are executed in sequence. When control is given to such a program at one of its entry points, however, the sequential execution that follows typically runs through only part of the orders in that particular program, and then on into parts of other programs. It is this dynamic unit, consisting of the complete set of instructions executed in unbroken sequence, that is most significant in understanding the flow of control governed by the main program. "Task program" is often used in this dynamic sense, but "task execution" might be a less ambiguous term for the complete sequence of order executions from the

time a dispenser relinquishes control until control is returned to that dispenser. Thus, during a task execution only task programs are executed (apart from possible insertions of interrupt programs), but many and varied task programs may contribute to a single task execution.

There can be no "real-time gaps" within a task execution; such a gap necessarily signals the end of a task execution, and the suspended call or other function can regain control only through the main program mechanism. Task executions differ greatly in length, of course, depending on the amount of continuous work it is possible and desirable to do. As explained in earlier sections of this paper and in the paper on call processing,[2] the work that is done for one call at one time tends to be small, due to circuit limitations and timing requirements as well as the natural fragmentation of customer actions.

Task executions vary also in the number of different "static" task programs they involve. One well known but important programming technique that tends to use many programs in one task execution is the use of subroutines. In the No. 1 ESS program these range from short programs performing simple conversions of equipment addresses from one form to another, to complex network and translation routines which themselves use several levels of subroutines internally.

A subroutine, as the term is generally used, returns control to its client program when it has completed its work. Often, however, two or more task programs arrive at a point where the remaining work to be done in both or all cases is identical, and they can be joined at that point. There is no need to preserve the identity of the source merely in order to return control properly at the end of the task execution, because the standard transfer to RETURN, explained earlier, solves that problem.

The supervisory line scan is an example of a self-contained task program. It could be considered, of course, as two programs: one that repetitively compares the line scanner points with the line state bits, and another program or subroutine to which control is transferred when a proper match of these readings occurs that makes the entries in the service request hopper. If these two closely related programs are combined, however, they become a self-sufficient whole.

A contrasting example is provided by the task execution that begins with a digit just unloaded from a digit hopper. The initial task program is a part of the general call processing functional subdivision called "digit analysis, lines." It begins with the assumption that the K register contains the new digit in bits 21 through 18 (with the digit "0" registered in binary as "1010," corresponding to the ten pulses generated by a dial

telephone), and that the X register contains the address of the originating register associated with the call. This task program could have received control from either of two task dispenser programs, depending on whether the digit was produced by a dial or TOUCH-TONE telephone set. In handling the digit up to this point, the actions of the two task dispensers differ. The dialed digit task dispenser distinguishes the digit entry from the abandoned call entry it may also find in its hopper; the TOUCH-TONE digit task dispenser converts its hopper entry from a code indicating two active tone generators into a binary digit. From this point on, however, the distinction vanishes, and subsequent programs treat the digit in the same way regardless of its original form.

The initial decisions depend on the contents of the originating register whose address is in the X register. This register is a block of 16 consecutive call store words, assigned to a call as soon as the originating line is connected to a receiver and given dial tone, and retaining information for that call until a ringing connection is set up (for a local call) or until outpulsing is completed (for an outgoing call). The initial arrangement of information in the register is shown in simplified form in Fig. 13. (The type of information stored in some words or parts of words varies with the type of call, and can change as the call progresses.) The program first checks the single bit END in word 6 to see if dialing has already been interpreted as completed; if so (an unlikely event) the extra digit is ignored and control is immediately returned to the dispenser program by a transfer to RETURN. Assuming that dialing has not been previously completed, the program adds one to the digit counter (item DC in word 12), stores the new digit in its proper digit slot (DS1–DS10) as determined by the new value of the digit counter, and then checks to see whether that new value matches the number in item DCA in word 6. This number has been put there at some earlier point in the call to tell the present program whether additional decisions must now be made or whether nothing is required beyond the digit storage that has just been accomplished. If DC is not equal to DCA, control is returned to the dispenser program. If DC equals DCA, control is given to the program determined by item AADD in word 6. (The common program device used here to reduce item AADD from a full 18-bit call store address size is to transfer into a fixed table of transfer instructions, using the value of AADD to select the point in the table to which control is transferred.) The program receiving control could be any one of about thirty. (Since AADD is eight bits, the maximum table size is 256, of course.) For example, it might be the program analyzing the first digit, in which case DCA would have been set to

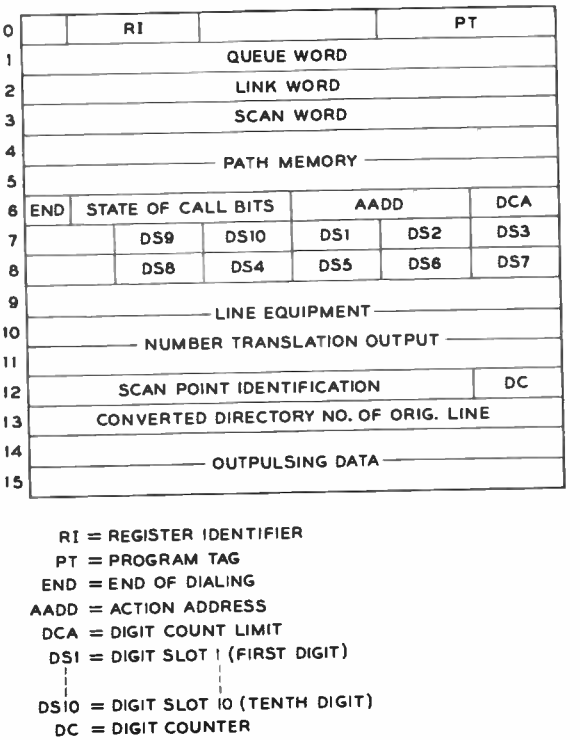


Fig. 13 — Simplified layout of originating register.

"1" by the program that connected the line to a receiver and dial tone. The analysis of the first digit involves checking to see whether it is a "1", "0", or the eleventh button on a TOUCH-TONE set, since any of these can have special significance. In an office without the "0" prefix option, an initial "0" necessarily completes dialing and requires a connection to an operator. In this case the task execution continues through subroutines in the translation and network area, transferring to RETURN only when it has made its first request for network action by placing proper entries in a peripheral output buffer. The foregoing assumes that no difficulties are encountered. If all operator trunks are busy, a request for a connection to overflow tone will be made instead; or, if no peripheral output buffer is available, the originating register will be left waiting in the corresponding queue before control is relinquished.

Referring again to the possible actions following a match of DC and

DCA, the program to which index AADD points might be one involved in providing a special service such as temporary transfer to a customer. The temporary transfer program would be controlling the call in this case, and would have seized an originating register to serve as a digit collector, placing the desired number of digits in DCA and a number in AADD that would lead to the appropriate temporary transfer task program.

Thus this digit analysis task execution may complete its work in a few order cycles, or it may branch into a multitude of different task programs before its sequence of instructions finally reaches a transfer to RETURN.

V. THE GENERIC PROGRAM

One of the important basic requirements stated earlier for the No. 1 ESS program is that it be generic — that is, that a single program be able to serve many offices with different features and characteristics, and during continuing growth. If the program does not change, then data to which it refers must change to take account of the different features, numbers of lines and trunks, types of signaling, and other variables. From a theoretic viewpoint, this concentration of all changeable information into one place instead of scattering it through the data and address fields of many instructions may seem a trifling difference. However, when a large number of offices are involved such an arrangement has substantial economic advantages in compiling programs, adding features, and making changes.

The constraint this places on the program design can be stated quite simply as the inability of the program to know anything that can change. Any quantity or address or option that can vary, as an office grows or from one office to another, must be looked up by the program in a concentrated data area of the program store. The call store may be used where faster reference is needed, but this may be considered an indirect reference to the program store, which must back up all long-term call store information. (In particular, when the office is first put into operation, the call store must be written to its proper initial state from program store data.)

The effect the generic program requirement has on memory allocation is essentially that, apart from the generic program itself, program store and call store are each divided into two parts. One area is of fixed size and fixed layout (though not fixed contents, of course), and the other

can be expanded and rearranged. A good call store example is provided by the hopper shown in Fig. 8. The four-word pointer block must be in the fixed layout part of call store, so the loading and unloading programs can refer to PBA, PBA+1, etc., in the address fields of their instructions. The hopper itself, however, could not be included in this area unless it were made big enough for the largest one needed in any office; but it is economical to reduce its size in smaller offices. The storage in the pointer block of the beginning and end addresses of the hopper itself allows the latter to grow or shrink as required in the expandable part of call store.

Similarly, call registers such as the originating and ringing registers are assigned in expandable call store, and while idle are chained together in linked lists, one for each register type. Only the two-word "head cell" of each linked list, containing the addresses of the first and last registers, is assigned in the fixed layout part of call store.

Nearly all call processing programs use such linked lists as their sole reference in seizing and releasing registers as needed. In program store, however, there must be a record in some form of the absolute addresses of all registers of each type. The call store initialization program, used when the office is turned on initially, must have such information; a few other programs also find it useful. This information is packed quite compactly, since reference to it is made only infrequently. Even so, the amount of program store space required for recording the locations of all registers in the biggest office is more than can be afforded in a small office. Hence the same division into fixed layout and expandable memory is made within the program store data area. A single word for each call register type is assigned at a certain address in the fixed layout portion. This word contains the address in the expandable part of program store of a variable-size block containing the locations of all registers of that type.

VI. SUMMARY

The reliable and prompt switching functions required of No. 1 ESS put stringent requirements on the program organization. The types of equipment with which the program must work and the services that must be provided are large in number. For a wide range of office sizes and traffic volumes, economy must be stressed, both for the initial installation and for growth. The program organization outlined was chosen because it satisfies these requirements. Certainly other means of implementation are possible.

The interrupt system helps to guarantee prompt attention to tasks that require it. To satisfy the input-output tolerances, it is also necessary to limit the time during which a given task is continuously processed. Most call programs have natural break points, and others must be created to be compatible with the equipment. Lengthy maintenance and administration programs are arbitrarily divided into segments whose execution time does not exceed a suitable limit.

This fragmentation of work, in turn, requires a system of call store registers and buffers in which to store data until processing can be resumed. These and other memory units are designed for ease of growth and economy throughout the range of office sizes.

Both the input-output and base level main programs follow simple schedules which check for work to be done according to its urgency, without consuming an inordinate amount of time.

To increase the efficiency of the very frequently executed input-output programs, special orders were devised, and certain groups of these orders are repeated in program store to avoid the extra transfer time that program loops would cost. For the greater part of the program, however, a very general order structure is used, which should be equally well suited to functions not yet conceived. For this majority of programs, furthermore, subroutines are heavily used to reduce program storage space. This organization also makes it easier to add new features to the program, since use can be made of existing program blocks and subroutines.

However, a major aim has been to minimize the causes that require an addition or any other change to the program itself. This is done by effecting a separation of the program proper from the parameters associated with the features of a particular office and the variables of growth.

VII. ACKNOWLEDGMENTS

The program organization of No. 1 ESS is the product of many contributors, from the areas of system engineering and design as well as call and maintenance programming. The choice of program techniques was closely coordinated with the analysis of the actions required for call processing, and depended also on the instructions, both general and special purpose, provided by the central control design. In particular, we wish to acknowledge the contributions of S. Silber, who provided many of the basic ideas for the main programs and is responsible for their actual design and programming.

REFERENCES

1. Keister, W., Ketchledge, R. W., and Vaughan, H. E., No. 1 ESS System Organization and Objectives, B.S.T.J., this issue, p. 1831.
2. Drew, G. G., Carbaugh, D. H., Ghiron, H., and Hoover, Mrs. E. S., No. 1 ESS Call Processing, B.S.T.J., this issue, Part 2.
3. Harr, J. A., Taylor, F. F., and Ulrich, W., Organization of No. 1 ESS Central Processor, B.S.T.J., this issue, p. 1845.
4. Downing, R. W., Nowak, J. S., Tuomenoksa, L. S., No. 1 ESS Maintenance Plan, B.S.T.J., this issue, p. 1961.
5. Martellotto, N. A., Oehring, H., and Paull, M. C., PROCESS III, A Compiler-Assembler for No. 1 ESS, B.S.T.J., this issue, Part 2.

No. 1 ESS Maintenance Plan

By R. W. DOWNING, J. S. NOWAK and L. S. TUOMENOKSA

(Manuscript received January 28, 1964)

The No. 1 electronic switching system has a much higher concentration of control than previous telephone systems. This makes the task of providing continuous telephone service more challenging than ever. It is important that troubles be detected almost instantaneously, before many calls are affected. The main subjects of this article are (1) a scheme for duplicating control equipment and (2) a description of circuit and program facilities for taking advantage of this duplication to detect, to automatically recover from, and to analyze troubles.

I. OBJECTIVES

The success of a commercial telephone switching system can be measured on the basis of customer satisfaction and economic considerations. For customer satisfaction, a telephone switching system must provide continuous and accurate service without unreasonable delays. This quality of service must be provided 24 hours a day throughout the design life of the system. Thus a successful system must be both dependable and maintainable. Dependability is defined as a measure of service continuity and accuracy; maintainability is defined as a measure of the ease with which component failures can be detected, diagnosed, and corrected. A high degree of system dependability and maintainability have been considered extremely important design objectives[1] throughout the development of No. 1 ESS.

The No. 1 ESS dependability and maintainability objectives have been chosen to be competitive with the existing electromechanical telephone switching systems. The dependability objectives are: the system down time should not exceed 2 hours over its 40-year life, and the calls handled incorrectly should not exceed 0.02 per cent. The maintainability objectives of No. 1 ESS are: to design a system in which troubles can be located and repaired easily and rapidly and which can be left unattended for extended periods of time.

II. OVER-ALL PLAN

The use of high-speed data processing circuits in No. 1 ESS has brought about an increase (as compared to existing switching systems) in the time sharing of circuits and consequently in the centralization of control functions. This centralization has many advantages, but it can make the system more vulnerable to component failures if not properly handled. For example, if no redundancy were provided it would be possible for a single component failure in the No. 1 ESS central control to cause a complete failure of the system. To avoid this catastrophe, duplication of units and other forms of redundancy are used throughout the system. Thus in No. 1 ESS, unlike its predecessors, redundancy is provided primarily for dependability rather than to increase the traffic handling capacity of the system.

Duplication is necessary to allow the system to operate in the presence of troubles. In addition, to insure that calls are handled correctly, interruptions of call processing due to component failures must be minimized. No. 1 ESS depends primarily on circuits for trouble detection. The check circuits activate a circuit labeled the "interrupt sequencer." This circuit transfers program control from the call processing programs to maintenance programs called "fault-recognition" programs. The function of these programs is to determine quickly an operational system configuration, establish it by appropriate switching of duplicate units, and then return to call processing. To minimize the length of interruptions to call processing, duplicate units can be switched at speeds comparable to the cycle time of the unit.* For errors and most faults,† the fault-recognition programs can be completed before the interruption has interfered with call processing, i.e., in a few milliseconds. For faults in the circuits interconnecting the units, the analysis by fault-recognition programs may extend to a point where some interference to call processing results.

Duplication increases system up time. However, a finite probability of simultaneous failures in duplicated subsystems remains (see Section 3.2). To decrease this probability, an extensive effort has been made in No. 1 ESS to design intrinsically reliable circuits and to minimize the repair time. To achieve intrinsically reliable circuits, long-life components such as silicon and magnetic devices have been chosen. Liberal margins have been provided between component ratings and the actual operating

* For example, the central controls can be switched between active and standby states (see Section 3.3) by operating flip-flops, whereas network controllers, which have a 25-millisecond cycle time, are relay switched.

† An error is defined as a malfunction, the symptom of which cannot be reproduced under program control; a fault is a malfunction whose symptom can be reproduced.

conditions. All logic circuits were designed using worst-case techniques.[2] To decrease the repair time, programs and circuits are provided to conduct diagnostic tests on a faulty unit and then inform the maintenance personnel of the test results. Maintenance dictionaries are provided for translating these test results to the location of the faulty package(s). By using standardized packages[3] and plug-in techniques,[4,5] faulty components are made readily accessible and repairable. No in-service adjustments (short of package removal) are required.

To summarize, the No. 1 ESS maintenance plan has the following major facets:

(1) long-life components and conservative circuit design are used to achieve an intrinsically reliable system;

(2) vital parts of the system are duplicated to retain an operational system in the presence of component failures;

(3) circuits are provided for trouble detection and switching of duplicate units, and fault-recognition programs are provided to identify the faulty unit and to control the switching to enable the system to recover from component failures rapidly and automatically;

(4) diagnostic programs are provided to automatically test a faulty unit, maintenance dictionaries are provided for translating diagnostic results into the location of the faulty package, and plug-in techniques are used to facilitate the replacement of the faulty circuit.

Section III of this paper describes the duplication and switching plan. The maintenance circuits are described in Section IV, and maintenance programs in Section V. Section VI reviews the man-machine reaction to trouble and describes the plans for producing the maintenance dictionaries.

III. DUPLICATION AND SWITCHING

Functionally, the system can be divided into a central processor and peripheral units (see Fig. 1). The central processor consists of program stores, call stores and the central control. The peripheral units include the switching network,[6] scanners,[7] signal distributors,[7] and the master control center.[8]

To process a call, a chain of units consisting of a central control, program stores, a central pulse distributor, some scan points, etc., must be operating properly. A failure in any one link in this chain will appear as a system failure. Our dependability objectives dictate that, on the average, the sum of the down times of all the links in this chain should not exceed 2 hours during the 40-year design life of the office.

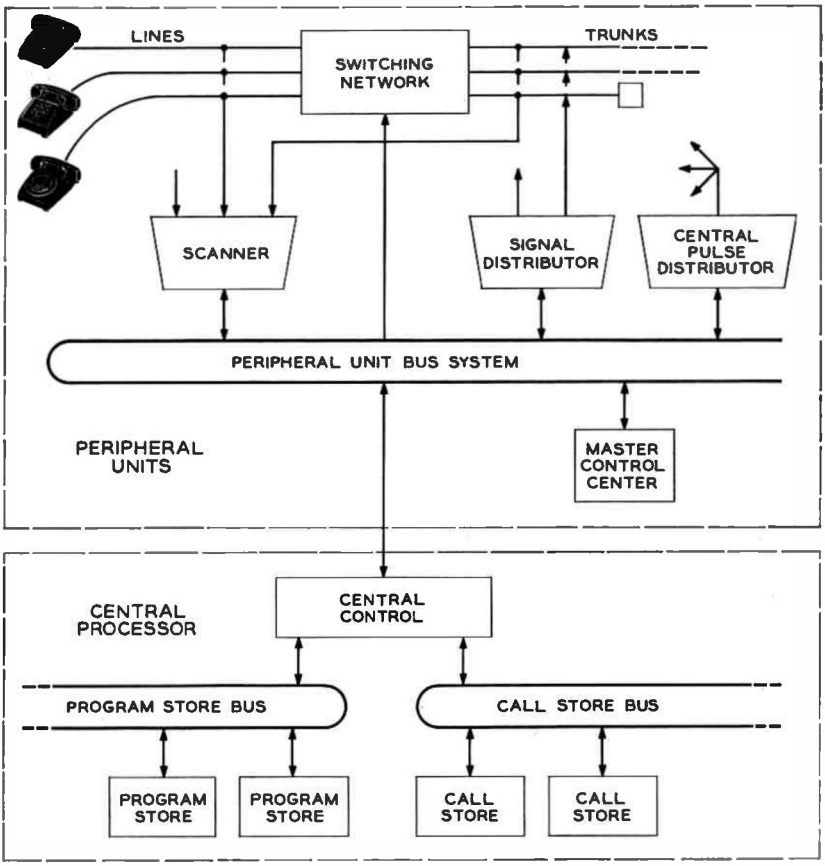


Fig. 1 — No. 1 ESS block diagram.

3.1 *Reliability Predictions*

To determine what form of duplication, if any, is required for each subsystem, one is interested in calculating the expected down time for each subsystem. The expected down time depends on the mean time between failures for each subsystem and the mean repair time.

Each subsystem is made up of large populations of many different component parts. These are operated at varying degrees of severity with respect to their nominal ratings. Failures observed in these types of units have, in general, been found to be distributed randomly, and the failure rate tends to remain constant with time. Assuming the com-

ponent failure rates remain constant through the 40-year design life of the system, i.e., that there is no wear-out effect, mean time between failures for the various subsystems can be calculated from the component failure rates. For some components, such as resistors, the failure rates can be predicted with considerable confidence, since plentiful data are available from systems similar to No. 1 ESS. For other components, such as diodes and transistors, there is more uncertainty about the failure rates, since there is no previous experience available with the particular transistors and diodes under the circuit stresses encountered in No. 1 ESS. Accelerated testing of components and the performance statistics of similar components in other systems give some indication of the failure rates we may expect.

Fig. 2, which plots down time as a function of the mean time between failures, indicates the extent to which duplication improves subsystem dependability. Consider the central controls as an example. A central control contains approximately 45,000 diodes, 13,500 transistors, 35,000 resistors, 225,000 soldered connections, 55,900 connector terminals, and a small number of pulse transformers, capacitors, etc. To meet our dependability objective of 2 hours down time in 40 years for the system, the central control down time should not exceed approximately 1 hour in 40 years.* Fig. 2 shows that with duplication we require a central control with a mean time between failures of 1200 hours to meet our objective.

Points A and B in Fig. 2 represent calculations which were made for two different sets of failures rates.† We expect that the component failure rates that will actually be experienced will fall in the range between these two points, and that they will be reasonably close to the point where our objectives are met.

* Based on the number of components in central control as compared with the rest of the system.

† Component failure rates assumed, in FITs:

	A	B
transistors (29A)	5	50
diodes (mostly 447A and 449A)	3	25
resistors	2	10
pulse transformers	20	40
solder connections	0.5	1
connector terminals	1	2.

(A FIT' is defined as one component failure in 10^9 operating hours.)

The component failure rates listed in columns A and B above will result in mean times between central control failures of 1800 and 360 hours respectively. Component failure rates in this range have been achieved in other systems. Although no system known to us has achieved the rates shown in column A, accelerated tests on components indicate that these rates may be realizable.

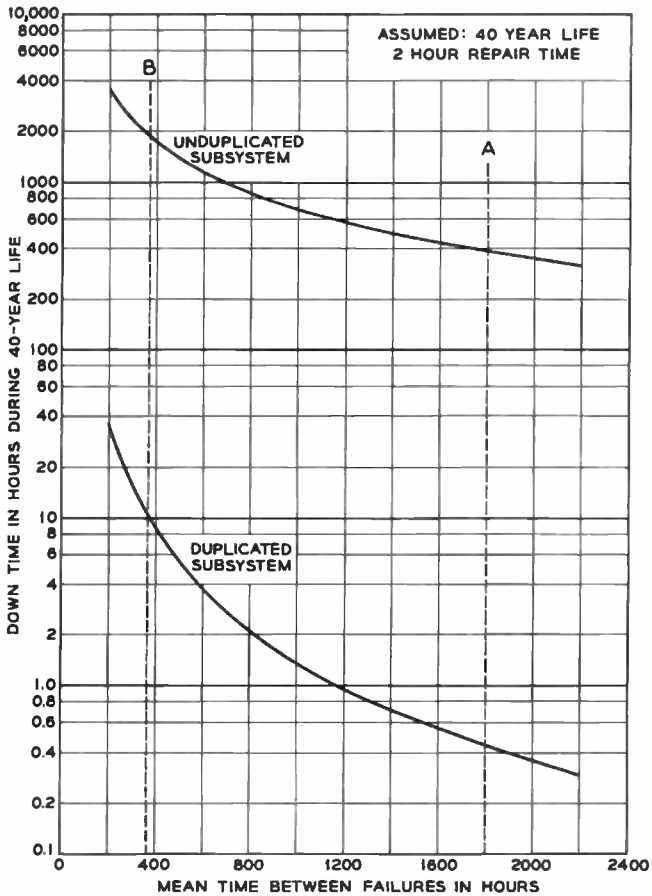


Fig. 2 — Down time vs mean time between failures for system with assumed 40-year life, two-hour repair time.

Similar studies in the early stages of development indicated the necessity for some type of duplication of all circuits where failure could affect a substantial number of customers.

Duplication on a subsystem level rather than on a circuit level was, in general, found to result in a more economical and maintainable design. The following sections describe the duplication plan adopted.

3.2 *Duplication and Switching in the Central Processor*

The major units of the central processor are interconnected by two bus systems:[9] the program store bus and the call store bus. The duplica-

tion in the community formed by the central control, program store bus, and the program stores will first be described.

The central control and the program store bus are completely duplicated. Each office, large or small, will have need for only one (duplicated) central control and one (duplicated) program store bus. Therefore no growth problem exists.

The program store requirements, on the other hand, are expected to vary from approximately 130,000 to 490,000 words, depending on the size of the office. To allow the number of program stores to vary from the minimum of two to the maximum of six in single-store increments, a "split" type of duplication was adopted. As shown in Fig. 3, the program stores can be viewed as links in a closed chain. Each store contains two 65,536-word blocks of information (labeled G and H) that are duplicated in the units to the left and to the right.

Each information block is assigned a 4-bit name, coded in 2/4 code. Identical names are assigned to the duplicate G and H information blocks. When a central control wants to fetch an instruction or data word from program stores, it broadcasts a 25-bit command on the address bus. Four bits identify the information block; 16 bits identify the word within the block; and one bit is a synchronizing pulse which gates the contents of the bus to the program stores. The remaining 4 bits identify one of the five modes in which the store can be addressed (see Section 4.2). Flip-flops called "route flip-flops" [10] within the stores control the inputs and outputs to the buses. One flip-flop controls the selection of an input bus. Two flip-flops determine whether the readouts from the H

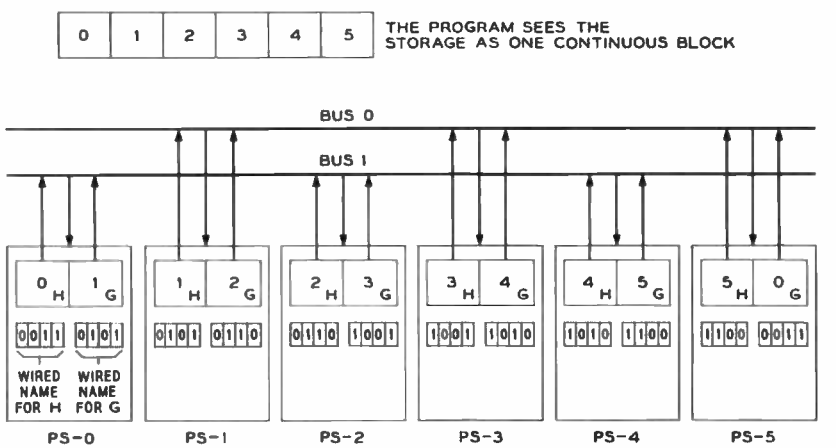


Fig. 3 — Program store duplication, six-store office.

block will be sent on bus 0 and/or bus 1. Similarly, two flip-flops control the sending of words from the G block. In each central control, there are three flip-flops which determine whether the central control transmits commands on address bus 0 or bus 1 or on both buses, and from which bus the central control receives the program store readouts.

Fig. 4 illustrates one possible address and readout routing in a three-store office. Consider a readout of an instruction from information block 1. Central control 0 broadcasts the synchronizing pulse, the 4-bit name of block 1, the 16-bit address of the word within the block 1, and the 4-bit mode (in this case normal mode) identification on bus 0. In synchronism with central control 0, central control 1 sends the identical information on bus 1. In store A, which is receiving from bus 0, the synchronizing pulse gates the address into the store. The 4-bit name code received from the bus matches the name of the G block in the store. The store therefore reads the location indicated by the address and sends the word stored there back to central control 0 via bus 0. Similarly, store B responds to the address on bus 1 and sends the duplicate word back on bus 1 to central control 1. Thus the identical word is fetched by the two central controls using separate routes to the two different stores.

One of the central controls is referred to as the active, the other as the standby. The status of which central control is active and which is the standby is controlled by a flip-flop in each central control. The active has a preferred selection with respect to store access; i.e., the active always fetches its own instructions and data. The standby may or may not fetch its own instructions, depending on the information block being read and the existing configuration. The active may transmit on one or

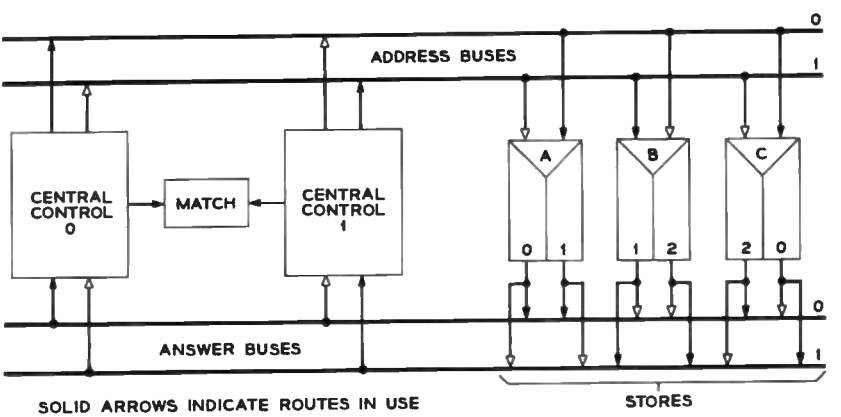


Fig. 4 — Three program store system — normal configuration.

both buses of a duplicate pair; the standby may transmit on at most one bus, and only if that bus is not used by the active. Also, as will be noted below, the active normally controls all the peripheral units.

The programs, with the exception of some maintenance programs, can be written without regard to the interconnection configuration. The program addresses a word; the address consists of the block name and the identification of the word within the block. The particular route used to obtain the word is determined solely by the route flip-flops within the stores and the central controls. The route flip-flops provide considerable flexibility in choosing configurations. Two examples will be described to illustrate this flexibility under trouble conditions.

As the first example, consider a situation where store C is not operating. Store C is isolated from the bus system by its operated trouble flip-flops. The configuration shown in Fig. 5 is established by appropriately setting route flip-flops in stores A and B and in the central controls. Central control 0 addresses both stores via bus 0. Words from information blocks 0 and 2 from stores A and B are sent back on both buses, since only one copy of these blocks is available at this point. The central controls still operate synchronously, and words from block 1 are obtained from independent sources, i.e., from stores A and B.

As long as any one copy of all information blocks is available, and one of the central controls and one of the buses are available, an operational configuration can be established. Fig. 6 shows the configuration which would be established if central control 0, store B and bus 1 were inoperative.

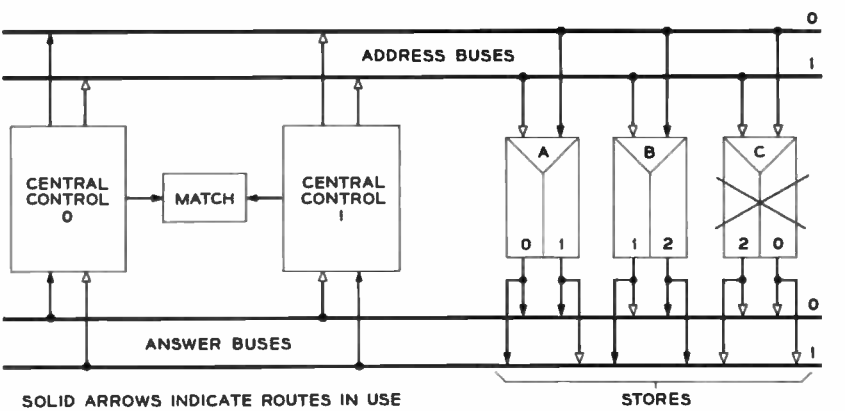


Fig. 5 — Three program store system — store C out of service.

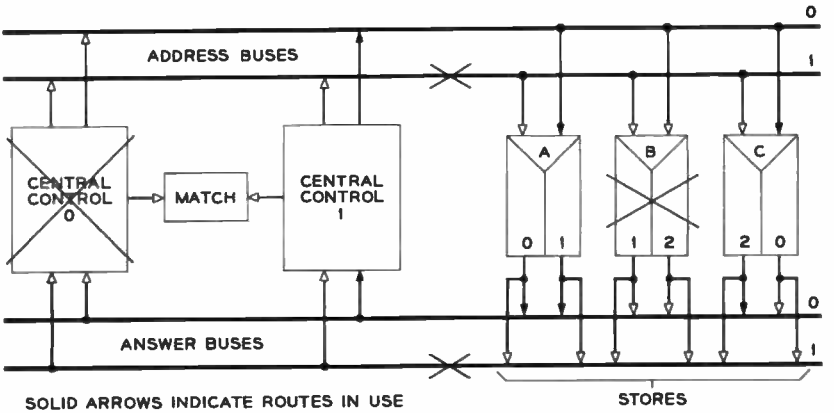


Fig. 6 — Three program store system — CC, PS, and bus out of service.

The four route flip-flops in each store that determine the routing of outputs can be controlled via the store address bus. The flip-flop that determines the address bus used is controlled via the central pulse distributors.

To summarize: since the correct operation of the central control-program store community is vital to the system operation, the central control, the bus and the program store memory are fully duplicated. To allow growth in increments of a single store, split duplication is used for memory. To allow (1) programs to be written independently of central control program store configuration, (2) rapid changes in configuration, and (3) growth without central control changes, route flip-flops are used to determine interconnection configurations, and enabling by name codes is used to select stores. In normal operation the two central controls, when possible, fetch their instructions from different stores via different paths. This provides for: (1) continuous exercising of standby equipment, and (2) an arrangement through which a thorough check of the configuration can be obtained by matching between the central controls.

The switching and duplication plan used in the call store community is nearly identical to that used in the program store community. Split duplication, name enabling, and configuration control by route flip-flops are used. The primary difference is in the size: whereas the number of program stores may vary from 2 to 6, the number of call stores may vary from 2 to 37.

3.3 *Duplication and Switching of Peripheral Units*

The peripheral units include the switching network,[11] signal distributors, central pulse distributors and scanners,[7] and the master control center,[8] i.e., all the units connected to the central control via the peripheral unit bus system.

The switching network is organized into frames. Each frame has its own controller, which sets up and takes down connections in the switching matrix associated with that frame. The active central control broadcasts addresses and commands via the common peripheral bus, which is connected to all the peripheral units. An enable pulse which the central control sends via the central pulse distributor gates the data from the peripheral unit address bus to the proper controller. (See Fig. 7a.)

The system time on a macroscopic level is divided into 5-millisecond intervals (see Section 5.1). At the beginning of every fifth interval, the central control sends orders to the controllers. During the following 25 milliseconds the controllers act on the orders and set up the requested paths.

Since failure of a controller would result in loss of the frame, controllers are duplicated (see Fig. 7b). The switching matrix in each frame is divided into two equal groups, each assigned to a controller. Under normal conditions each controller operates with its assigned switch group. Simultaneous path selections can be made within the two switch groups on the same frame. This has the additional advantage of providing continuous exercise of standby equipment. However, when a controller is in trouble its mate can be given control over both switch groups. This mode of operation, called the "combined mode," is established by operating a relay in the fault-free controller.

Since the operation of the entire peripheral system depends on the peripheral unit bus, this bus is duplicated (see Fig. 7c). Each frame is assigned four enable outputs from the central pulse distributors. The central control selects the controller and the bus by selecting one of four enable points. The central pulse distributors are provided in pairs. Two of the four enable points are assigned to one central pulse distributor, while the other two are given identical assignments in the mate central pulse distributor.

The central pulse distributors are connected by a duplicate bus system with the central control. The central pulse distributor address bus used by the active central control is determined by a flip-flop in the central control. To cause a connection to be set up in a network frame, the central control:

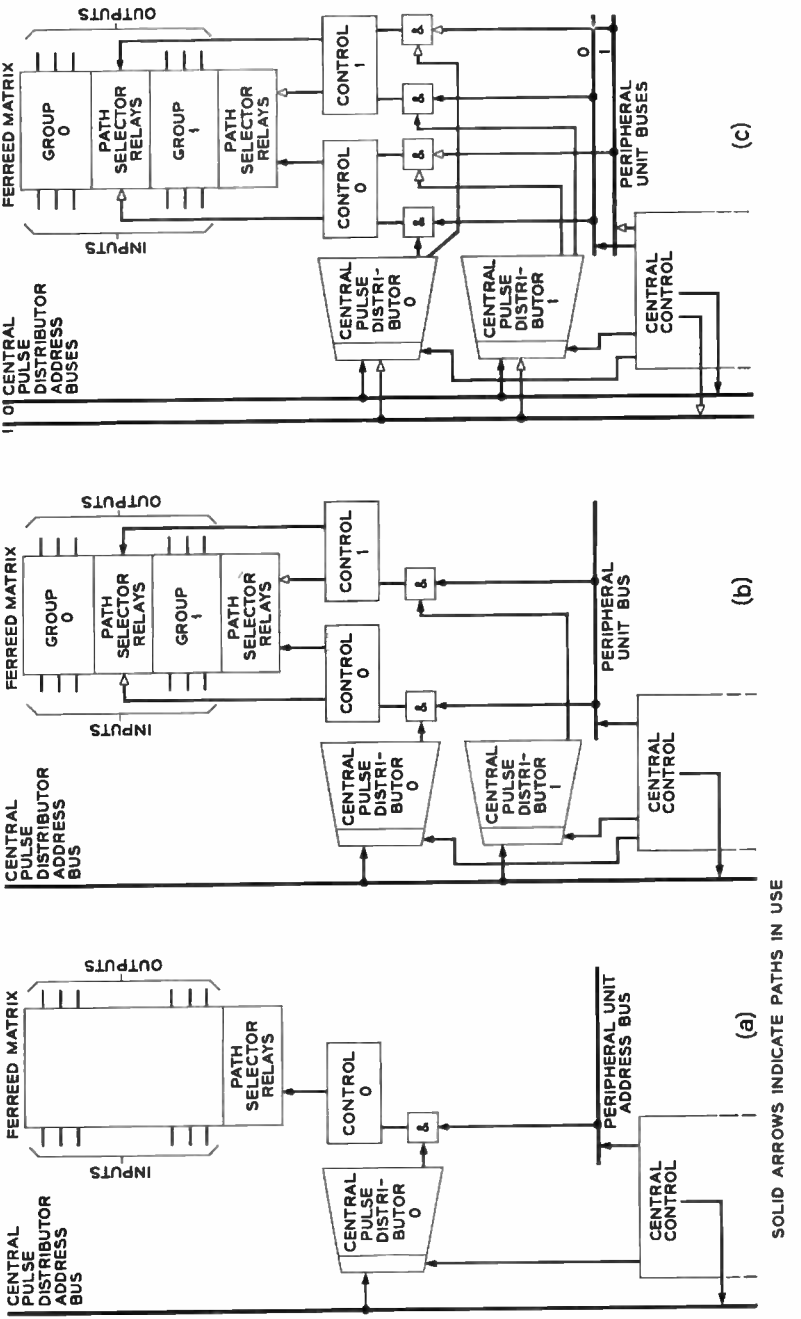


Fig. 7 — Network duplication.

(1) selects the proper central pulse distributor (1 out of a possible 16) by an individual execute signal to that central pulse distributor, and

(2) broadcasts the address of the appropriate enable point to the central pulse distributor via one of the central pulse distributor address buses;

(3) broadcasts the address of the path to be selected over the peripheral unit address bus. The enable signal from the central pulse distributor gates this address from the proper bus and controller.

Thus, unlike the central processor where the configuration is entirely determined by the route flip-flops, the peripheral unit configuration used for peripheral actions is determined by the addresses sent out by the central control. For each peripheral unit operation, the central control consults a table in the call store memory, called the "enable table," to determine the correct peripheral address bus and controller to use. In the case of network and signal distributor controllers, this consultation is also necessary to make certain that only one command is sent to the controller in a given 25-millisecond period. To switch a controller out of service the central processor places the switching frame into the combined mode of operation by operating a relay in the fault-free controller; it then modifies the enable table so that all commands to the frame will be routed to the fault-free controller. To switch a bus out of service, only the enable table needs to be modified.

The network matrix is not duplicated, but it provides redundancy, since alternate paths exist over which a particular call can be placed. This redundancy is provided to insure that during normal operation of the system the probability of calls being blocked is within the system requirements. This redundancy also serves to insure adequate network performance in case of network troubles. For example, a faulty crosspoint, after having been located and removed from service by marking associated links "busy" in the memory, will have the same effect upon system performance as a busy link.

The function of the signal distributor is to provide the central processor means for controlling relays in the system, primarily the relays in trunk circuits. A controller is provided which accepts the commands of the central control via the peripheral unit bus and translates these commands to an operation of an output relay. The output relays are in a sense analogous to the network matrix. They are not duplicated as such, but redundancy exists for traffic reasons, and in case of a relay failure an alternate trunk exists to complete a particular call. Since the loss of a controller would result in a loss of access to all 768 output points on the frame, two controllers are provided per frame. As with network con-

trollers during trouble-free conditions, each controller provides access to half the output terminals. If one controller is faulty, the remaining controller can be given access to the entire output field. As with all the peripheral units, the central processor selects the controller and the bus by using the appropriate enable output.

The scanners provide the central processor with means for supervising line and trunk on-hook and off-hook conditions, monitoring dial pulses, and observing various points within the system for administrative and maintenance purposes. Each scanner consists of a matrix of current sensing elements (ferrods) and a controller which allows the central processor to interrogate the ferrods. Since the loss of a ferrod affects only one customer or one trunk circuit, the ferrod matrix is not duplicated; since a fault in the controller may affect the entire matrix, the controller is duplicated. In case of network and signal distributors, both controllers can be active in that they can both be working simultaneously. Only one of the scanner controllers is active in any given 11-microsecond cycle. The active controller, selected by the central control using the appropriate enable output, receives the commands from the central control and transmits back to central control the state of the requested 16 ferrods via both of the peripheral answer buses. The central processor switches controllers or peripheral address buses by selection of enable points. A flip-flop in the central control selects the answer bus used by the central control.

The master control center contains the AMA recorders, maintenance teletypewriters, a magnet card writer, trunk and line test panel, and manual controls and displays. The AMA units are duplicated not only for dependability but to allow data storing while changing the magnetic tape. These units have access to both peripheral unit buses. The unit used currently for recording, designated the active unit, is determined by relays in the AMA control circuits. The central control can control these relays via central pulse distributors. The peripheral unit address bus used in any given transfer of data to the unit is determined by the enable output used.

A number of teletypewriters can be connected to No. 1 ESS. These include maintenance, traffic, and service order teletypewriters. The maintenance teletypewriters provide personnel with means for requesting system action and also means for the system to report diagnostic results or results of requested actions. Since this maintenance teletypewriter is the main communication link between the office personnel and the system, two maintenance teletypewriters are provided. One is always located at the master control center; the second may be at the master control center or may be in a remote location.

Master control center equipment other than the AMA units and maintenance teletypewriters is not duplicated. Continuous operation of this equipment is not vital to the system operation; a fault in the trunk and line test panel, for example, may delay the routine testing or restoration of some trunk but has no direct or immediate effect on system operation.

To summarize: unlike a fault in an unduplicated central processor, which would be likely to cause an office failure, a fault in the peripheral system may have a less serious affect. Therefore, only those parts of the peripheral system whose failure would affect a substantial number of customers (e.g., peripheral bus or network controller) are duplicated. Those parts of the peripheral system where failures would affect a single subscriber, (e.g., line ferrod) or where failures would reduce the traffic handling capacity of the system (e.g., network crosspoint) are not duplicated. The bus and controller used in each operation are selected by an enable point. This selection is determined by enable tables in the call stores. This method can be used because the frequency of use of peripheral unit controllers is lower than the frequency of use of stores in the central processor, making the consultation of enable tables possible. This method is more suitable for peripheral units than the route flip-flop and enable code method used in the stores, since it places minimum equipment in each peripheral unit and requires substantially the same equipment in the central controls regardless of the office size. These factors are important since (1) most controllers are directly associated with a relatively small number of lines and trunks and therefore any additional circuits placed in the controllers will reflect strongly in the over-all cost, and (2) the number of peripheral units is, in general, much greater and is more affected by office growth than the number of units in the central processor.

IV. MAINTENANCE CIRCUITS

The No. 1 ESS maintenance plan is implemented in part by circuits and in part by program. Frequently a given function can be performed by either. For such cases, the use of circuits and programs is balanced to yield the most economical solution. In evaluating the cost of a circuit solution, the cost of maintaining the maintenance circuit itself must be included. The over-all cost of the maintenance plan depends to a large extent on finding this correct balance.

The principal functions of maintenance facilities are trouble detection, recovery of an operational configuration after a trouble has been detected, and generation of diagnostic data that the maintenance personnel can use to locate the trouble.

Trouble detection is a function that requires continuous attention, and is therefore much better suited to circuits than programs, so that real time is not used for this function. In some instances, when trouble is detected the best course of action is first to retry the operation under circuit control before program operation is disturbed by an interrupt. Once the circuit has established with some certainty that trouble exists, the system control is transferred, by circuits, to programs that analyze the problem, determine an operational configuration and control the switching to establish the new configurations. Such analysis is logically complex but occurs infrequently. Therefore a program solution is more suitable. In the generation of diagnostic data for pinpointing the trouble, circuits are used to provide intermediate access in addition to normal input-output access, while the program is used to actually perform the tests and to interpret the test results. In diagnostics, the circuit-program balance is particularly important. If adequate test access is not provided, it becomes either costly in terms of program words, or impossible to pinpoint the fault.

Thus, circuits are used for: (1) trouble detection, (2) automatic retrials, (3) administration of coarse program priority, (4) switching of duplicate units, and (5) diagnostic access. Of these functions, trouble detection, retrials, and diagnostic access are the subjects of this section. The administration of coarse program priority, i.e., the transfer of program control by interrupting current programs, will be discussed in Section V. The switching of duplicate units was discussed in Section III.

4.1 *Central Control*[2,3]

Since the central control is a data processor which continuously manipulates and modifies data, it is difficult to incorporate self-checking features within a single central control. The concepts of parity checks and other checks which can be provided for transmission or storage units are not practical for a unit such as the central control.

The primary tools incorporated for detecting and diagnosing troubles within the central controls are match circuits capable of comparing a number of internal nodes.

To match information between central controls, each central control transmits information describing its internal state to the duplicate central control. The match information is buffered in match registers, and the registers are then compared. Each match operation compares 24 bits in parallel (i.e., one word of data). Each complete match operation requires nearly one 5.5-μsec machine cycle. In order to match two words per machine cycle, two match circuits are provided in each central con-

trol. Fig. 8 illustrates the matching scheme. As shown, each matcher has access to 6 internal points, providing access to 12 points or a total of 288 internal leads. These intermediate access points partition the central control into a number of smaller circuit blocks. The ability to match the internal masked and unmasked bus[2] provides indirect access to all of the index registers. Each match point can be sampled at any one of three times during a machine cycle.

The match circuits are designed to operate in five different modes. The modes of operation and their uses are:

(1) Routine match mode — In this mode, two points are matched each cycle, one in match circuit zero and one in match circuit one. In this mode a match is always expected. If a mismatch is detected, special

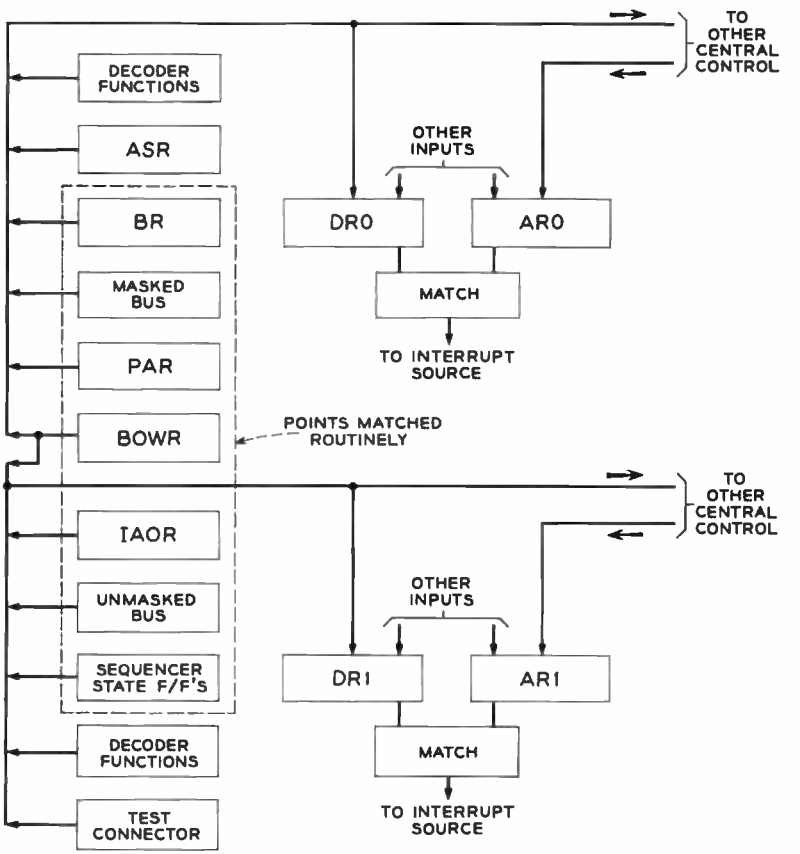


Fig. 8 — Central control matching access.

actions are initiated. The special actions are optional and are specified when the mode is established. The options available (in any combination) are:

(a) to stop matching and save the contents of the match registers,
(b) to generate a maintenance interrupt,
(c) to stop the standby central control.

These options are also available for all the modes to be described below. In the normal use of the routine match mode, options (a) and (b) are used.

In the routine match mode, the points matched each cycle are dependent upon the order being executed. On "write" orders, the index adder output register (IAOR) and buffer register (BR) are matched. This compares the address and data generated on the write order. On "read" or "peripheral" orders, the IAOR and the masked bus (MB) are matched. This again compares addresses and data. After a reread (see Section 4.2) or error correction of a program store word, the buffer order word register (BOWR) is matched. On transfers, the program address register (PAR) and unmasked bus (UB) are matched. If none of these special operations is being performed, the program address register, the unmasked bus, the masked bus, and a number of sequencer points are matched in a cyclic manner.

The routine match mode is the normal trouble detection mode for the central control matchers. The two central controls operate in parallel, and as long as they continue to do so, the matchers always detect a match condition. If a mismatch occurs, an interrupt causing a transfer to a fault recognition program occurs.

(2) Directed match mode — In this mode, the points matched and the time of the match in each matcher are determined by the contents of the match control register. This register is set up under program control when the mode is established. In this mode, the specified points are matched once every cycle. The interrupt and other options are also available. This mode is used to determine whether a mismatch occurs at a specified match point during the execution of a specific program.

(3) Sampled match — The sampled match mode is used in conjunction with a mismatch sampling order (EMMS) to perform one sample per matcher at a specified point in a program. During this mode no matching occurs until the EMMS order is executed. The EMMS order specifies the points to be matched, the time of the match, and the number of machine cycles following the EMMS order on which the match is to occur. The information sampled by the matchers is retained in the match registers for examination. This mode is used in the central control

diagnostic program to obtain critical information during the execution of test programs.

(4) Breakpoint mode — In this mode, the active central control can monitor program store addresses, call store addresses, or both, of the standby central control and perform any of the match options when the standby reaches any prespecified program store or call store address. This mode is useful in controlling off-line testing of the standby central control. The standby system can be set up to execute a program independently of the active program to stop the standby system, and to interrupt the active program when the standby program reaches a preselected program address.

(5) Preset match — As an extension of the breakpoint mode, it is possible to sample any of the match points of the standby and to compare the point with a prespecified number in the match registers of the active. In this mode, if a match is detected, any of the available options can be executed.

There are additional circuits in the central control for detecting troubles, providing diagnostic access, and providing automatic retrials. Some of these are designed primarily for detecting troubles in units external to the central control. These will be described in the following sections. The remaining circuits provided for central control maintenance are:

(1) Inputs to the emergency-action circuit — Circuits are provided to detect catastrophic failures, such as a faulty clock, loss of power, or locked-up sequencers. If these circuits detect a failure, they signal the emergency-action circuit. The actions performed by the emergency-action circuit will be described in Section 5.6.

(2) Voltage regulator monitoring circuits — When voltages of 4.5 volts are required in the central control, regulators[3] are provided to drop the 24-volt supply to this level. Monitoring circuits are provided to check these voltage regulators. These monitors are connected to scan points for diagnostic testing.

(3) Start-stop and control word write — The active central control has the ability to stop and start the standby central control. It can do this automatically as a function of the matching mode, or under program control. This is accomplished by a sequencer which inhibits the standby decoders and sequencers. The stop action allows the sequencers to complete so that the standby central control can be started at some later time without mutilating the program actions.

While the standby central control is stopped, the active central control can insert data into a limited number of registers and control flip-flops of the standby central control. This feature provides additional input

access for diagnostic testing. The active central control can write into the match control registers of the standby central control and establish match modes without depending upon the standby central control to execute an instruction. The ability to control write into the program address register of the standby provides a simple way for the active central control to force the standby central control into step.

4.2 *Program Store*[10]

Words stored in the program store include seven Hamming error detecting and correcting bits. These bits are computed over the word stored and the 16-bit address. On each reading, the central control performs a check over the address sent and the word received. The central control can detect and correct single errors in the received word, detect double errors in the received word, and detect double or single errors in the address. Single errors in the word are corrected by circuit means at the cost of one 5.5-microsecond cycle but without interrupting the program. When an error is corrected, an error counter in the central control is incremented. This error counter can be read under program control. Detection of a single error in the address or a double error in the word or the address will cause an automatic reread. A failure of the reread will cause an interrupt of the current program and a transfer to the program store fault-recognition program. Rereads will also cause an error counter to be incremented in the central control.

The information block name of the store[10] is transmitted in a two-out-of-four code. The store checks the validity of the code that it receives. Within the store, internal timing checks, waveform checks, and one-out-of-n checks are made. If all these checks pass, the store transmits an all-seems-well pulse back to the central control together with the 44-bit word. If the central control fails to receive the all-seems-well pulse, it will reread. Failure of the reread will cause an interrupt and a transfer to a maintenance program.

Table I summarizes the various maintenance features and circuits in the program stores and in the program store-central control communication facilities. A maintenance read mode allows the central control to read the state of various points within the program store address circuits. The maintenance write mode allows the central control to change the state of the route flip-flops and other control flip-flops within the stores. Both of these modes use the existing communication buses. In addition, the central controls have access to some critical program store test points via the scanners and input access to control flip-flops via the central pulse distributors. Each store has a number of multicontact relays which,

TABLE I — MAINTENANCE CIRCUIT FEATURES OF PROGRAM STORE-CENTRAL CONTROL COMMUNITY

(1)	Split duplication
(2)	Common duplicate ac bus system
(3)	Enabling by name code
(4)	High-speed switching by route and trouble flip-flops
(5)	Hamming code over word and address
(6)	Name in 2/4 code (2–6 stores)
(7)	"All-seems-well" pulse
(8)	Automatic reread and single-error correction
(9)	Error counters in central controls
(10)	Matching of words between central controls
(11)	Continuous use of standby equipment
(12)	Maintenance read and write operations
(13)	Access to program stores via scanners and central pulse distributors
(14)	Monitor bus
(15)	Automatic "shut-up."

when operated, connect a number of internal points of the store to the monitor bus which terminates in a scanner. This allows additional access to the store for diagnostic purposes. The automatic "shut-up" isolates the store from the communication bus in the event of a failure which might cause the store which has failed to transmit falsely on the bus.

4.3 *Call Store*[12]

The interconnecting, duplication and switching plan used for the call store-central control community is similar to that used for the program store-central control community. The main differences between the two plans are italicized in Table II. One difference is the size of the community. The call store community may contain as many as 37 call stores. Another difference is in the information coding. Instead of a Hamming

TABLE II — MAINTENANCE CIRCUIT FEATURES OF CALL STORE-CENTRAL CONTROL COMMUNITY

(1)	Split duplication
(2)	Common duplicate ac bus system
(3)	Enabling by name code *(2–36 stores)*
(4)	High-speed switching by route flip-flops
(5)	*Parity bit over word, name, and address*
(6)	*Parity bit over address*
(7)	"All-seems-well" pulse
(8)	Automatic reread
(9)	Error counter in central controls
(10)	Matching of words between central controls
(11)	Continuous use of standby units
(12)	Maintenance read and write operations
(13)	Access to call stores via scanners and central pulse distributors
(14)	Monitor bus
(15)	Automatic "shut-up."

code, there are two simple parity checks. One parity bit is over the address, including the store name, and the word. This bit is stored with the word. It is checked in the central control each time a word is read from the call store. The other parity bit is over the address only. This bit is checked by the store each time it accepts an address from the bus. A failure of this check, as well as any failure of the internal store checks, inhibits the all-seems-well pulse. A failure to receive an all-seems-well pulse or a failure of the over-all parity check will cause the central control to reread or rewrite and to increment an error counter. A reread or rewrite failure will cause an interrupt of the current program and a transfer to the call store fault-recognition program.

4.4 *Central Pulse Distributor*[7]

The central pulse distributor, a high-speed electronic translator, provides two types of outputs in response to commands from the central control. The first output is a bipolar pulse used to control trunk circuits and special control circuits located throughout the various subsystems. The second output is a unipolar pulse used to control certain trunk circuits and to enable peripheral subsystems such as the scanner controllers, signal distributor controllers, network controllers, etc. The unipolar enable pulse is used to select one out of n subsystems connected to a common address bus.

The receiving circuit returns a one-out-of-n acknowledgment signal to the central pulse distributor by way of the same twisted pair over which it received the unipolar pulse. The central pulse distributor translates the one-out-of-n code into the central pulse distributor address form (three one-out-of-8 codes) and returns this address to the central control, where it is matched against the original address.

Within the central pulse distributor, internal checks monitor the operation of the pulse receivers and check that one and only one pulse was received and acted upon from each of the three one-out-of-8 codes. In addition, an internal check is made to assure that current used to drive the matrix is within prescribed limits. If all checks pass, a signal labeled the "all-seems-well pulse" is returned to the central control. A check failure inhibits action on the command (thereby preventing false operations of the central pulse distributor) and inhibits the all-seems-well signal.

The private execute signal, which selects one of a plurality of central pulse distributors, is returned to the central control for verification. This check assures that the correct central pulse distributor, and only the correct one, received and acted upon the address.

Failure of any of the checks causes the central control interrupt sequencer to transfer program control to the peripheral unit fault-recognition programs.

As described in Section III, the central pulse distributors are provided in pairs. One central pulse distributor may be taken out of service by means of a signal from the mate central pulse distributor. This signal controls a flip-flop which will remove power from the execute signal circuit and from circuits which send the verify signals and the all-seems-well signals. This action prevents false generation of outputs. This out-of-service mode is called "quarantine."

To provide diagnostic access to the central pulse distributor, 64 simulated matrix loads are provided. Addresses with good or bad parity can be supplied by the central control to test the final matrix current sampling circuits and access circuitry. The combination of both the simulated matrix load and the trouble detection circuits allows the controller and the common matrix to be diagnosed without placing unnecessary restrictions on the assignments of central pulse distributor points and without generating harmful outputs.

4.5 *Network and Signal Distributor*[6,7,11]

When a network or signal distributor receives a command, a number of dynamic checks are made within the controllers. These include: a check that one and only one path selection relay in each one-out-of-n code group is operated; a check on the continuity of the path through the ferreed control windings; and a check of the amplitude of the pulse which operates the ferreed.

If any of the internal checks fail, the controller sequencing is halted and the controller remains in the trouble state. Each controller is assigned three scan points. The presence of trouble in a controller is detected by program interrogation of these scan points. The controller can be reset by a special common reset command broadcast by the central control.

To prevent erroneous network actions caused by a faulty controller, the mate controller, on command by the central control, can place the faulty controller into a "quarantine" mode. While in this mode, the faulty controller can receive addresses and orders but is prevented from gaining access to the ferreeds.

Diagnostic access to the controllers is provided by a diagnostic bus. Test points are located at key locations within a controller. These test points can be connected to the diagnostic bus by relays operated via the mate controller.

The switching matrix and the signal distributor output matrix include relatively few circuit checks. Most troubles in these circuits are detected by maintenance programs which are integrated into the call programs.

When a path is to be established through the line link network, a partial path is first set up and tested using a detector circuit that is connected to the path momentarily in the fourth switching stage. This detector, labeled "false cross and ground detector," can detect shorts between the tip and ring conductors and the presence of ground or any potential on either conductor.

The customer dial pulse receivers include a detector for foreign voltages. When a line is first connected through the network to a dial pulse receiver, this detector is read via a scan point.

Some of the more complex trunk and service circuits[13] include some check features. For example, seven scan points are connected to multifrequency transmitters. Some states of these scan points indicate trouble conditions.

The signal distributor contains a circuit which checks whether the selected output relay actually changes its state. A check failure, i.e., no change in state, is reflected in the state of three scan points connected to the controller.

4.6 *Scanner*[7]

The scanners are addressed by the central control via the peripheral unit address bus. The scanners receive commands in groups of one-out-of-n codes. Like other peripheral units, the scanner, upon receiving an enable pulse, returns a verify signal to the central pulse distributor.

Within a scanner controller, the matrix current is sampled and a check is made that one and only one row in the scanner matrix has been selected. The results of these checks are sent back to the central control together with the states of the interrogated ferrods. The check signal is labeled an "all-seems-well scanner pulse." A failure inhibits the all-seems-well pulse, thereby alerting the central control that a trouble has been detected. The central control responds by generating a program interrupt, causing a program transfer to a fault-recognition program.

For purposes of diagnosis, a special maintenance command is provided. This command, broadcast together with an appropriate enable pulse, causes the scanner detector circuits to produce a known output which is sent back to the central control.

4.7 *Alarms and the Master Control Center*[8]

Communication between the No. 1 ESS and the maintenance personnel is achieved by the following means:

(a) an office alarm system,

(b) local alarm circuits, display lamps, and power removal switches at the individual system units, and

(c) a teletypewriter, visual displays, and manual controls at the master control center.

The local alarm system consists of detecting and indicating circuits located in the individual equipment frames. The office alarm system consists of aisle pilot lamps, main aisle lamps, exit lamps and audible alarms.

The main power equipment does not require a series of locating lamps, since it has its own alarm circuit providing both major and minor alarms. Any failure in the central processor or failures of both controllers in a peripheral subsystem will cause a major alarm. Other failures in peripheral subsystems will cause a minor alarm. Failure of a trunk in a group of trunks, of per-line equipment, or of network cross points will be reported via teletypewriter without an audible alarm.

Whenever trouble is detected by a local alarm circuit, the office alarm system alerts the maintenance man; pilot lamps direct him to the faulty equipment unit. Most troubles, however, are detected by the system under program control. In this case, the office alarm system directs the maintenance man to the master control center.

For a locally detected trouble, the alarm is retired at the faulty unit by removing power. For a system-detected trouble, the alarm is retired at the master control center, where a diagnostic printout is given at the teletypewriter. Using this printout, the maintenance man consults a dictionary to obtain the location of the fault.

Teletypewriters are used by the operating personnel to make recent changes of translation information, to request status reports, etc. The system, in turn, uses teletypewriters to print out test results, traffic information, permanent signal conditions, etc.

Additional facilities are provided at the master control center for storing AMA information on magnetic tapes, for updating the translation information contained in the program stores, and for testing lines and trunks. Thus the master control center represents the maintenance and administration center of the office.

V. MAINTENANCE PROGRAMS

As described in Section IV, circuits are, in general, used for functions such as trouble detection and diagnostic access. On the other hand, the determination of an operational configuration, the control of switching to establish such a configuration, the pinpointing of faults, the conduct-

ing of tests, and recording and interpreting the test results are implemented by program.

The maintenance programs for the No. 1 ESS can be divided into three categories:

(1) fault-recognition programs and emergency-action programs which determine and switch into service an operational system after a trouble has been detected;

(2) diagnostic programs which conduct tests on a faulty unit to pinpoint the faulty circuit pack(s); and

(3) exercise programs which supplement the hardware trouble detection facilities.

The vast majority of these programs are executed on-line* as opposed to off-line. Some programs can be executed off-line upon request by the maintenance personnel. This facility will be discussed in Section 5.8.

As general design objectives, all of the maintenance programs should be:

(1) generic — the same programs should be applicable to all offices. Parameters may be used to specify items such as the number of units in a given office.

(2) uniform — both in their relationship to maintenance personnel and other programs

(3) simple — for easy design and understanding

(4) noninterfering — the maintenance programs should interfere with call processing only when absolutely necessary.

Before embarking on a discussion of the three categories of maintenance programs, a brief review of the call program operation is included so that the maintenance-call program relationship can be established.

5.1 *Review of Call Program Operation*[14,15]

All system programs are placed into a hierarchy according to their urgency. This hierarchy has many grades or steps. The programs in the upper part of the hierarchic ladder are called "nondeferrable" programs; programs in the lower part are called "deferrable." The nondeferrable programs are either programs that must be performed in order to secure the system call processing ability, which may be in jeopardy, or programs which are associated with the system's input-output functions. The input-output programs must be performed punctually and usually

* Many of the programs are executed by both the active and standby data processor. For the purposes of this paper, the distinction between on-line and off-line is the following: an off-line program is a special program which is executed by the standby data procesor while the active data processor executes its normal call programs; all other modes of program execution are termed on-line.

repetitively — otherwise data, e.g., dial pulses, will be lost. The deferrable programs process data already in the system and are not, therefore, as critically synchronized to real time as the nondeferrable programs. Dial pulse scanning, for instance, is a nondeferrable function. Tasks such as network path hunts and processing of traffic data are carried out by deferrable programs.

A control program called the "base-level main program" administers the priority of the deferrable programs. For example, scanning for new originations is given preference over collecting traffic data. In effect, all deferrable programs are placed in one of five work lists of the main program. The main program administers priority by initiating the programs on different lists at different frequencies. For example, jobs on List A are initiated 16 times for each time that programs on List E are initiated.

The main program and the programs it initiates are not "aware" of the passage of real time. To insure punctual performance of the nondeferrable programs without impairing the efficiency of the deferrable programs, an interrupt facility was designed. Every 5 milliseconds, a clock output triggers the interrupt sequencing circuit in the central controls. The interrupt sequencer allows the handling of the current instruction to be completed, inhibits the work on the next instruction, stores in a reserved area of the call store the address of the next instruction and the contents of some other flip-flops, and then transfers to what is called the "interrupt" program (see Fig. 9). The interrupt program stores the contents of the remainder of the central control registers in the call store and then transfers to the interrupt level main program, which initiates the various nondeferrable programs. When these programs have been completed, the central control registers, which at the time of the interrupt were stored in the call store under program control, are restored under program control. A special instruction is then executed which fetches the program store address which was about to be used at the time the interrupt occurred and transfers to the interrupted nondeferrable program at the base level. Thus the interrupt mechanism provides a means for executing the nondeferrable jobs without disturbing the deferrable programs. Under normal conditions, the system operates either on this interrupt level or on the base level. The coarse priority is determined by the circuits, i.e., the 5-millisecond clock and the interrupt sequencers. The finer steps in the priority are administered by the program, i.e., the main program on each interrupt level.

The nondeferrable jobs, in turn, are of two categories: high-priority and low-priority. High-priority programs, such as dial pulse scanning, *must* be performed punctually or information may be lost; low-priority nondeferrable programs, such as sending commands to the network, *should*

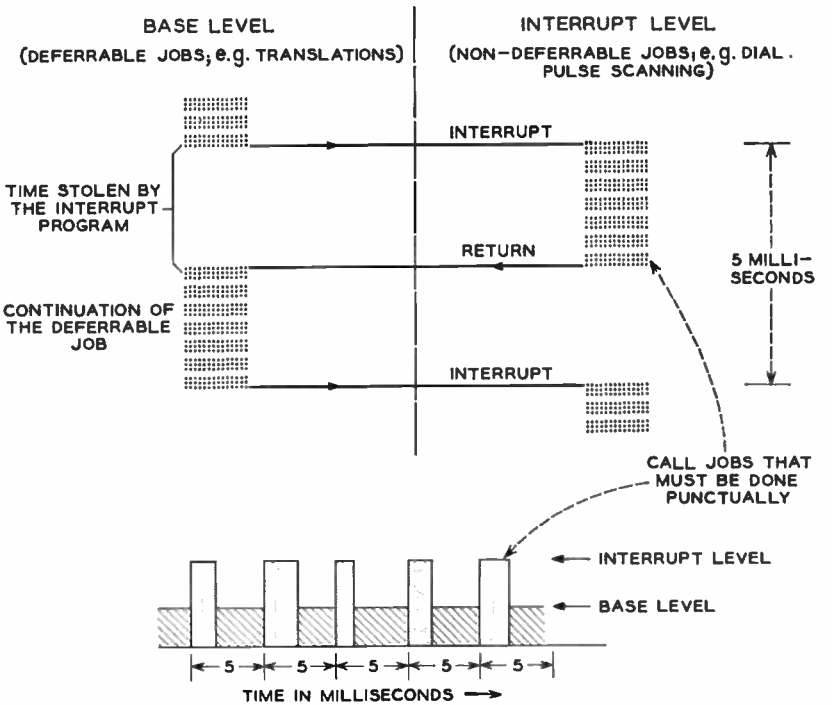


Fig. 9 — Call program operation (1).

be performed punctually. Occasionally, nondeferrable work may not be completed within a 5-millisecond interval. To insure the punctual performance of the high-priority work, a second interrupt level is provided. This interrupt, the H-level interrupt (the first interrupt described above is the J level), will occur if at the end of a 5-millisecond period the high-priority nondeferrable work has been completed but the low-priority has not (see Fig. 10). The H-level interrupt is implemented in much the same way as the J-level interrupt. The interrupt sequencer transfers the program control to the H-level main program, and after carrying out the high-level programs, returns the control to the J-level programs.

As long as the system is trouble-free, it operates in the base level and at H–J interrupt levels as determined by the interrupt hardware.

5.2 *Fault-Recognition Programs*

When circuit troubles occur and are detected by fault-detection circuits, program control must be transferred to programs that recover

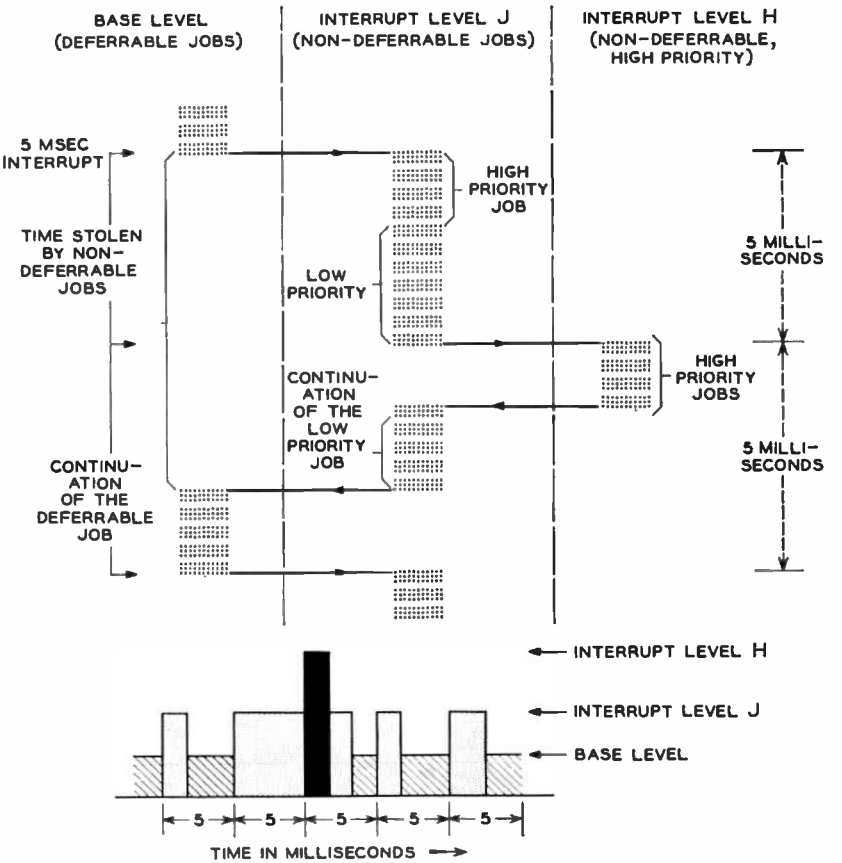


Fig. 10 — Call program operation (2).

the call processing ability of the system. These programs are called "fault-recognition" programs. The transfer of program control is implemented by the central control interrupt circuits. In call program operation, this mechanism is activated by the 5-millisecond clock and is used to interleave the three major priority classes in the call program hierarchy (high-priority nondeferrable, low-priority nondeferrable, and deferrable). In maintenance program operation, the interrupt circuit is in general activated by fault-detection circuits.

There are a total of ten interrupt levels, designated level A, level B, $\cdots$, level K (I omitted) in descending order of priority. Levels A through G are associated with the master control center and with the fault-detection circuits listed in Fig. 11. The base-level programs may be inter-

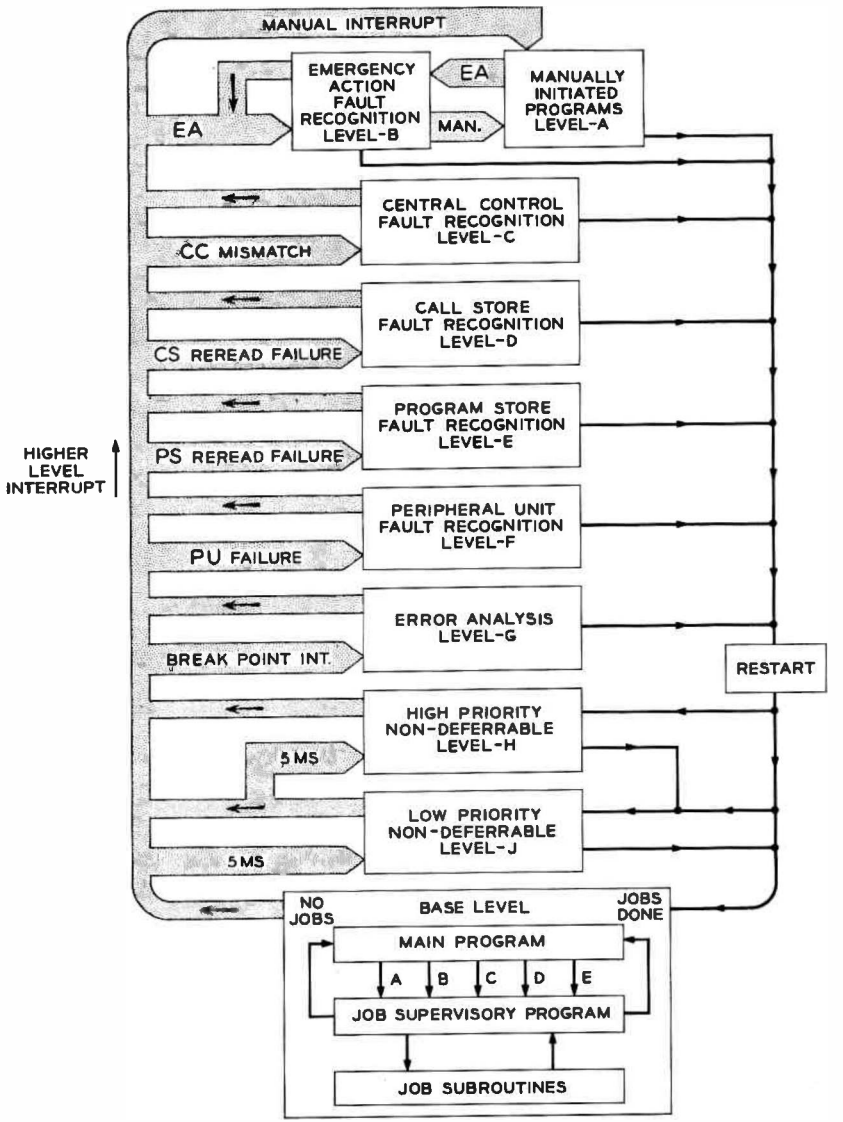


Fig. 11 — Interrupt levels and priorities.

rupted in the manner described previously to begin the execution of one of the ten interrupt programs. Once one of these programs is initiated, it in turn may be interrupted to permit performance of higher-level interrupt functions. However, with the exception of levels A and B, a given interrupt program may not be interrupted to perform a function at the same or lower level. The high-priority programs are assigned to the levels in the interrupt hierarchy according to the relative urgency of the action to be taken.

In order of descending urgency, there are five broad classes of programs:

(1) Programs that recover the system's data processing ability; these are programs operating on interrupt levels A through E.

(2) Programs that recover proper operation of the peripheral system; these are interrupt level F programs.

(3) Programs that handle inputs and outputs; these programs operate on interrupt levels H and J interrupt programs.

(4) Special test programs that operate on interrupt levels G and K.

(5) Programs that process information within the system; these are the base-level programs.

Thus a mismatch between central controls generates a C-level interrupt, whereas detection of trouble in a peripheral system generates a level-F interrupt. In case of a central control mismatch, the data processing ability of the entire system is in jeopardy; in case of a peripheral unit malfunction the data processing ability is intact, although service to some or all subscribers may be affected.

In Section 5.5, the fault-recognition programs associated with levels C, D and E are discussed in detail. This section will describe the design considerations and general characteristics common to all fault-recognition programs.

The function of the fault-recognition programs is to recover the data and/or call processing ability of the system. The design objectives in decreasing order of importance are:

(1) *The data processing ability must be recovered as long as a sufficient number of fault-free subsystems exist to form an operational configuration.*

(2) *The malfunction that caused the entry to the fault-recognition program should not be allowed to interfere with the calls being processed.* Such interference could occur in a number of ways: (a) the call being handled in the central processor at the time of interrupt might be mutilated; (b) the malfunction may cause mutilation of temporary memory or alter the state of network connections in such a way that calls (both current and future) are affected; (c) the call processing ability may be lost for a

sufficient time to cause input-output information to be lost. For example, if the call store fault-recognition program that operates on interrupt level D takes 50 milliseconds to recover the system, no dial pulse scanning will take place during this interval and dial pulses from lines in the process of dialing may be lost.

(3) *The faulty subsystem should be located and switched out of service.* In some situations, it may be possible to recover an operational system without isolating the faulty subsystem. For example, it may be known that a fault exists in either a program store, a program store bus, or a central control. By switching out all these units, an operational active configuration can be established. The isolation of the faulty subsystem is also considered a function of the fault-recognition program.

(4) *The fault-recognition program should be able to distinguish between errors and faults.* An error is defined as a malfunction, the symptoms of which cannot be reproduced under program control. A fault is defined as a malfunction, the symptoms of which the program can reproduce at will. If the fault-recognition program determines that an error caused the interrupt, it will make a record of the interrupt. These records are utilized by error analysis programs to recognize abnormally high error rates and to determine the cause of such error rates.

It is not feasible to design a fault-recognition program which will always recover an operational system, isolate the faulty subsystem, and separate errors from faults rapidly enough so that there is no interference to call processing. For example, if the level-D fault-recognition program were to check thoroughly all call stores in a large system before returning the system to call processing, many calls might be lost, since such a check would take several hundred milliseconds. Therefore the fault-recognition programs were designed to minimize the *average* recovery time by doing the following:

(1) The programs are designed to recover an operational system rapidly (within 5 milliseconds) from the great majority of interrupts, i.e., from interrupts caused by errors and most faults. (Errors are assumed to be far more prevalent than faults.)

(2) Where the recovery is not simple or where subsequent interrupts indicate that the first attempt to recover was not successful, whatever time is necessary to recover an operational system will be taken.

(3) To expedite the return to call processing once an operational configuration is recovered, the isolation of the faulty subsystem and the analysis of errors are postponed and initiated later as a base-level program.

The fault-recognition programs can be divided into a main program, tests, service programs, and a restart program. The main program con-

trols the general course of action taken. The tests, as the name indicates, are programmed questioning of circuits to determine whether or not the circuits respond properly. Service routines include programs such as a program store configuration change routine, which calculates and establishes a program store configuration that fulfils constraints given as inputs to the routine. The restart program determines at which program point data and call processing should resume, restores the memory and central controls to the appropriate state, and returns the program control to the interrupted level.

5.3 *Diagnostic Programs*

The function of diagnostic programs is to generate test data to isolate the fault to a small number of plug-in circuit packs within the subsystem that has been taken out of service by a fault-recognition program.

Typically, a diagnostic program carries out a fixed sequence of tests. These tests are performed by observing the normal outputs of a unit or monitoring some special test points strategically located in the unit. The test points may be observed via a scanner, via normal communication routes (such as used by the control read operation of the stores), or via special communication buses (such as the match buses of the central controls). The test results are recorded in the call store and then printed out via a maintenance teletypewriter. The combinational pattern of which tests passed and which tests failed defines for the maintenance man the circuit pack(s) to be replaced. The translation from test results to the faulty circuit is done with the aid of a maintenance dictionary. The techniques employed to derive the dictionaries will be described later.

The diagnostic programs are normally requested by the fault-recognition programs at a time when an operational configuration already exists. The fault-recognition programs are high in program hierarchy, and hence the length of the fault-recognition programs adds directly to the system down time. The diagnostic programs, on the other hand, add only to the repair time.

Repair time is defined as the interval of time from the occurrence of a fault to its repair. The repair time affects both the dependability and maintainability of the system: when the repair time increases, the probability of the mate unit failing goes up. The repair time includes the time to detect the fault, recover an operational system, inform the personnel, get someone into the office, analyze and repair the fault. In this chain of events, the actions taken by personnel, particularly in an unattended office, may stretch to hours. Thus the diagnostic programs can be and

are assigned to the lowest step in the ladder of program hierarchy without substantially affecting the repair time.

The length of a diagnostic program may vary from a few milliseconds to several hundred milliseconds, depending on the unit being diagnosed. To prevent the diagnostic programs from interfering with call processing, they are divided into 10-millisecond segments. When a fault-recognition program discovers a faulty unit and switches it out of service, it also records the incident in the call store memory reserved for maintenance programs. This memory area is labeled the "maintenance control register." A program called the "maintenance control program" administers and controls this register.

Periodically, the base-level main program calls in the maintenance control program, which in turn discovers the need for the diagnostic action and initiates the first segment of the diagnostic program (see Fig. 12). At the end of a segment, the diagnostic program returns the

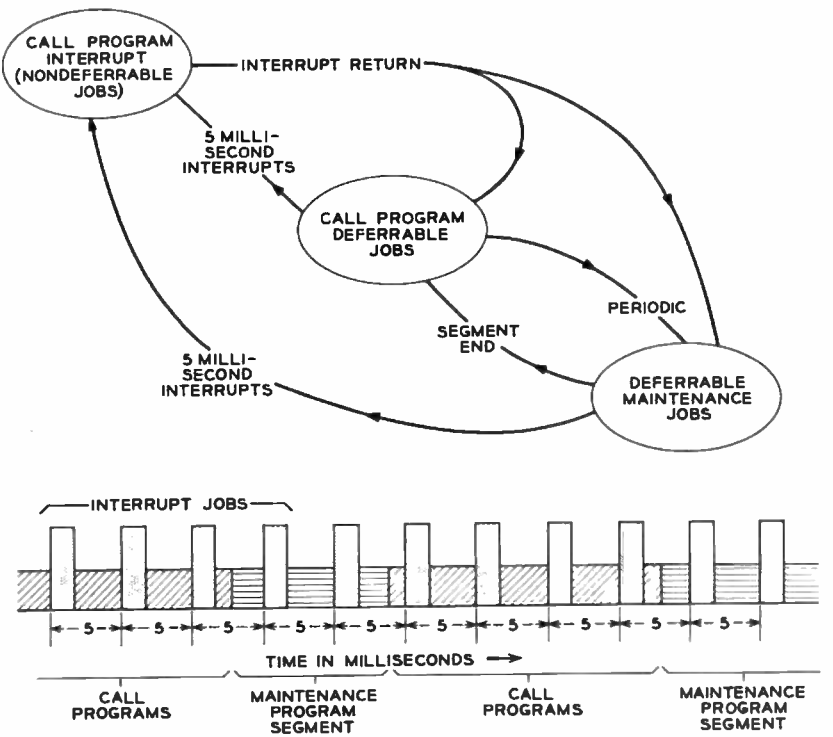


Fig. 12 — Maintenance-call processing interleaving.

control to the maintenance control program, which in turn returns to the base-level main program. On the subsequent main program visits to the maintenance control program, the other segments of the diagnostic program are initiated until the diagnostic program is completed. Each of the diagnostic program segments is interrupted by the J and possibly higher-level interrupts.

The maintenance control program also administers priority among diagnostic programs (central processor diagnostic programs are given higher priority than the peripheral unit programs), insures that no program holds the maintenance control register for more than 10 minutes, and performs tasks that are common to many of the maintenance programs (tasks such as timing, recording error information, control of common maintenance facilities, etc.).

There are a number of problems common to all diagnostic programs. One problem is that of ensuring consistent test results for the same fault every time that fault occurs. One possible cause of inconsistent results is intermittent troubles, i.e., troubles which cause a system failure at some times but not at others. However, if the diagnostic programs are repeated a number of times, it is likely that during some pass of the program the fault will become apparent. By repeatedly performing the diagnostic programs, intermittent faults should be located.

Initial conditions can also affect the consistency of diagnostic results. A fault can occur randomly in time, with the system memory elements in any possible state. The diagnostic tests must be so designed that the same test results are obtained (for a given fault) regardless of the state of the memory elements at the time the fault occurs. This implies that a unit under test must be properly initialized. This problem also affects the choice of the information to be recorded as pertinent test results, and it affects the order in which tests must be performed.

The effect on the recording of test results can best be illustrated by an example. Assume that we are attempting to test a flip-flop controlled by a single input gate. One "obvious" approach would be to write a zero into the flip-flop and record one bit of information indicating whether or not the flip-flop read a zero after the write operation. Next, one could write a one in the flip-flop and record an additional bit indicating whether or not this was successful. However, if this is done, inconsistent results will be obtained for certain types of troubles. Assume a fault in which the gating function controlling the inputs to the flip-flop is inoperative, so that it is impossible to write anything into the flip-flop. Then the information in the flip-flop is variable, depending upon the state of the flip-flop when the fault occurred. If at one occurrence of the

fault a zero was in the flip-flop, the zero test would pass, but the one test would fail. If at another time a one was in the flip-flop when a fault occurred, the zero test would fail, but the one test would pass. If both of these results are recorded as independent test results, then the results may differ from one fault occurrence to another. This problem can be avoided if only one result is recorded for the above two tests. This result should be the union of the results of the zero test and the one test.

This same problem affects the order in which tests must be performed on a unit. To test a unit, one must apply test inputs to the unit and observe the outputs. As described in a previous section, for nearly all units test points are provided to give input-output access in addition to the normal input-output access. Even so, it is not possible to provide independent input-output access to all circuits within each unit. Consequently, in testing a specific circuit it is sometimes necessary to use another circuit within that unit as an input or output device. However, if this circuit, used as a tool to test another, contains memory elements which may suffer from the type of fault described above, inconsistent results may occur if the problem is not handled properly.

Generally, a circuit is tested before that circuit is used to test another; the diagnosis is terminated if a failure is found within the original circuit. In most cases this can be done with little loss in fault resolution. Where this cannot be done, one must design the test in such a way that the results obtained are independent of the memory state of the circuit.

Normally, when a diagnosis is requested after a fault is discovered by some program, a complete diagnosis is performed, and the results are printed out in a convenient reduced form. In addition, the maintenance man will be able to request a diagnosis from the system teletypewriter. He will be able to request a complete diagnosis, in which the results are printed out in reduced or in unprocessed form, or he can request that only certain parts or phases of the diagnosis be performed.

A diagnostic program is automatically initiated whenever power is restored to a unit which previously had power removed. When a unit is being repaired, power is normally removed. When power is restored, a diagnostic program will be automatically requested and, if it passes, the unit will be restored to service. If a failure is detected, the unit will be left out of service and the test results printed out on the teletypewriter.

5.4 *Exercise Programs*

The third category of maintenance programs consists of exercise programs which exist for the following reasons:

(a) *Supplementing trouble-detection facilities:* test calls, for instance, are initiated periodically to detect troubles that might otherwise go undetected.

(b) *Searching for uncorrected errors:* some programs, for example, look for discrepancies between the network hardware and the network map stored in call store.

(c) *Checking trouble detection circuits:* mismatches, for instance, are intentionally introduced to check the response of the system.

(d) *Exercising infrequently used hardware:* the program store configuration, for example, is periodically changed to ascertain that it can be changed when needed.

The exercise programs may be initiated automatically and periodically by the system. They may also be initiated on demand by other programs or by maintenance personnel.

The exercise programs, like diagnostic programs, are of low priority and operate on the base program level under the control of the maintenance control programs. As with the diagnostic programs, the prime design consideration is to minimize the program length.

5.5 *Implementation of Fault-Recognition Programs*

In the interests of brevity the following discussion is limited to those programs concerned with maintaining the central processor.

5.5.1 *Central Control Fault Recognition*

The match circuits of the central control are the primary tools for detecting central control troubles. These circuits normally operate in the routine match mode discussed previously. When the system operates in this mode, a mismatch results in a level-C interrupt source being set. Provided that no higher interrupt level is active, the setting of this source results in a C-level interrupt program being entered by the interrupt sequencer. The C-level interrupt program is the central control fault-recognition program.

When this program is entered, the only fact that is readily apparent is that there has been some disagreement between the two central controls. This disagreement may have been caused by a random error which affected one of the central controls, by a fault in the active central control, a fault in the standby central control, or by a fault in some external unit which affected only one central control. The basic function of this program is to determine which of these possibilities exists. If it is determined that the mismatch was caused by an error, the two central

controls are put back into step and routine matching restored. If it is determined that one of the units is faulty, that unit is removed from service and the appropriate diagnostic program is requested.

As will be described later, when troubles are detected in some units external to the central control (such as the stores), information (such as an address) is often saved within the central control. This aids the fault recognition program in locating the suspect unit. This is not true in the case of the central controls, however. Since the matching is not instantaneous, nor are all internal points matched continuously, and since trouble can occur randomly within any program, no information is available when the central control fault recognition is entered to give any indication as to which central control contains incorrect information. Information is available within the match control register (MACR) and mode control register (MOCR) of the central control which defines the internal point where the mismatch was detected. However, this information gives no prior knowledge as to which central control contains incorrect data.

The central control fault-recognition program strives to determine which unit (if either) is faulty by attempting to reproduce the trouble symptoms under controlled conditions by logically exercising the central control hardware. If the trouble symptom cannot be reproduced, the trouble is classified as an error and the central controls are returned to parallel data processing. If the trouble symptom is reproduced, the faulty central control is removed from service.

Fig. 13 shows the basic program actions performed by the central control fault recognition program. First the central controls are forced into step and a directed match mode of the program address register (PAR) and index adder output register (IAOR) is established in both central controls. In this mode, the PAR and IAOR are matched once per machine cycle, and if a mismatch occurs at any time, this fact is retained in central control memory.

With these conditions established, the testing begins. The tests are divided into routines which exercise specific hardware areas of the central control (for example, the index adder and its associated registers). These test routines consist of data manipulating operations which expect to find known answers if all operations are performed successfully. The expected answers are checked using conditional transfer orders. For example, a very simple test of the accumulator adder would be to add zero to zero and transfer to a failure routine if the answer is not zero.

If all conditional transfers pass, the program checks to see if the two central controls are still in step by examining whether a mismatch has

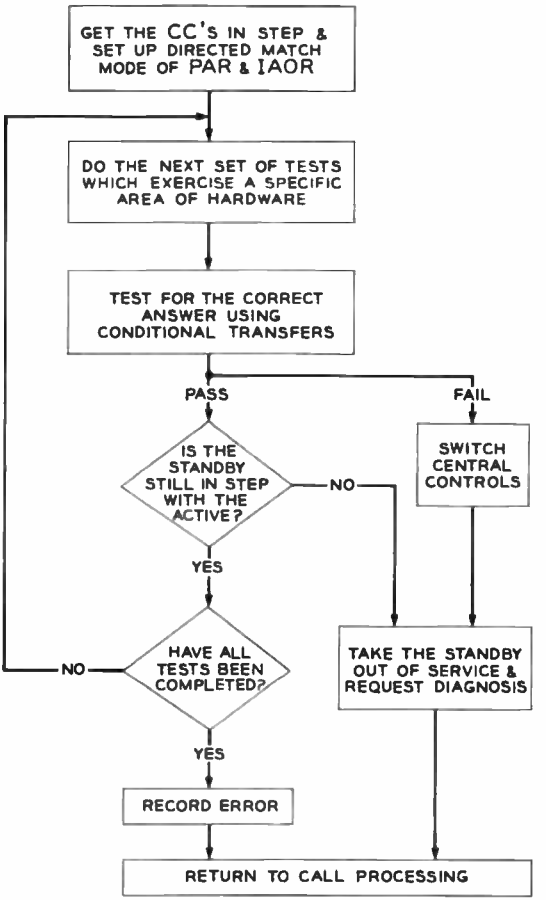


Fig. 13 — Central control fault-recognition test mode.

occurred somewhere in the test program. If they are in step, the testing continues until all tests have been completed successfully or a faulty unit is found and taken out of service. If all tests pass, the error is recorded and the restart program returns control to the call processing programs.

Let us review briefly what is happening within the two central controls during the fault-recognition program. First the two central controls are brought into step and the directed match mode established. With this accomplished, the test exercises are executed. During these exercises each central control is testing itself via the mechanism of

conditional transfers. If the active central control fails a test, it will transfer to a routine for switching central controls (i.e., interchanging the active and standby central controls) and presumably will switch them by changing the activity flip-flop via a central pulse distributor operation. (The possibility of the active central control being incapable of performing the switch will be discussed below.) Switching of central controls generates a B-level interrupt which will lead back to the central control fault-recognition program, which should find the standby central control faulty (as described below) and take it out of service. Thus the active central control will have been switched out of service.

If the standby central control is faulty and fails a conditional transfer test, it will also transfer to the routine for switching central controls. However, the standby unit is incapable of switching central controls because of its restricted access to the peripheral system. After attempting to perform a switch it will get stuck in a program loop or attempt to follow the actions of the active central control. If the active unit passes its tests (and we assume a fault in only one central control), then when the active unit later examines the match circuits it will see the mismatch generated by the fact that the standby unit transferred to the switch routine. Upon detecting this condition, it will take the standby central control out of service, request a central control diagnosis and return to call processing. The standby central control is taken out of service by modifying the bus control flip-flops so that the standby unit transmits to no external equipment, disjoining the central controls, and setting the trouble flip-flop to inform the emergency-action circuit of the faulty status of the standby.

If a trouble is found in the standby central control, it can be readily taken out of service by the active central control. If the active unit detects trouble within itself it may switch itself out of service. For some very basic troubles the active unit will be incapable of performing this switch operation. For these troubles the emergency-action circuit is relied upon. When the level-C interrupt is generated, a flip-flop is set by the interrupt sequencer, which will activate an emergency-action timer. This circuit will time out in 40 milliseconds if the fault-recognition program does not return to call processing within the 40-millisecond interval. If the central control fault-recognition program is capable of locating the faulty unit and removing it from service, the 40-millisecond timer will be stopped. If, however, the program gets "lost" because of a very basic trouble in the active central control, the emergency action circuit is activated after 40 milliseconds. This circuit will switch the central controls. In addition, if the fault-recognition program recognizes

that it is incapable of performing the switch it will attempt to induce an emergency-action cycle before the 40-millisecond timeout.

To exercise the central controls completely would require at least 25 milliseconds. To avoid taking this much time for each interrupt, the central control fault-recognition program is divided into two parts, a first-look program and a complete check program. Fig. 14 shows how these two programs are used to perform the over-all fault recognition function. As shown, a mismatch causes an interrupt to the first-look program. This program exercises only those portions of the central control which are directly associated with this mismatch point. For example if the mismatch was detected at the index adder output register, the first-look program will exercise the index adder and its input regis-

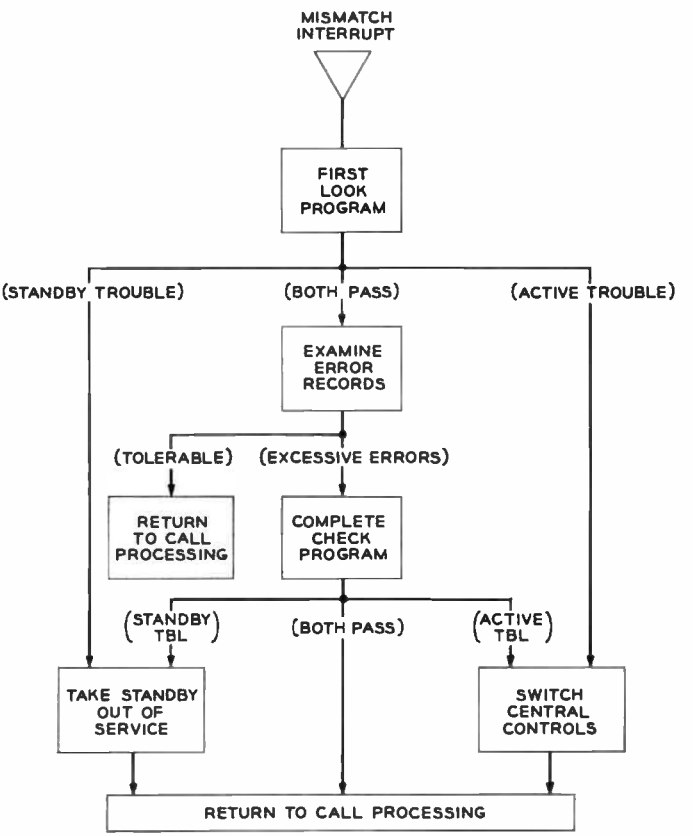


Fig. 14 — Central control fault-recognition program.

ters, and the index registers. If no fault is found in this limited area, it is assumed that the mismatch was caused by an error; the error is recorded and past error records are examined. As long as the error rate is low, the first-look program will return to call processing.

If the error rate is found to be excessive, the complete check program is entered. This program is a fairly complete test of the central controls. The complete check program is also designed to find and classify faults in the circuits interconnecting central control and other units. When necessary, this program will link to the call store, program store and peripheral unit fault recognition routines to recover a complete system. If the complete check program finds a faulty unit, that unit is taken out of service and a diagnostic program requested. If the complete check program is performed a number of times to no avail, additional maintenance actions, such as a diagnosis of the standby central control, will be requested.

5.5.2 *Program Store Fault Recognition*

The program store fault-recognition program is entered by a level-E interrupt following a reread failure of a program store word. A reread is performed when the program store fails to return an all-seems-well signal or when the central control detects a double error or an address error in a program store word.

Since the trouble causing the interrupt may be in the program store containing test programs (the base program store), the initial portion of the level-E interrupt program is located in the call store. Thus a level-E interrupt results in a wired transfer to a call store controlled program.

The principal function of this call store controlled program is to establish a "base" program store (i.e., a program store containing the remaining test programs) with which the active central control can communicate. This program first examines flip-flops within each store which indicate whether that program store had an all-seems-well failure. These flip-flops are examined by scanning. If it is determined that a single program store is faulty, the call store controlled program will take the store out of service, establish connections to a usable base program store, and transfer to a program within this store to complete the fault recognition function.

Not all program store troubles are detected as all-seems-well failures. Troubles in readout channels of the program store or buses will be detected by the central control as double errors, or address errors (or as

an excessive number of single errors, which will be discussed later) and will not be registered in the all-seems-well flip-flops within the program stores. In this case, the faulty store is not immediately identifiable.

To handle troubles of this type a more exhaustive back-up program is required. If the call store controlled program does not find a store with its all-seems-well flip-flop set, it uses an "establish base" routine to connect the active central control to a base program store. If this is successful, control is transferred to a "bootstrap" program located in the base store. This program will test the remaining program stores until it has found a sufficient number of working stores to be able to connect a full copy of memory to the active program store bus. A deferred fault-recognition program is then requested, and control is returned to call processing.

If the establish base and bootstrap programs are unable to establish an active system of program stores, they will induce the emergency-action circuit to take over and alter central control-program store interconnections until a workable configuration is obtained. (See Section 5.6.)

The deferred fault-recognition program which may be requested by the above programs is the highest priority deferrable maintenance task. The function of the deferred fault-recognition program for the program stores is to test all standby program stores and access to all stores from the standby bus, the standby program store bus, and the standby central control. Whenever a trouble occurs which is not a clear-cut single fault in one program store, the interrupt program described above does sufficient testing and switching to recover a complete copy of memory for the active system. However, the status of the standby stores, bus and central controls is left in doubt. The deferred fault-recognition program tests the standby stores, buses and central control. If any of these units is found faulty, it is removed from service and the appropriate diagnosis requested.

The program store fault-recognition program is entered whenever a program store reread failure occurs as a result of a repeated double error, address error, or an all-seems-well failure. Single errors in program store words can be detected and corrected, but will not result in a level-E interrupt. Thus a permanent fault in a readout channel would not initiate the fault recognition program. Routine maintenance programs are provided for troubles of this type.

Whenever a single error is detected, a hardware counter in the central control is incremented. A routine exercise program examines this error counter periodically. As long as the error count remains low, the counter

is reset and call programming is reentered in the normal manner. If the error count is found to be excessive, special error analysis programs are initiated. These programs attempt to locate the unit producing the high error rate by modifying the interconnection configuration of central control, program stores and buses, and again monitoring the error rate. With the error information of the initial configuration plus that obtained by changing the configuration twice, it is possible to locate the error source, provided it is consistently producing errors. When the suspected unit is located, it is taken out of service and a diagnosis for that unit is requested.

5.5.3 *Call Store Fault Recognition*

The duplication and switching plan for the call stores is nearly identical to that for the program stores. Consequently, many of the problems encountered are similar. The program is complicated by having to deal with a larger number of units and by having to cope with the fact that for some troubles it may not have any temporary memory to depend upon, with the exception of the internal central control registers. In addition, all testing of call stores must be performed while protecting the information stored in the temporary memories.

The call store fault recognition is called in as a level-D interrupt program when a reread failure or rewrite failure of a call store word is encountered. When this failure is detected and the interrupt request is made, the address at which the failure occurred is saved in the central control match registers. This program is again divided into a first-look program, which handles most simple troubles, and a bootstrap program to handle the more difficult troubles in the call store community.

The first-look program uses the failing address to determine which of two (recalling duplication) call stores failed. It then examines the failure indications retained in the central controls to determine which central control detected the trouble. Knowing this and the call store interconnection configuration, it determines which call store responded improperly and removes it from active use if a duplicate is available. It then performs an access test to ensure that the new call store configuration is set up properly. If this is performed successfully, a deferred fault-recognition program is requested and call processing is reentered. The deferred fault-recognition program will, at a later time, check to see that the call store removed from service is truly faulty and, if so, mark it in trouble and request a diagnosis. If it determines that the call store is fault-free, it will update its memory and return it to normal operation.

If any of the conditions assumed above are not met, the call store fault-recognition employs a bootstrap program to restore a complete copy of temporary memory to the active system. This program assumes all call stores are faulty until proven otherwise. It tests call stores and buses until it has a complete copy which the active central control can use. It then requests a deferred fault-recognition program to check out the remaining call store units to determine their operational status. Any faulty units are removed from service.

There are also error programs associated with the call stores. These programs use an error counter in the central control plus selected configuration changes to identify the unit generating the errors.

5.6 *Emergency-Action Functions*

System troubles are normally detected by trouble-detection circuits, and the call processing ability of the system is recovered by means of fault-recognition programs as described above. This approach requires a reasonably good central processor. Even though each of the central processor subsystems is duplicated and only one of the duplicated subsystems may be faulty, the fault-recognition programs just described depend upon the active central processor to recover a working system.

To recover from situations where a "sane" central processor is not available, a combined circuit-program facility, labeled "emergency action," has been designed. When trouble is detected within the central control, program store or call store, a 40-millisecond emergency action timer is started. The fault-recognition programs should be able to recover the call processing ability of the system within this interval. However, if the fault-recognition program is not successful, the 40-millisecond emergency-action timer will time out and activate the emergency-action circuit.

The emergency-action circuit establishes various combinations of data processor subsystems without reliance on program instructions. Program instructions are used to determine whether or not the assembled central processor is sane. This program performs a series of tests on the central processor subsystems involved in the new configuration. The program is designed as a maze. To qualify, the central processor must proceed through the maze via one and only one correct path. The rearrangement of subsystems is accomplished by a logic circuit which selects, one at a time, all combinations of central control, program store and program store bus systems. For each selected configuration, the maze program is started and a sanity timer is activated. As long as the maze

program proceeds through the predetermined course, the sanity timer is program reset. When the maze program is completed, the selected configuration is considered sane. On the other hand, if the program strays off course, the sanity timer is not reset and will time out after 0.704 millisecond (128 cycles). The timeout produces an input signal to the emergency action circuit which selects a new program store-bus-central control configuration. This procedure is repeated until the maze program qualifies a configuration as sane.

Since the emergency-action facility is provided as a back-up to the fault-recognition programs, this facility must be made as independent of normal central processor operation as possible. For example, the emergency-action facility is not dependent on the system clock. The emergency-action hardware generates its own pulses that sequence the emergency-action circuit through its actions.

The initial activation of the emergency-action circuit begins a cycle. Subsequent input signals produce state changes advancing a four-stage state counter. This counter records the successive enable signals within a cycle and directs the specific actions which are to be carried out for each timeout.

As the emergency-action sequencer advances through its various states it switches through all possible combinations of central controls, base program stores, and buses. The configurations established by the emergency-action sequencer during each state are summarized in Table III. Note that during the initial state 0000 and during states 0111, 1000, and 1111 no switching is performed and only an interrupt is generated.

TABLE III — CONFIGURATIONS ESTABLISHED BY EMERGENCY-ACTION SWITCHING

Status of Units after a Switch Performed by the Indicated State

EA State	CC0	CC1	PS0	PS1	Bus0	Bus1	Other Stores
X000	U	U	U	U	U	U	U
X001	C	C	U	U	U	U	U
X010	U	U	U	U	C	C	U
X011	U	U	A	S	A	S	T
X100	U	U	A	S	S	A	T
X101	U	U	S	A	S	A	T
X110	U	U	S	A	A	S	T
X111	U	U	S	A	A	S	T

X: don't care
A: this unit is switched active
S: this unit is switched standby
U: the status of this unit is unchanged
C: the status of this unit is complemented
T: this unit is marked in trouble.

With each activation of the emergency-action sequencer, a B-level interrupt is initiated. The configuration changes and the actions listed above are completed before the interrupt program is started. The maze program, if successful, is followed by an emergency-action recovery program which is designed to recover the call processing ability of the system.

The emergency-action interrupt program is divided into five phases. These are as follows:

(1) basic sanity maze program,
(2) operational checks of the central pulse distributor,
(3) operational checks of the call stores,
(4) bootstrap recovery of program stores, and
(5) emergency-action evaluation.

The basic sanity testing is not designed to isolate trouble, but instead to test the ability of the active central processor to process program instructions properly. The program with the aid of a sanity timer determines if the active central processor is operable (sane).

The check of the central pulse distributor is made to ascertain that the central pulse distributor can be used successfully in the program store recovery program. The operational check of the call stores is made for similar reasons. A bootstrap recovery of the program stores is carried out only after the emergency-action sequencer has advanced to or beyond the state where the program store complex was forced into a special configuration.

The emergency-action evaluation program determines the rate at which emergency actions are occurring and attempts to determine the cause of any recurring cycles. It also requests subsequent maintenance program actions to isolate faulty units and to return the fault-free units to service.

The emergency-action circuit is activated initially by one of the following conditions (see Fig. 15):

(a) a clock check circuit detects trouble in the microsecond clock,
(b) a sequencer check circuit finds a locked up central control sequencing circuit, or
(c) a "real-time" check determines that the program is out of step with a time reference provided by the emergency action.

The real-time check is a test of the normalcy of call processing by both circuits and programs. This check assures that call processing has not been limited by the exclusion of some program functions. In addition, this check compares the passage of "real time" as counted by the program to that counted by the hardware. A record of time is kept by the program on the basis of the 5-millisecond interrupts generated by a

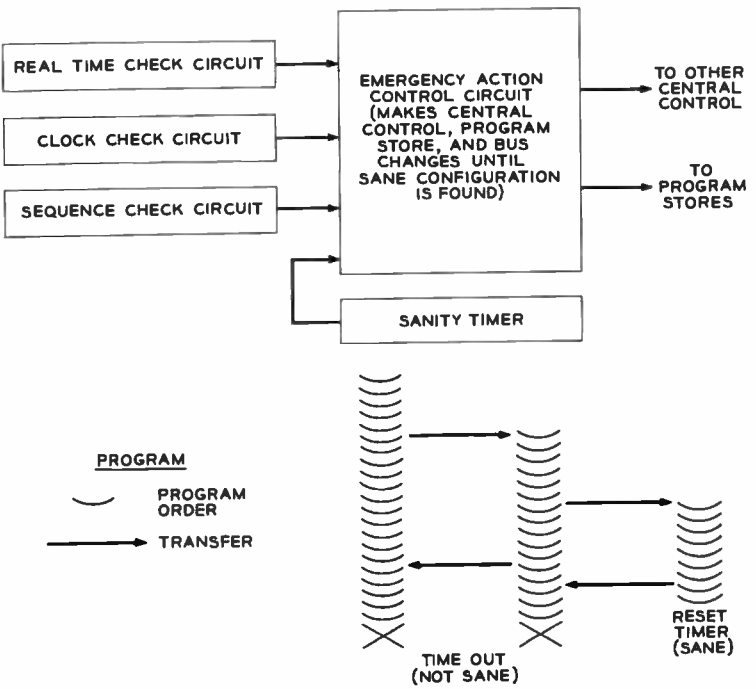


Fig. 15 — Emergency action.

5-millisecond clock. The same clock circuit supplies a signal once every 10 milliseconds to the emergency action counter. This counter, which can count up to 640 milliseconds (see Fig. 16), has two other inputs (enable and reset) which are program controlled. If the program progresses properly through its various tasks and if it stays in step with the emergency action counter, it will first reach a point (T2) where it must generate an enable signal and later a point (T3) where it must generate a reset signal. If the program goes astray and fails to generate either or both signals, the emergency-action counter will time out and activate the emergency-action sequencer within 640 msec.

During the 640-millisecond period, call programs operate on base level and on interrupt levels H and J. The proper recurrence of the interrupt level is reflected in the program count of time and in proper operation of enable and reset inputs to the emergency-action real-time counter.

Associated with each priority level (A through E) of the base level is a reserved location within the call store which is set to the "1" state after the priority level has been visited and each of the jobs at that

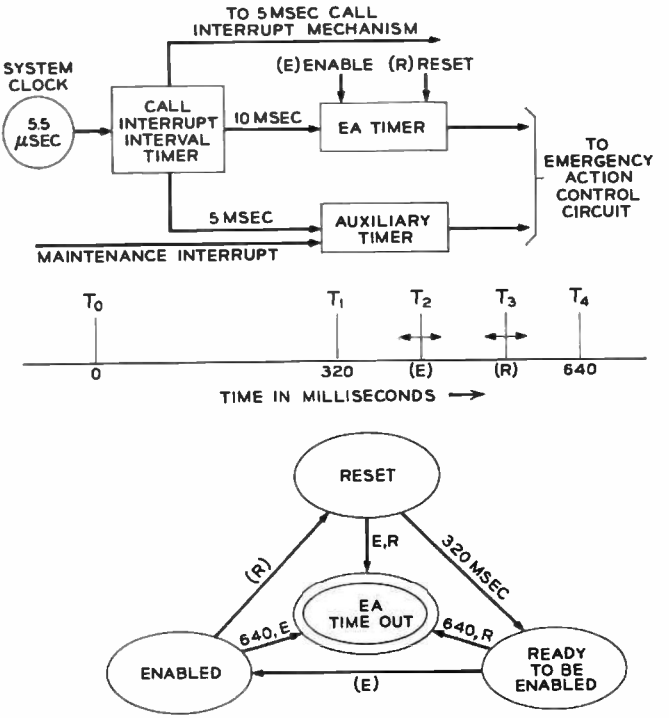


Fig. 16 — Emergency-action timers.

level has been performed. The real-time check program examines these call store locations to determine whether the base-level work is proceeding normally. If these locations indicate the failure to complete the normal amount of work, the real-time control program initiates an overload control. The over-load control slows down the acceptance of new work to allow the assumed backlog of work to be completed. If the overload control fails to recover the normal visits to the base-level jobs, the real-time check assumes that the failure to cycle through base level is due to mutilated data in call store. A reinitialization of vital call store constants is made. If the trouble condition continues, further reinitialization of data is carried out.

5.7 *Implementation of Diagnostic Programs*

5.7.1 *Central Control Diagnosis*

The object of the central control diagnosis is to isolate faults in the standby central control to a small number of replaceable circuit packs.

This is accomplished by performing a series of tests on the standby central control, recording the results of these tests and printing them out on the teletypewriter. The faulty packages are identified by locating the printout in the maintenance dictionary.

In the fault-recognition program, each central control in effect tests itself by performing logical operations and conditional transfers. This technique was employed since it was not known which central control, if either, would be faulty. However, in the diagnosis it is known that the standby central control is faulty and that the active central control is fault-free (assuming only one central control is faulty). Knowing this, a different (more reliable) testing technique can be employed. In this environment the active system can be used to test the standby central control.

To perform a test, one wishes to apply certain inputs to the circuit under test, observe the outputs, compare them with expected outputs, and record results indicating which tests passed and which failed. The facts that the central controls are complete duplicates, are capable of running in synchronism, and that we have a way of comparing their operations (the match circuits) can be used to advantage in testing the standby central control.

The principal testing technique employed in the central control diagnosis is to force the two central controls to execute the same test program and to compare certain critical points using the match circuits. A test result or results are then recorded for each match operation. A mismatch indicates a failure for the operation being tested, and a match indicates that the test passed. In this testing mode, the mismatch sampling order is used to sample the desired information at the desired time. Thus the most frequent type of test in the central control diagnosis is of the following type:

(1) the central controls are forced into step;

(2) a test program is executed which completely exercises a specific hardware function of the central control;

(3) at a number of times during this program, the output (or the nearest available output) of the circuit being tested is sampled using the match circuits and the mismatch sampling order;*

(4) as the sampled matches are executed, the results of these matches are recorded; and

(5) at various points, the central controls are forced back into step to ensure meaningful match results.

* This can be done without depending upon the standby central control to execute the mismatch sampling order properly by operating the standby central control in the directed match mode and the active central control in the mismatch sampling mode.

The programs which actually test the standby central control are divided into subroutines labeled "test phases." A test phase exercises a specific hardware area of central control. For the central control there are 20 test phases. The test phases, defined by the circuits which they test, are:

(1) power and clock circuits,
(2) start, stop and control word reception,
(3) alternate routes to hardware in phase 2,
(4) buffer order word register — error detection and correction,
(5) index adder and index adder registers,
(6) index registers and bus circuits,
(7) decoders,
(8) homogeneity and transfer logic,
(9) program address incrementing,
(10) arithmetic and logic circuitry of the accumulator register,
(11) memory operations,
(12) sequencing circuits,
(13) buffer bus registers,
(14) parity generators and checkers,
(15) maintenance circuits (matchers, etc.),
(16) call store address circuits,
(17) program store address circuits,
(18) scanner answer circuits,
(19) enable control, and
(20) peripheral address circuits.

The phases are performed in the order indicated, except that at certain points the diagnosis is terminated if failures have been detected. The first phase consists of scanner operations which check the power state of the standby central control to determine whether all voltage regulators are functioning normally and whether the microsecond clock appears normal. The above circuits have dc trouble detectors associated with them which are in turn connected to scan points. If a failure is detected in phase 1 no further tests are performed. Phase 2 tests the ability of the standby central control to receive control words from the active central control. The control write facility is used throughout the diagnosis to set up the match control circuits of the standby central control and to get the two central controls in step by control writing into the program address register (PAR). This phase is performed using both call store buses if they are available. If a failure is detected in phase 2, phase 3 is performed and the diagnosis is terminated, with the remaining test results being recorded as all-tests-pass.

Phase 4 tests the buffer order word register (BOWR) and the error

detection circuits. This phase is performed from both program store answer buses if they are available. If a failure is detected, the diagnosis is terminated.

Phase 5 tests the index adder circuits, and phase 6 tests the index registers and bus logic circuitry. If a failure is detected in either of these phases, phase 7 (which tests the decoders) is performed, and the diagnosis is terminated. These phases test registers which must be used in testing all circuits which follow; the diagnosis is terminated to avoid inconsistencies created by initial conditions described earlier.

If the above phases find no trouble, phases 8 through 15 are performed. These phases test all internal circuits which are not associated with external buses. If any failures are detected in these phases, the diagnosis is terminated after phase 15.

If the fault has not been located by any of the preceding phases, phases 16 through 20 are executed. These phases are again concerned with circuits associated with external buses. Each of these phases is performed from both of the duplicate buses associated with the circuits being tested, unless one is not available. If a failure is detected in a phase, the diagnosis is terminated at the completion of that phase.

It is estimated that the central control diagnostic program requires approximately 6,000 program words, and generates approximately 2,000 bits of test results. The 2,000 bits of information will normally not be printed out on the teletypewriter. Instead, a number generation program (to be described later) will operate upon this data and print out a much smaller, easier to handle, number.

5.7.2 *Program Store and Call Store Diagnostic Programs*

The program store and call store diagnostic programs are similar in function and in design. Both of these programs attempt to locate a fault within a store which has previously been found faulty. This objective is accomplished by performing a series of exercises on the faulty store and recording the results of these exercises.

Since it is necessary to be able to diagnose a store with one bus out of service and with one central control out of service, the exercises are performed using the active central control and active bus. Most of the testing is performed using special maintenance orders provided for this purpose. Using the maintenance orders, only the store specified by the order responds, and it sends its answers back on the bus from which it receives, regardless of the state of the answer routing flip-flops. When the faulty store is being tested it is normally connected to receive from the active bus but to send on neither bus for normal orders. Thus, for the

case of program stores, the active central control receives its instructions from a fault-free store connected to the active bus; yet it can also interrogate the faulty store by reading data words using the maintenance orders. Similarly, for the call stores the active central control uses a good set of call stores connected to the active bus for storing and reading temporary information, and it gains access to the faulty store for test purposes by using the maintenance orders.

As in the central control tests, the store tests are divided into meaningful blocks called "phases" which test various blocks of the store circuits. If both buses are available, all of these phases are performed from both buses.

5.8 *Routine Exercise Programs*

The routine exercise programs are basically of two types: those which check to see that various memory items (both call store and flip-flop) are updated, and those which exercise hardware which is not used in normal system operation.

The automatic programs have periodic schedules. There are three classes of automatic programs, where the class is determined by the scheduling technique. Class I programs are rigorously scheduled at a relatively high frequency. These programs are entered from the high-priority main program regardless of the office traffic. They must of necessity be fairly short, since they are performed religiously, even during the busy hour. An example of a program assigned to this class is that program which interrogates and resets the store error counters. This must be done on a strict schedule to ensure meaningful interpretation of the contents of the counters.

Class II programs are also rigorously scheduled, but at a much lower frequency. Programs which must be performed every hour, or at some specific time during the day, are assigned to this class. An example of a program in this class is a program for testing the emergency action circuit. This test should be performed only when traffic is low and consequently would be scheduled daily at 2 a.m. or some other nonbusy hour.

Class III routine exercise programs are the lowest-priority programs in the system. These exercises are performed in system spare time when no other jobs are waiting. These exercises are ordered in a circular list, so that when one is completed, the next one in the list is initiated. Most of the routine exercises are assigned to this class. Some examples of routine exercises in this class are:

(1) a program to exercise the match circuits of the central control to

insure that they are capable of detecting a mismatch and that they will operate correctly in all available modes

(2) a program to exercise the error detection and correction circuits of the central control

(3) programs to check that the stores will respond properly to all maintenance orders

(4) programs to change the interconnection configurations of duplicate units

(5) programs to verify and update status words in temporary memory to insure that they agree with the actual system status.

The demand exercise programs are initiated upon request. The request can be initiated by the teletypewriter or by some other program. All automatic programs can be requested as demand programs. Some examples of demand programs are:

(1) a program to remove a unit from service

(2) a program to restore a unit to service

(3) a program to print out the status of a particular unit

(4) a program to print out the contents of a specified call store location, etc., and

(5) programs that audit the network memory.

All of the maintenance programs are normally executed by the active data processor or by both the active and standby systems. Tools in the form of demand exercise programs will also be provided to allow the execution of almost all of these programs on a repeated basis by the standby system. The need for this ability may arise if a marginal trouble develops which is not detected by the diagnostic programs. For troubles of this type it may be desirable to execute some maintenance programs continuously in the standby system while maintenance personnel make observations with an oscilloscope or some other manual tool.

Circuit tools are available, and program tools will be provided, for this purpose. To run the standby central processor independently of the active central processor (i.e., off-line), all that is required is to interconnect the central controls, call stores, and program stores so they operate as two independent systems. As described in Section III, with complete duplication this ability exists. The start-stop control, control write facility, and the breakpoint match mode also provide circuit tools which allow the active system to start the standby system in any program and to stop it at any desired address.

For example, if it is desired to execute some program continuously, starting at address A and ending at address B, this could be accomplished in the following manner. An input message would request that the

standby system execute the program from address A to address B. This demand exercise would modify call store and program store configurations to set up two independent systems. The active central control would stop the standby central control and control write the start address A into the standby program address register. It would then set up the breakpoint match mode to monitor address B with the interrupt and stop standby options specified. Next, it would start the standby central control and return to call processing.

The standby system would then begin executing program at address A while the active system ran its normal call programs. When the standby system reached address B, the active matchers would be alerted, stop the standby system and interrupt the active system with a G-level interrupt. This interrupt program could restart the standby system at A and return to call processing, etc.

VI. MAN-MACHINE RELATIONSHIP

6.1 *Reaction to Trouble*

Let us now review briefly the facilities through which the maintenance man normally communicates with the system. Most of these facilities are included in the four frames which are called the "master control center." When a failure occurs in the system it is most frequently detected by circuits. Programs are immediately brought in to remove the faulty unit from the system and to establish a working system configuration. At a later time, diagnostic programs are run on the faulty unit. For example, assume that a scanner controller has failed, has been switched out of service, and has been diagnosed. At this point, the office alarm system will sound an audible minor alarm and light lamps that direct the maintenance man to the master control center. At the master control center a maintenance teletypewriter prints out the identity of the faulty controller and also a code which the maintenance man, with the aid of a dictionary, translates to the location and identity of the faulty package. The man, using this information, will locate the frame containing the faulty controller. At the frame a red trouble lamp will indicate which controller within the frame is out of service. By pushbutton he will remove the power from the controller and replace the faulty package. With another pushbutton he reapplies power. This action signals the system to start a diagnosis on the controller. If the diagnosis passes, the system will extinguish the trouble light at the controller as a signal that the controller is fault-free. Fig. 17 illustrates this flow of actions.

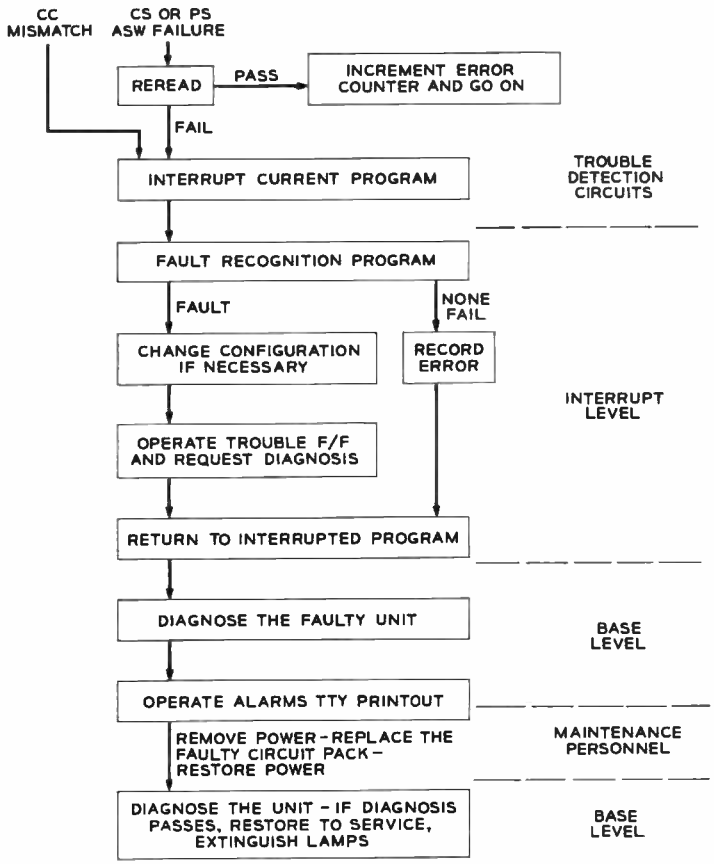


Fig. 17 — Maintenance reaction to trouble.

6.2 *Maintenance Dictionary Production*

The maintenance dictionaries are used for translating diagnostic program output, as printed out on maintenance teletypewriters, to specific package locations. Because of the complexity of No. 1 ESS and because of the number of diagnostic tests, it is considered infeasible to produce complete and accurate dictionaries by logical reasoning alone, i.e., to predict the reaction of each test to every possible fault. The method that will be used to produce dictionaries is: each plug-in circuit pack will be replaced by a fault simulator which will introduce every possible type of single fault on the replaced package one at a time and

then record the system reaction on a high-speed output tape.* From this record, the dictionaries can be produced by sorting and ordering the test results using auxiliary data processing equipment. In addition to the relatively complete and accurate dictionaries which will result, this method also provides early feedback for evaluation of the maintenance plan — feedback which otherwise would take years to collect from operational experience. At the time diagnostic data for dictionaries are collected, additional data will be collected on key program reactions and decisions. The kinds of data which can be collected are: the system configuration changes, the length of time required by the various phases of the program, program interactions, and interrupts generated. These data can be evaluated to find any weaknesses of the maintenance programs and also to find redundant or troublesome programs.

The data generated by diagnostic programs consist of a binary word of n bits. Each bit may represent the result of a test (pass or fail), or it may take several bits of information to represent the result of a test. A simple way of organizing the dictionaries is to convert these binary words to decimal numbers, place them in numerical order, and list next to each number the faulty packages that generated that number. This is essentially what was done for the Morris ESS central control dictionary.[16] This dictionary, although somewhat bulky (it consisted of some 1200 double-thickness pages), was quite satisfactory when the printout in the field exactly matched a dictionary entry. When faults were introduced into the Morris central control, it was found that approximately 60 per cent produced test results that exactly matched the dictionary entries. Another 16 per cent of the faults produced entries that could be located by using some relatively awkward interpretative rules. There are many factors that can cause the diagnostic results for a given fault to differ from one machine to another or from one occurrence to another. Test results may depend on the memory state of the machine at the time the trouble occurs,† and variations in component and voltage values in two systems may be sufficient to produce different results. It is expected that a similar situation may exist in No. 1 ESS, although to a lesser degree. Consequently, it is desirable to present diagnostic data in the dictionaries in a form such that if an exact match cannot be found, the dictionary entry (or entries) most nearly resembling the diagnostic printout can be easily located.

* This process has to be carried out only once unless changes are made in the maintenance program or in the circuits.

† As stated previously, in designing diagnostic tests extensive efforts are being made to avoid this problem.

The diagnostic data can be given a geometric interpretation by considering the given test result as a point in a multidimensional space where the binary word that represents the test results gives the coordinates of the point. The method that will be used in No. 1 ESS to generate diagnostic dictionaries takes an advantage of the fact that (1) of the many possible test patterns relatively few will occur* (the space is sparsely populated) and (2) that the significance of individual tests varies. The dictionary entries will express the weighted (according to the significance of the tests) distance of each point from preselected reference points, rather than specifying the exact coordinate of the test result. This method will yield (1) a more compact dictionary, since the distances can be expressed more concisely than the actual coordinates, and (2) a dictionary in which similar patterns can be easily located when an exact match cannot be found. The methods of diagnostic data collection and dictionary generation will be subjects of a future article.

VII. CONCLUSIONS

The design of maintenance programs has been based on the logical analysis of circuit diagrams rather than on actual experience with the system. Because of this and the complexity of the system, it is expected that the first design will not completely meet all of the objectives. The data produced for the dictionaries will provide early feedback on the strengths and weaknesses of the maintenance plan. This feedback should enable us to evolve a design which will meet all of the design objectives. It is also hoped that with adequate feedback the 47,000-word maintenance program can be substantially reduced.

VIII. ACKNOWLEDGMENTS

To design a central office which is both reliable and maintainable has required the complete cooperation and awareness of everyone connected with the design of No. 1 ESS. The authors would like to pay special tribute to their coworkers in the No. 1 ESS maintenance planning department whose work is summarized herein.

REFERENCES

1. Keister, W., Ketchledge, R. W., and Vaughan, H. E., No. 1 ESS System Organization and Objectives, B.S.T.J., this issue, p. 1831.

* For example, in the No. 1 ESS central control approximately 2×10^5 faults will be introduced; the diagnostic program consists of approximately 2000 tests. Thus, assuming all tests are independent, there are 2^{2000} possible test patterns, of which only 2×10^5 will occur.

2. Harr, J. A., Taylor, F. F., and Ulrich, W., Organization of No. 1 ESS Central Processor, B.S.T.J., this issue, p. 1845.
3. Cagle, W. B., Menne, R. S., Skinner, R. S., Staehler, R. E., and Underwood, M. D., No. 1 ESS Logic Circuits and Their Application to the Design of the Central Control, B.S.T.J., this issue, p. 2055.
4. Ferguson, J. G., Grutzner, W. E., Koehler, D. C., Skinner, R. S., Skubiak, M. T., and Wetherell, D. H., No. 1 ESS Apparatus and Equipment, B.S.T.J., this issue, Part 2.
5. Chevalier, J. G., and Eisenhart, R. K., No. 1 ESS Circuit Packs and Connectors, B.S.T.J., this issue, Part 2.
6. Danielson, D., Dunlap, K. S., and Hofmann, H. R., No. 1 ESS Switching Network Frames and Circuits, B.S.T.J., this issue, Part 2.
7. Freimanis, L., Guercio, A. M., and May, H. F., No. 1 ESS Scanner, Signal Distributor, and Central Pulse Distributor, B.S.T.J., this issue, Part 2.
8. Dougherty, H. J., Raag, H., Ridinger, P. G., and Stockert, A. A., No. 1 ESS Master Control Center, B.S.T.J., this issue, Part 2.
9. Connell, J. B., Hussey, L. W., and Ketchledge, R. W., No. 1 ESS Bus System, B.S.T.J., this issue, p. 2021.
10. Ault, C. F., Gallaher, L. E., Greenwood, T. S., and Koehler, D. C., No. 1 ESS Program Store, B.S.T.J., this issue, p. 2097.
11. Feiner, A., and Hayward, W. S., No. 1 ESS Switching Network Plan, B.S.T.J., this issue, Part 2.
12. Genke, R. M., Harding, P. A., and Staehler, R. E., No. 1 ESS Program Store — A 0.2-Megabit Ferrite Sheet Memory, B.S.T.J., this issue, p. 2147.
13. Biddulph, R., Budlong, A. H., Casterline, R. C., Funk, D. L., and Goeller, L. F., Line, Trunk, Junctor and Service Circuits for No. 1 ESS, B.S.T.J., this issue, Part 2.
14. Harr, J. H., Hoover, Mrs. E. S., and Smith, R. B., Organization of the No. 1 ESS Stored Program, B.S.T.J., this issue, 1923.
15. Carbaugh, D. H., Drew, G. G., Ghiron, H., and Hoover, Mrs. E. S., No. 1 ESS Call Processing, B.S.T.J., this issue, Part 2.
16. Tsiang, S. H., and Ulrich, W., Automatic Trouble Diagnosis of Complex Logic Circuits, B.S.T.J., **41**, July, 1962, p. 1177.

No. 1 ESS Bus System

By J. B. CONNELL, L. W. HUSSEY and R. W. KETCHLEDGE

(Manuscript received January 15, 1964)

Communication between the various units within No. 1 ESS demands a complex network of transmission lines. The requirements are radically different from previous offices because a large amount of information must be transmitted at high speed, in digital form, between a multiplicity of locations with a high degree of reliability.

A description is given of objectives and of organization and operational modes for the network. The problems involved are discussed together with the hardware by which the objectives were successfully realized.

I. INTRODUCTION

The function of the bus system of No. 1 ESS is to provide intramachine data and control communication. The need for large amounts of such communication is a natural outgrowth of the functional unit structure of the over-all system. At the same time the interconnection method must be highly flexible to permit easy growth and to adapt to large variations between different installations.

Existing electromechanical switching systems send most of their intramachine data by dc ground-return signaling. Most relays used in telephony are relatively easy to control at a distance because of their limited frequency response and high signal threshold. Semiconductor logic circuits, on the other hand, have a wide frequency response and can be disturbed by relatively small noise signals.

It became apparent quite early in the development that conventional interconnection techniques would not be satisfactory. The No. 1 ESS bus system described in this paper has proved to be a successful solution of the interconnection problem.

1.1 *Size*

Telephone switching systems tend to be relatively large assemblies of equipment consisting of many separate functional units. Compared to

previous systems, No. 1 ESS is physically smaller and is composed of fewer separate functional units. However, even a small 5000-line No. 1 ESS might typically contain 3 temporary memories, 2 permanent memories, 2 central controls, 2 master scanners, 2 central pulse distributors and a master control center plus 19 switching network, trunk or junctor units — a total of over 30 separate functional units. A very large office would contain hundreds of such units. Similarly, while distances between units may be only a few tens of feet in a very small office, a very large office will require interconnecting leads hundreds of feet long.

To assemble the functional units into a working system requires interconnections which can transmit data and control information between units with the utmost dependability. In most cases a multiplicity of sources and sinks is involved. For example, either central control must be capable of sending orders to hundreds of network controllers.

1.2 *Duplication*

All system units required to provide continuous service are duplicated. This includes the bus system. When one or more functional units are out of service the effective interconnection pattern must be modified appropriately.

This requires that the buses provide suitable interconnections both normally and under trouble conditions. The bus structure must permit assembly of a fully operational system in the presence of any pattern of faults that does not include simultaneous failures of both units of a duplicate pair.

Through the use of duplicated buses and provision of multiple-source, multiple-sink capability, the buses become a means for achieving the necessary duplicate switching.

The Morris, Illinois, trial[1] demonstrated that, to avoid any disturbance of telephone service when a functional unit fails, the duplicate switching must be performed rapidly. Relays used in the Morris trial equipment to switch service to duplicates were found to be too slow. Thus duplicate switching at electronic speeds appeared necessary. To achieve this goal, the bus system provides all interconnection facilities that the system will require under any conditions. At any instant only a limited set of these possibilities will be in use. Each functional unit can be instructed to receive on a particular bus or its duplicate, to send on a particular bus or its duplicate, or in some cases to send on both. Means are also provided to disable the sending circuits so that false pulses due to trouble conditions cannot destroy the usefulness of the bus to the other units it serves in multiple.

1.3 *Speed*

Cycle times for the control actions of No. 1 ESS are 5.5 microseconds. In a typical cycle, a read operation in the program store can occur simultaneously with a read or write operation in a call store, and both of these may be simultaneous with an instruction to a peripheral unit. To keep the time wasted in propagation to a minimum, the bus lengths are kept as short as possible and the buses are word organized, with parallel transmission of bits. Maximum lengths of 125 feet between any central control and any store keep propagation times to a small but not inconsequential fraction of a cycle. For peripheral units, longer bus lengths of up to 450 feet are required in large offices.

Considerations of speed are also involved in the choice of a 0.5-microsecond pulse as the basic bus signal. Shorter pulses would be more difficult to generate, transmit and detect, while longer pulses would cost time. Also, the 0.5-microsecond pulse is short compared to the repetition time of 5.5 microseconds, and dc restoration is therefore not necessary.

1.4 *Environment*

A number of environmental factors complicate the bus problem. These include the use of common storage battery power and the large physical size of the system, which together cause significant dc potential differences to exist between the grounds of various units. This ground potential problem is avoided by making the bus system ac-coupled. This takes care of dc noise, but other noise sources are also important. Relay circuits in No. 1 ESS are carefully protected to minimize noise from this source. However, a No. 1 ESS may be adjacent to an electromechanical central office whose relays might cause interference. Balanced transmission, shielding where necessary, and separation of bus conductors from other leads minimize such noise pickup. In addition, buses generally include synchronizing or enabling signals that reduce noise effects by time discrimination. An enable pulse is used to activate the other bus sensing circuits. Occasional noise pulses occurring in the absence of a pulse on the enable lead will not normally be sensed.

1.5 *Standardization with Flexibility*

A strong effort towards standardization and code minimization has been made in the No. 1 ESS development. Thus the same techniques, circuits and components have been used for all of the high-speed interconnections of the system. However, flexibility of system arrangements has been retained. This is largely due to the multiple-source, multiple-

sink capability of the bus system and to the use of separate program store, call store and peripheral unit buses. This flexibility is particularly important when additions are made to a working office. When a new equipment unit is added, the appropriate buses are easily extended to the new unit. The bus duplication is used to make the extensions without service interruption.

II. INTERFRAME COMMUNICATION SYSTEM ORGANIZATION AND OPERATION

2.1 *General Organization and Growth*

A simplified diagram of the No. 1 ESS System is shown in Fig. 1. The input-output units represent a wide range of physical units: the space-division network controllers, ferrod scanner units, trunk and junctor signal distributor units, message accounting tape units, and teletype-writers. In its simplest form, the central control complex is capable of only one operation with these equipments during any particular input-output cycle. As a result, all such units are designed to work from one master bus system called the "peripheral bus system." This peripheral bus system is expandable to a very large number of input-output units.

Fig. 1 shows duplicate central control units connected to the above mentioned peripheral bus system. It also shows them connected to a group of program stores via a program store bus system and to a group of call stores via a call store bus system.

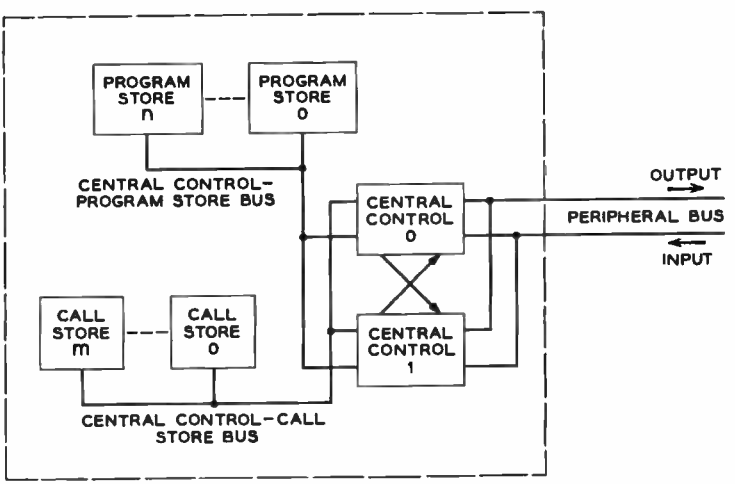


Fig. 1 — Simplified system diagram.

All of the central control operating programs and some rarely changed data (translation data, primarily) are maintained in the program stores. The call store contains data normally modified as the result of processing. By using, with some exceptions, separate memory systems for instructions and for data it is possible to utilize parallel memory operations and instruction overlap operation. This permits more data processing per unit of real time, which in turn increases the call handling capacity of the system.

There are groups of leads interconnecting the central controls shown in Fig. 1. The central controls operate in parallel, and these leads are required for synchronization and for maintenance matching.

2.2 *Bus Control*

Each of the three major bus systems can be thought of as a separate subsystem. The major difference between a bus subsystem and any of the other subsystem units, for example a program store, is that the bus is physically distributed over many equipment frames and its control, although conceptually centralized, is similarly distributed over many units. All bus operations are initiated by the central control. It is the central control that requests information from the program store, reads or writes in the call store, and requests input information or transmits instructions to units on the peripheral bus. Fig. 2 depicts the basic problem for a typical bus system. The central controls are shown on the left of the figure. The buses are shown duplicated, and each consists of two one-way groups of twisted wire pairs (designated "address" and "answer" groups). Each one-way group contains one or more functional groupings, although no such breakdown is shown on this figure. The operation of the bus is such that only one unit (source) can transmit on any bus

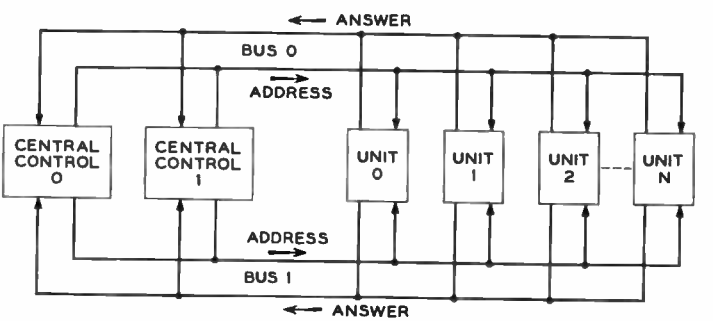


Fig. 2 — Typical bus configuration.

group (e.g., bus 0 address group). However, all the sink units can receive from the same bus. The control problems common to all the No. 1 ESS bus systems are the following:

(1) control of the configuration between central controls and the buses for both transmission and reception,

(2) selection of the unit on the bus for which the transmission is intended, and

(3) control of the configuration between the units on the buses and the buses for both transmission and reception.

The solution to these three problems depends very heavily on the duplication scheme used with the equipment served by the bus system.

2.3 *Program Store Interconnections and Operation*[2]

Due to two major factors, the frequency of use of the memory units and the memory address limitations of the central control, a code select method is used with the memory bus systems. Each transmission from the central control to either the program store or call store bus system is accompanied by a group of code bits (4 bits with the program store bus system and 6 bits with the call store bus system). These code bits are received by all memory units on the bus system. Each store has preset into it the codes for the two information blocks its contains. Only if a store contains the block requested in either its left or right half will it respond to the address bits. The address bits are used to specify the location in the memory block. Normally, two stores will respond to each request.

The addresses for both the program stores and the call stores are generated by the central control. These addresses are formed from a 21-bit memory address field. Table I lists the memory address spectrum assignments. If a program store is being addressed, 20 bits of the memory address spectrum are required. The 21st bit is used to determine which part of the program store word is to be used by the central control. This bit is not sent to the program stores, and it is used only when receiving data from the program store. Of the 20 bits that are transmitted as a program store address, 4 are used as information block code bits and 16 are used to specify an address within the information block.

The program store is a read-only type of memory unit. As a result, there is no data bus for writing into the program store. However, there are other bits of information sent to the program stores. Four bits are used to specify one of five program store operating modes. A synchronizing bit is also sent to the program stores. It is used to control the gating of the information bits into the program store receiving circuits. The

TABLE I — MEMORY ADDRESS SPECTRUM

P.S. Bus Code Bits			C.S. Bus Code Bits							
20	19	18	17	16	15	14	13	12	11–0	
0	0	0	0	0	0	0	0	0	all 0's	buffer reg operations
0	0	0	0	0	0	0	0	0	all 1's	
0	0	0	0	0	0	0	0	1	all 0's	central control call stores
0	0	0	1	0	1	1	1	1	all 1's	
0	0	0	1	1	0	0	0	0	all 0's	reserved for future allocation
0	0	0	1	1	0	1	1	1	all 1's	
0	0	0	1	1	1	0	0	0	0's–1's	
0	0	0	1	1	1	0	0	1	0's–1's	
0	0	0	1	1	1	0	1	0	0's–1's	
0	0	0	1	1	1	0	1	1	0's–1's	
0	0	0	1	1	1	1	d	d	0's–1's	standby central control
0	0	1	0	0	0	0	0	0	all 0's	right half program store words
0	1	1	1	1	1	1	1	1	all 1's	
1	0	0	0	0	0	0	0	0	all 0's	unused
1	0	0	1	1	1	1	1	1	all 1's	
1	0	1	0	0	0	0	0	0	all 0's	left half program store words
1	1	1	1	1	1	1	1	1	all 1's	

use of synchronizing pulses reduces the time during which the program stores are exposed to noise on the buses.

The central control receives 44 information bits and an all-seems-well signal from the program store. The all-seems-well signal is returned by the program store if certain conditions are satisfied during the execution of the central control request. A synchronizing signal is also sent to the central control together with the program store readout and all-seems-well signal. This is used to reduce the time that the central controls are exposed to noise on the bus system. Fig. 3 depicts the bus system and points out its ability to expand. Also shown in Fig. 3 are two separate answer buses, an east and a west. The need for two separate answer buses arose because of timing considerations and because of the directionality of the cable receivers used in the bus system. The address bus does not need to be separate, since the cable drivers transmit along the bus leads in both directions.

Fig. 4 presents a timing diagram for program store bus operations. This timing diagram shows the relative time between transmissions and receptions. A diagram showing the relative timing between program store bus operation, call store bus operation and peripheral bus operations will be presented later in this article.

Route control flip-flops located in the central controls and program

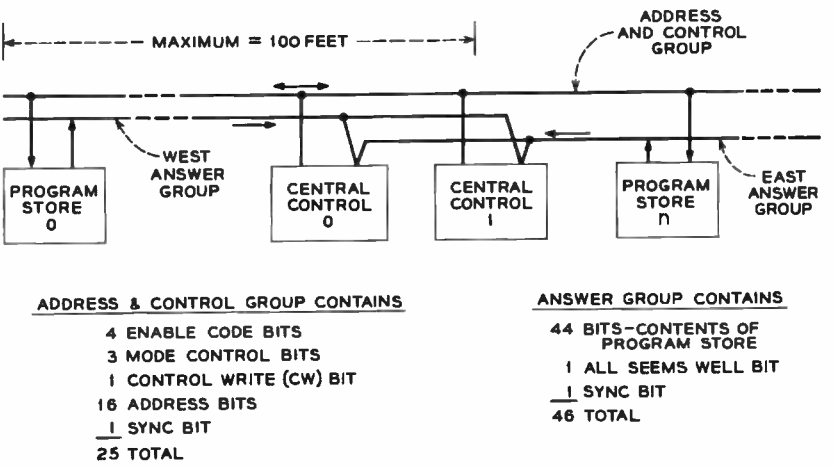


Fig. 3 — Program store bus system.

stores allow a multiplicity of program store bus configurations. Three flip-flops are used in the central controls for normal control. Seven flip-flops within each program store control the configuration between the buses and the store.

2.4 *Call Store Interconnections and Operation*[3]

A central control can address its own call stores, the other central control, or a group of special control flip-flop locations termed "buffer control registers." These are all addressed via the central control index adder output register, receive data from the central control data buffer register, and transmit data to the central control data buffer register. Choice among these actions is dictated by mutually exclusive addresses

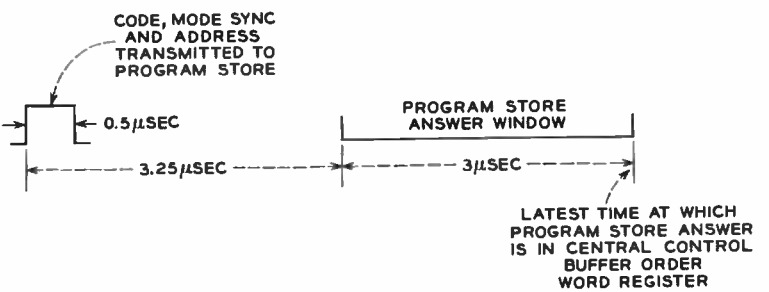
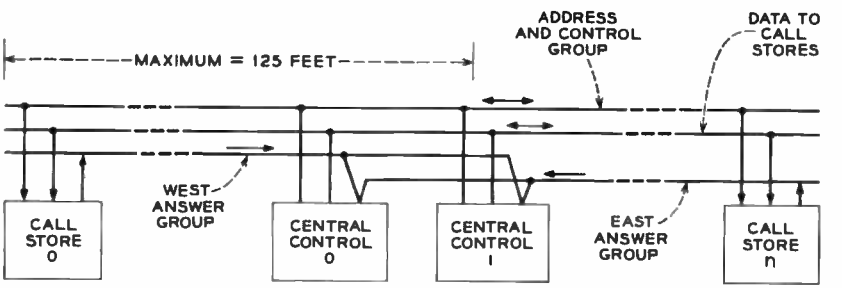


Fig. 4 — Program store system timing.

recognized at the output of the index adder output register in the central control. Table I contains the address assignments for these operations. The buffer control registers are internal to the central control and are not accessed via the call store bus system except when one central control is writing in one of these registers in the other central control. They are mentioned here for completeness.

The call store bus is duplicated. Only one bus is shown in Fig. 5. As shown in this figure, this bus contains an address and control group, an information group for writing in a unit on the call store bus and an answer group for reading from a unit on the call store bus. The address and control group transmits two synchronizing pulses, six enable code bits, three mode-control signals (C, G, H), a 12-bit address, read or write signal and a parity bit which is computed over the address and code. Two synchronizing pulses are required, since the information is sent during two separate time periods. The information group transmits a synchronizing pulse, a 23-bit information word, and a parity signal which is computed over the information, address, and code. This parity bit is stored with the word and is checked later when the word is read out of the call store. Fig. 6 shows a timing diagram for call store write and read operations. The answer bus returns a synchronizing pulse, the stored



ADDRESS & CONTROL GROUP CONTAINS

6	ENABLE CODE BITS
3	MODE CONTROL SIGNALS
12	ADDRESS BITS
1	READ BIT
1	WRITE BIT
1	PARITY BIT
2	SYNC BITS
26	TOTAL

ANSWER GROUP CONTAINS

24	DATA BITS
1	ALL SEEMS WELL BIT
1	SYNC BIT
26	TOTAL

DATA TO CALL STORE GROUP CONTAINS

24	DATA BITS
1	SYNC BIT
25	TOTAL

Fig. 5 — Call store bus system.

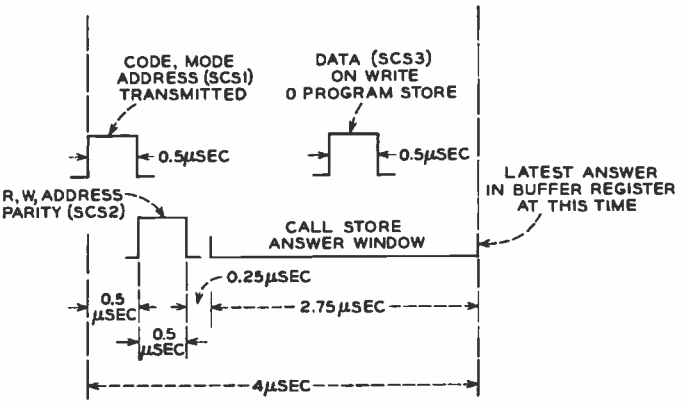


Fig. 6 — Call store system timing.

parity bit, 23 bits of information, and an all-seems-well signal. The all-seems-well signal returns only if certain conditions are met while the call store is executing a central control requested operation. There are east and west answer buses similar to those discussed with the program store answer bus. They are required for the same reasons.

Route control flip-flops located in the central controls and call stores allow a multiplicity of call store bus configurations. Three flip-flops are used in the central controls for normal control. Seven flip-flops within each call store control the configuration between the buses and the store.

2.5 *Peripheral System Interconnections and Operation*

All peripheral units will receive either data or instructions from the duplicated peripheral address bus. Peripheral units such as scanners, signal distributors, and network switch bays have duplicated controllers. Each of the two controllers can be connected to either address bus. The basic bus-to-controller logic for all such units is shown in Fig. 7. The four leads marked E_0, E_1, E_2 and E_3 are the enable leads. They activate the unit and simultaneously select the bus-to-controller path. These four signals are supplied from central pulse distributor units, which are duplicated. Two enable leads are supplied from each of the duplicate central pulse distributors. There are some units on the peripheral address bus which have unduplicated controllers. These controllers will have access to both address buses. For such units, two enable paths, one from each central pulse distributor, will be used to activate the unit and select the bus-to-controller path.

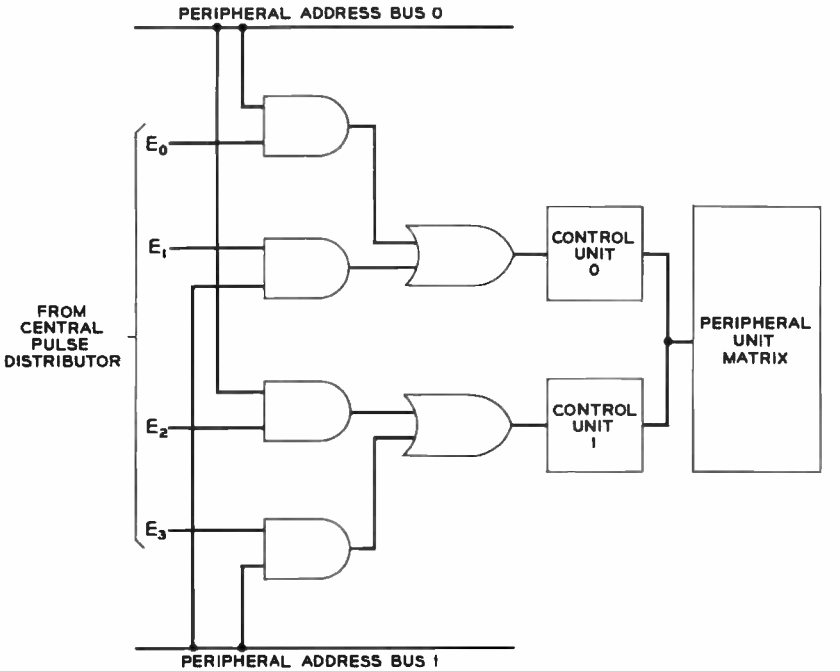


Fig. 7 — Peripheral unit enabling.

In addition to the address bus and the enable facilities for peripheral units, there is also an answer bus for units which transmit information to the central control.

The central pulse distributor outputs are used to perform one of two major functions. The first of these is the peripheral unit enable function and the second is that of providing high-speed discrete control over various flip-flops that may be located throughout the system. The enabling function is performed concurrently with peripheral address operations. As a result, a separate bus system is needed between the central controls and the central pulse distributors. When a non-enable operation is performed using the central pulse distributor, only its bus system is used. Although the central pulse distributors have a separate bus system, it is thought of as being nested into the peripheral bus system and is considered for purposes of operation as well as discussion part of the peripheral bus system.

Fig. 8 is a diagram of the interconnections between the central controls and the peripheral system, including the central pulse distributors. Shown in this figure are:

(1) peripheral unit address bus
(2) peripheral unit answer bus
(3) central pulse distributor address bus
(4) central pulse distributor verify-answer bus
(5) central pulse distributor execute leads
(6) central pulse distributor echo leads
(7) central pulse distributor unipolar outputs (enable leads)
(8) central pulse distributor bipolar outputs.

The central pulse distributor is a matrix of 1024 high-speed, low-level pulsing sources. It can provide two types of outputs; bipolar pulses and unipolar pulses. A bipolar pulse uses both polarities on a single twisted wire pair, whereas a unipolar pulse uses a single polarity on a twisted wire pair. Although the bipolar output is on a single twisted wire pair, it

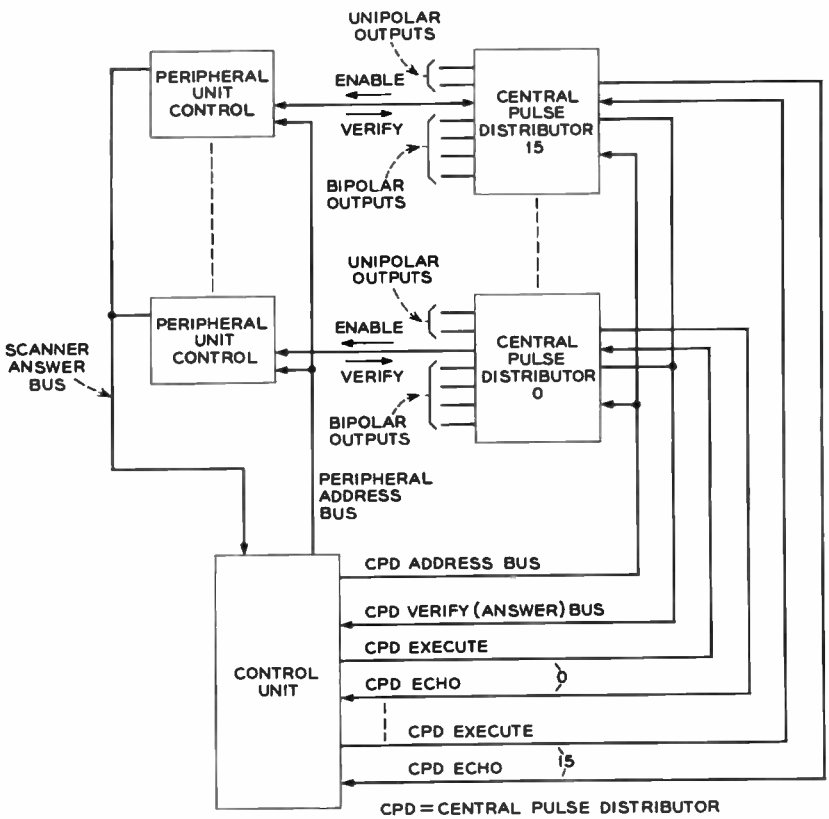


Fig. 8 — Peripheral bus system.

requires two central pulse distributor matrix points to produce the two polarity pulses. One point in the matrix is required for each unipolar output.

The central pulse distributor is effectively a one-out-of-n translator for enabling purposes. As shown in Fig. 8, there can be as many as 16 central pulse distributor units. They are all accessed over a common address bus and they reply over a common answer bus. An execute translator located in the central control is used to select a single central pulse distributor. There are 16 outputs in this translator, one for each central pulse distributor. Upon reception of an execute pulse, the central pulse distributor will transmit an execute-verify signal to the central control over a separate lead. This signal is called the "central pulse distributor echo." Sixteen echo inputs, one from each central pulse distributor, connect to the central control. These signals are received and compared against the output of the execute translator whenever a central pulse distributor is used.

A central pulse distributor has a variable range of combinations of unipolar and bipolar outputs. A maximum of 512 unipolar points can be served by one central pulse distributor. This uses one half of its output matrix. The other half must be used for bipolar pulsing only and can produce 256 such outputs. The unipolar half can be adapted for bipolar use, thus affording up to 512 bipolar outputs per central pulse distributor.

Unipolar pulses are the only type used for peripheral unit enabling. The half of the central pulse distributor matrix that can produce unipolar outputs is equipped with a verification feature. Whenever a peripheral unit is being addressed via the peripheral address bus, an enable address corresponding to that unit must be sent to the central pulse distributor. This address chooses a unipolar output point, and a pulse is transmitted by the central pulse distributor on a single twisted wire pair to a peripheral unit. It is this pulse which enables a peripheral unit to receive information from the peripheral bus. After the information is received from the peripheral bus and checked for proper address, a verify pulse is returned to the central pulse distributor on the same twisted wire pair. The central pulse distributor receives this verify pulse and codes the matrix point on which it was received into the same address code format initially sent to the central pulse distributor from the central control. This is then transmitted to the central control on the central pulse distributor verify answer bus. The central control compares this answer to the address it had sent out. If a unipolar output is used for other than peripheral unit enabling and does not return a verify pulse, it is up to the central control to recognize this as the case and ignore the

central pulse distributor verify answer. The verification feature does not exist for the bipolar half of the matrix nor for those normally unipolar half outputs which are connected to provide bipolar outputs.

Binary information is used to select the proper central pulse distributor, address the central pulse distributor and address peripheral units. However, it is not transmitted to the peripheral units or the central pulse distributors in this form. It is pretranslated in the central control.

There are 14 binary bits used to address central pulse distributors. Four of these bits are used to select which central pulse distributor will be used. The ten binary-coded address bits are pretranslated by the central control into two coded groups of one-out-of-eight and one group of one-out-of-sixteen. These require 32 address bus leads. The one-out-of-sixteen group can be thought of as two groups of one-of-eight where one and only one of these groups is used for an operation. It is the division of this group that determines which half of the output matrix is being used (e.g., verification half or not). Thus only three groups of one-out-of eight are required to specify a point in the verification half of the output matrix. The answer bus then consists of 24 leads treated as three groups of one-out-of-eight. Although more address bus leads are required with this pretranslation technique, the saving of translation equipment at the central pulse distributors is sufficient to warrant this operation.

A multiplicity of units share the peripheral address bus system. The major units — scanners, signal distributors and network switch frames — have their information pretranslated at the central control before being placed on the peripheral bus. The information for these units (addresses and instructions) is maintained in the central control in binary form. One register location is used for generating peripheral addresses in the central control; this is the addend K register. A group of translators connect between this register and the peripheral address bus. Selection of the proper translator must be made with every peripheral operation. Fig. 9 shows the information groupings for the units on the peripheral bus. Fig. 10 shows the binary form as placed in the addend K register. Information can also be placed on the peripheral address bus in untranslated binary form. Choice among translators will be discussed below. The size of the peripheral bus in terms of the number of leads is dictated by the translator coding for the network switch bays. As seen from Fig. 9, 36 leads are needed in the peripheral address bus for these units. There are two additional leads in the peripheral address bus; these are the network reset lead and the false cross and ground test lead.

The peripheral answer bus is made up of 17 leads; 16 of these carry answer information, while the 17th is an all-seems-well scanner lead.

The control of the peripheral bus system must be flexible, because of

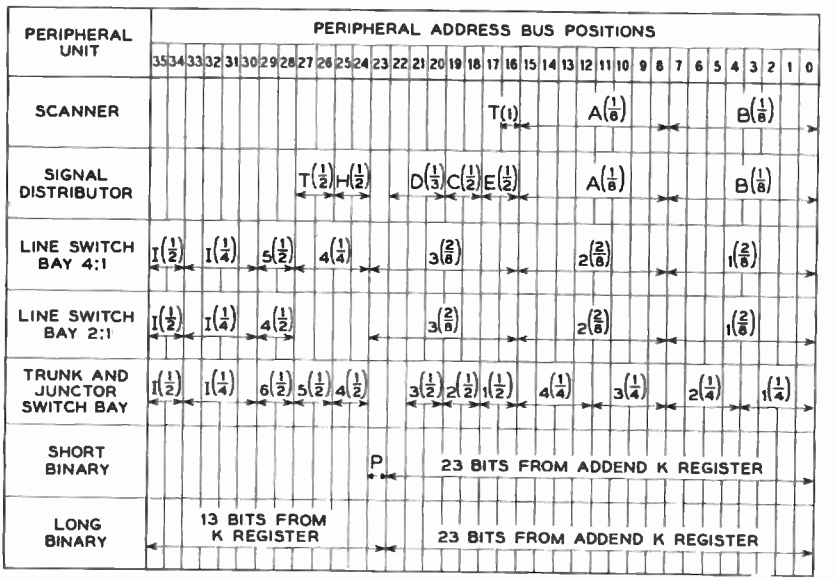
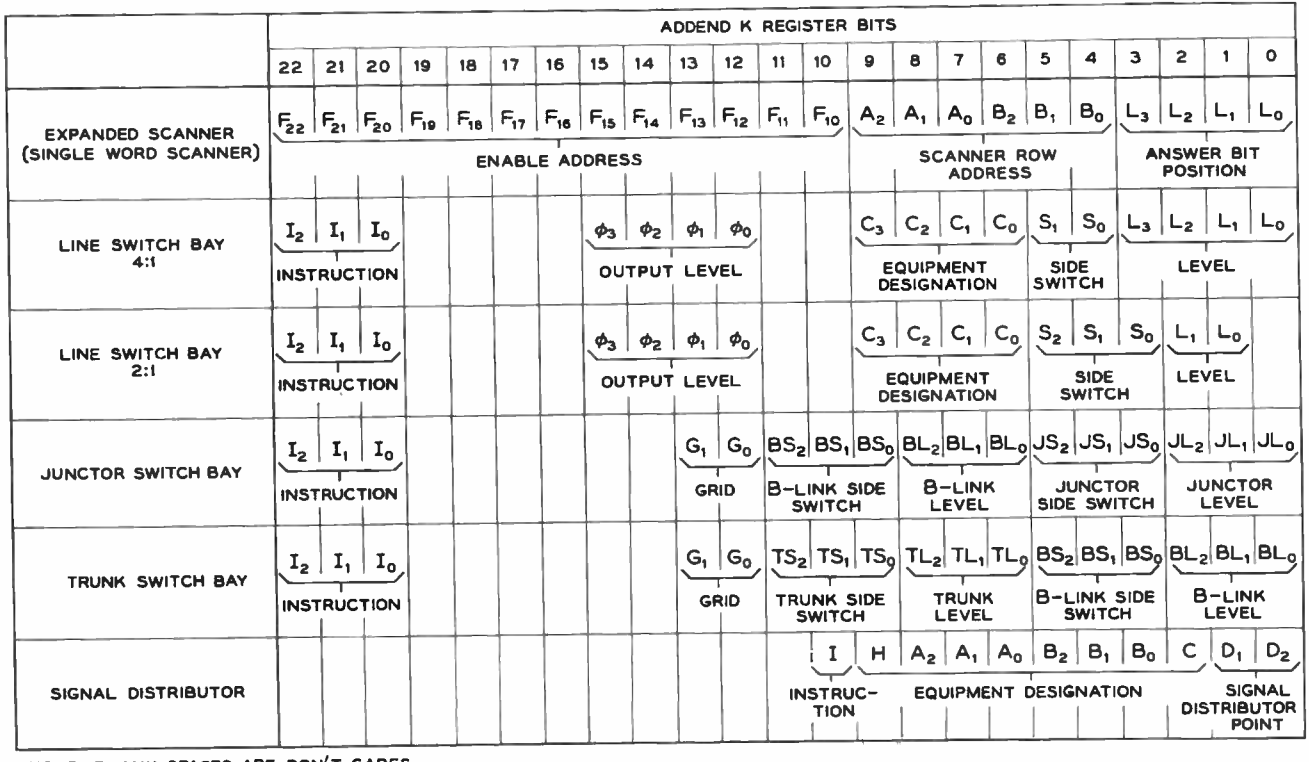


Fig. 9 — Peripheral address bus bit assignments.

the nonhomogeneity of the units connected to it. A given peripheral bus operation takes two machine cycles (11 microseconds). Subsequent peripheral operations could go to entirely different types of units: in fact, bipolar (nonperipheral unit) operation could alternate with peripheral unit operation. Thus each peripheral bus cycle can and must be completely independent of any preceding or subsequent operation. The manner in which units are selected, i.e., unipolar enable (see Fig. 7), also determines the bus configuration with respect to the peripheral unit. This is unlike the program and call store bus configuration control. In those systems the bus configuration is independent of the unit selection and the route does not change with the memory unit address. The peripheral system selects the bus configuration while selecting the peripheral unit. The bus configuration between the central controls and the central pulse distributors is selected by one flip-flop located in the central control. This is the CPDB flip-flop, and once set it remains set for many operations until changed via a maintenance configuration control program. At the start of peripheral operations the state of this flip-flop is transmitted to all central pulse distributors, telling them which of the two buses will be used for a subsequent address transmission.

The central control F register usage is the key to the peripheral system control. For each peripheral operation, two binary words must be used



NOTE: BLANK SPACES ARE DON'T CARES

Fig. 10 — Addend K register bit assignment.

by the central control. One is the internal instruction for the peripheral unit and is placed in the central control addend K register; the other word is the peripheral unit enable word and is placed in the central control F register. On nonperipheral unit operations, when only a central pulse distributor output is to be activated, the address of the point is placed in the central control F register and the addend K register is not used.

The F register word contains 23 bits. Bits F_{10} through F_{13} control the execute translator and select the proper central pulse distributor. Bits F_{14} through F_{22} combined with F_9 are the ten binary bits from which the central pulse distributor point address is generated. The use of the F_9 bit is not direct. It becomes part of the address only for non-enable operations. For enabling operations, F_9 is considered equal to zero regardless of its actual value. For non-enable operations, it assumes its actual value. This bit is the one used to select which half of the central pulse distributor matrix is used.

Those central control instructions used for peripheral operations are treated as enable or non-enable instructions. The central control instruction decoder must determine which it is and whether or not a peripheral answer is expected if it is an enable-type instruction. Whenever a peripheral answer is expected, a special flip-flop designated scanner request flip-flop (SCR-FF) is set. When no answer is expected, it is reset. Whenever a non-enable operation is to be performed, another special flip-flop designated the F inhibit flip-flop (FINH-FF) is set. It is also set for a class of single-word scanner instructions (see Fig. 10). It is reset for all other peripheral instructions. The F_9, F_8 and F_7 bits are used to select the peripheral address translator for all enable operations except the single-word scanner instruction. The codes for translator selection are given in Table II. The F_6 bit is used to determine whether or not a verify address is expected from the central pulse distributor. The F_5 bit is used

TABLE II — TRANSLATOR SELECTION CODING

F_9	F_8	F_7	Translator
0	0	0	short binary to peripheral bus
0	0	1	long binary to peripheral bus
0	1	0	line switch bay, 4:1 concentration
0	1	1	line switch bay, 2:1 concentration
1	0	0	junctor or trunk switch bay
1	0	1	long binary to CPD address bus
1	1	0	signal distributor
1	1	1	scanner

to determine whether or not to expect an all-seems-well scanner response. The F_4 through F_0 bits are reserved for maintenance use and exert no control over peripheral operations. Upon detecting a peripheral operation, the central control instruction decoder either sets or resets the SCR and FINH flip-flops and starts a peripheral control sequence circuit. This sequence circuit will control all peripheral operations. This sequence circuit uses the SCR and FINH flip-flops as well as many of the F register bits to control peripheral operations. The use of the F register as functions of the SCR and FINH flip-flops is as follows:

(1) FINH = 0 and SCR = 0 or 1, use the F_9, F_8 and F_7 bits to select the peripheral translator,

(2) FINH = 1 and SCR = 1, use the scanner translator for the peripheral address bus, ignore the condition of F_9, F_8 and F_7 bits with regard to selection of the translator. For this particular case, as can be seen from Fig. 10, these bits are part of the scanner address.

(3) FINH = 1 and SCR = 0, no use is made of the peripheral address bus, and the F_9 bit is used as part of the central pulse distributor address selection.

The bus-to-controller path for a peripheral unit is selected by the enable pulse that it receives. A typical two-controller unit has four possible enable pulse inputs, two from each central pulse distributor of a duplicate pair (see Fig. 7). Bit position F_{10} of the central control F register is the least significant of the four central pulse distributor select bits. It is used to select between a duplicate pair of central pulse distributors. The F_{14} bit of the same register is used to select between halves of the enable portion of the central pulse distributor output matrix. Bits F_{10} and F_{14} control the selection of the one-out-of-four unipolar enabling distributor output points, two of which are located in each central pulse distributor of a duplicate pair. Table III shows the route selection for all such peripheral units. From this table it is clear that F_{14} is also used to select the proper address bus for transmission.

The F register word is called the "enable address," since it chooses the address bus and indirectly the proper controller. This word can differ

TABLE III — PERIPHERAL UNIT ROUTE SELECTION

F_{14}	F_{10}	CPD of a Pair	Peripheral Unit Controller	Address Bus
0	0	A	A	A
0	1	B	B	A
1	0	A	B	B
1	1	B	A	B

from unit to unit. For this reason, the normal procedure is to "look up" the enable address in a call store memory table. This table is termed the "enable" table.

When performing non-enable operations, the F_{14} bit is still used to control the selection between halves of whichever portion of the central pulse distributor matrix the F_9 bit selects. As such, the F_{14} bit is used to determine whether or not a bipolar output pulse is the set or reset polarity. If a pair of unipolar outputs is used for such control, the F_{14} bit is used to choose between them.

Although the major peripheral bus configuration control takes place via the same mechanism as the unit selection, there are several operating modes which affect this bus configuration. Mode-control flip-flops are used to allow additional operating modes. There are two mode flip-flops in each central control for central pulse distributor address bus control (see Table IV) and two for peripheral address bus control (see Table V).

In order to properly diagnose certain classes of faults, it is necessary to provide independent operation of central controls. This independent mode is referred to as "off-line" operation. The two central controls can work with independent memory units. However, many of the peripheral units are common matrix units, and therefore two nonsynchronized machines cannot be given simultaneous control of the peripheral system. It is still necessary, however, to allow off-line operation of the peripheral system. This is implemented using two flip-flops. These are located in each central control. The central controls react to these flip-flops as shown in Table VI.

Fig. 11 is a timing diagram for peripheral operation. All transmissions in the program store and call store system were accompanied by sync pulses. These are used to reduce the time the receivers on the buses are

TABLE IV — CENTRAL PULSE DISTRIBUTOR MODE CONTROL

CPD Bus Mode Control Flip-Flops		Mode
CDMA	CDMB	
0	0	normal[1]
1	0	mode A[2]
0	1	mode B[3]
1	1	not used

[1] Normal mode: only active control unit transmits over bus designated by CPDB flip-flop.
[2] Mode A: only active unit transmits over both buses.
[3] Mode B: active unit transmits on bus designated by CPDB flip-flop — standby unit transmits on other bus.

TABLE V — PERIPHERAL BUS MODE CONTROL

Mode Control Flip-Flops		Mode
PBMA	PBMB	
0	0	normal[1]
1	0	mode A[2]
0	1	mode B[3]
1	1	not used

[1] Normal mode: only active central control transmits over bus designated by bit 14 of F register.
[2] Mode A: active unit transmits on both buses. Standby does not transmit on either bus.
[3] Mode B: active unit transmits on bus designated by bit 14 of F register; standby unit transmits on other bus.

open and susceptible to noise pulses. The peripheral bus system uses no sync pulses. Some noise protection is afforded, however, through the use of other techniques. When a peripheral unit is enabled by a central pulse distributor unipolar pulse, the mechanism that recognizes the enabling is also used to open a "window." The address for this unit must arrive on the peripheral address bus during the time this "window" is open. The window interval is 2.5 microseconds maximum. Thus, the address and enable signals must arrive at the peripheral unit within a narrow band of time coincidence. Two factors control this coincidence. One is the time of transmission from the central control; this can be strictly controlled. Another is the cabling of the address bus and the enable leads throughout the office. To control the differential between the address and enable route in all sizes of offices, the central pulse distributor frame is used as a distribution center for the peripheral address bus. In this way the distance, and therefore the delay, between the point of transmission, the central control, through the central pulse distributor frame to the peripheral unit is the same for both the address and enable signals. Fig. 12 shows

TABLE VI — OFF-LINE OPERATION

Off-Line Control Flip-Flops		Active Unit	Standby Unit
OL1	OL2		
0	0	E & A	nothing
0	1	E	A
1	0	A	E
1	1	nothing	E & A

E = Enable operation.
A = Address operation.

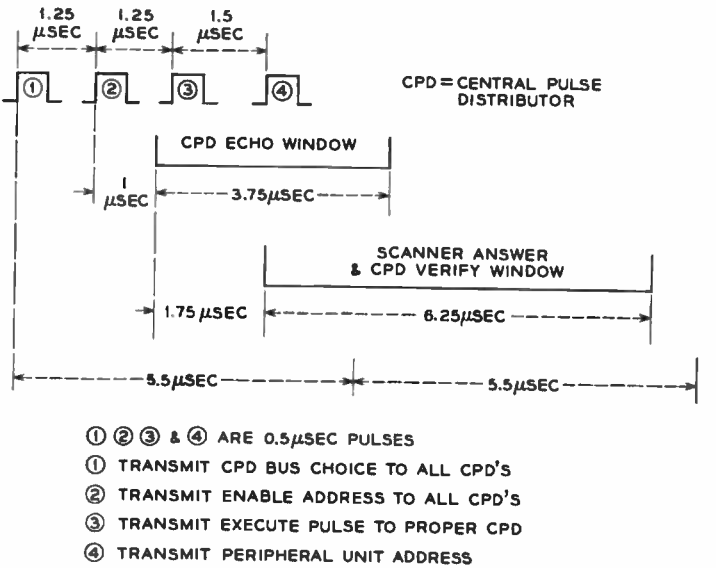


Fig. 11 — Peripheral system timing.

the arrangement in a central pulse distributor. There is provision for four branch address buses in each central pulse distributor. If a second central pulse distributor pair is required for enabling purposes, then the branch address bus used for the units enabled by the additional central pulse distributor pair must emanate from the new central pulse distributor frame.

The peripheral answer bus also passes through the central pulse distributor frames. Within these frames, the answer buses are "fanned in" from a maximum of four to one main answer bus. The main answer bus connects to the central controls. The branch address and branch answer buses give the peripheral system greater flexibility than was required with the program store and call store systems. This flexibility was not needed in these cases since there was a limited number of units on each bus system. With the peripheral bus system, the number of physical frames can run as high as 400 or 500.

With both the address and answer buses, the 0 bus is both fanned in or fanned out at the 0 central pulse distributor of a pair and the 1 bus at the 1 central pulse distributor of the pair.

In the central pulse distributor, there are many output points associated with functions other than the enabling of peripheral units. This was mentioned earlier. Some of these output points control critical

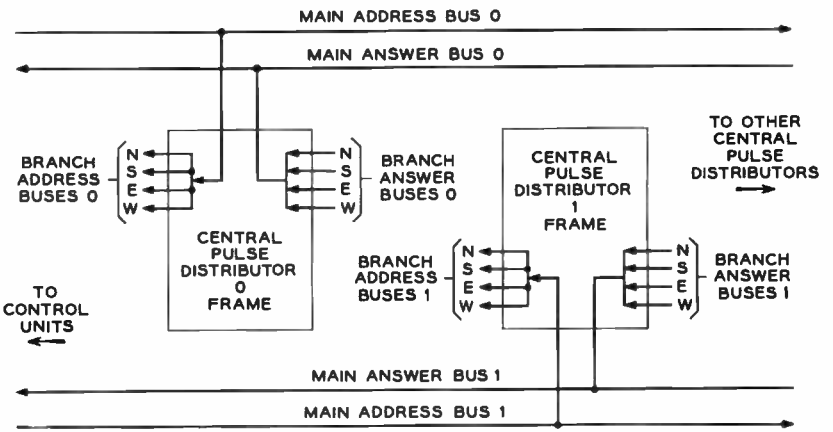


Fig. 12 — Address bus fan-out and answer bus fan-in.

system flip-flops — for example, which central control is active and which is standby — as well as the call store and program store route flip-flops. An unwanted change of these flip-flops could be caused by random noise which may appear on the wires connecting them to the central pulse distributor. A means of protecting such flip-flops has been incorporated into the system. This consists of a common synchronizing signal sent to all such flip-flops simultaneously with the central pulse distributor output pulse. Only the active central control can transmit this pulse. This common signal is designated WRMI ("we really mean it"). It is transmitted from the central control over two twisted wire pairs for duplication and synchronizes in the central pulse distributor frames with the central pulse distributor execute signal. It is fanned out much the same as the address bus.

Fig. 13 shows the relative occurrence of operations on the three bus systems. All call store and peripheral bus operations are controlled by program instructions. The program store bus system is controlled by program instructions only for transfer instructions and data readings.

Because of the necessity for speed and the use of each bus system for round-trip communications, the propagation time, and hence the length, of the buses becomes a critical factor. In the No. 1 ESS system the program store bus is limited to 100 feet and the call store bus to 125 feet from the farthest central control to the farthest memory unit. It was mentioned earlier that two bus cables are provided in both the program and call store bus systems. This is accomplished by combining two unidirectional answer buses at the central controls. The address bus is bidirectional.

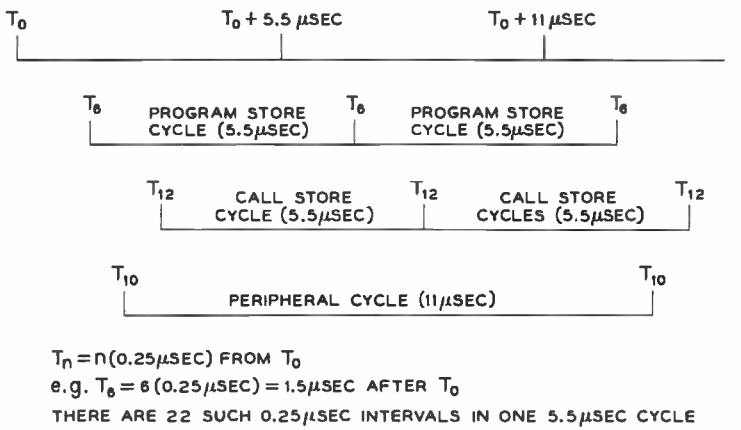


Fig. 13 — Basic communication timing.

The peripheral bus system is used to communicate with many more units than either of the memory bus systems. As a result, longer bus lengths must be used. In addition, two cycles, or 11 microseconds, are used for peripheral bus operations. The distance from the farthest central control through the bus fan-out points to the farthest peripheral unit can be no more than 450 feet. An equally critical factor in the peripheral system is the differential distance between the address bus and the enabling path for the peripheral unit. This differential is minimized by controlling the cable routes.

III. BUS CIRCUITRY

3.1 *The Peripheral Address Bus*

3.1.1 *Typical Pair*

Consider a typical pair in the peripheral address system, starting at one of the central controls. The twisted pair is in a standard switchboard cable and is made of 26-gauge wire with approximately one twist every 2.5 inches.

In a small office one end will go up from a central control to the cable rack, into a special shielded compartment, and over the row of frames to an end guard at the end of the row. Within this end guard are banks of resistors. The address pair terminates in one of these resistors.

The other end goes to the other central control, also by way of the cable rack. After going into the second central control and out again, it

goes, via the cable rack (as always), to a central pulse distributor. Here it has an option. In a large office it would go in and out of a CPD and might go on to do the same at several others. In the small office it is simply connected to a pair of terminals of the CPD, but does not actually go in (see Section 3.7), and then continues in turn to all the peripheral unit frames which contain scanners and network controllers. Beyond the last controller or scanner the other end terminates in a resistor on a nearby end guard.

Before considering the associated hardware it may be noted why a twisted pair was chosen. From the previous discussion, it is evident that this is a transmission line with primary considerations of transmission loss and speed, immunity to noise, reflections and crosstalk, and low cost. One obvious possibility, because of its excellent transmission properties and shielding, is coaxial cabling. However, the cost of coaxial cable is high* and the costs of terminating coaxial wiring are worse. Fortunately, twisted pair, carefully balanced and properly terminated in 100 ohms, has characteristics which, for our purposes, are satisfactory. At the frequencies of interest, the effective delay is 2 μsec/1000 feet. The loss varies with frequency (6 db/1000 feet at 1 mc), producing the distortion illustrated in Fig. 14. As will be noted, tolerable delay limits the length to less than 450 feet where the loss and distortion are relatively small. As will be discussed later, reflections, noise, and crosstalk can be held within satisfactory limits.

3.2 *Grounding Inductance*

Precise balancing to ground is essential to the good noise and crosstalk characteristics required.

The balanced grounding cannot be done satisfactorily by means of the terminating resistors. This would require an expensive pair of matched resistors at the terminals. Instead, a 0.25 mh inductor is used, connected across the pair and grounded at the midpoint. A much better balance is obtained than is practical with resistors, and the ground can be located at the most desirable point near the middle of the longest buses.

The inductor is built into terminal blocks where the pairs are brought in or out of a particular frame.[4]

3.3 *Cable Receivers*

Since this pair transmits a signal from a central control to a large number of receivers in the frames which contain network controllers,

* Even in a relatively small office, the 38 pairs in the address bus alone may add up to about 3 miles of pairs and some 3000 terminations, so interconnections are an appreciable part of the cost of an office.

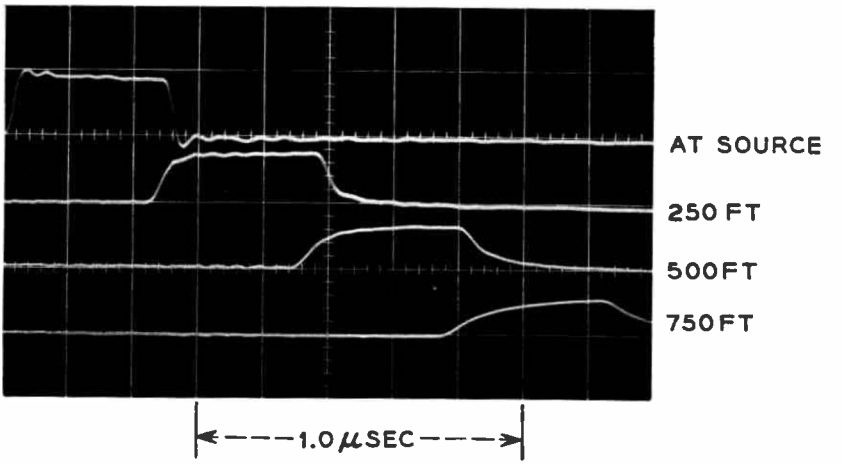


Fig. 14 — Pulse distortion by transmission.

scanners, and signal distributors, it is essential that the individual receivers should not seriously degrade the transmission characteristics of the pair. In addition, a failure in any one receiver should not interrupt transmission to the other locations.

To meet these requirements, a small current transformer is used for pick-off. The primary winding is essentially the bus pair passing through a toroidal core in such a manner as to constitute a balanced, two-turn primary winding. The secondary is a 50-turn winding.

As shown in Fig. 15, the secondary winding goes to a simple grounded-emitter transistor amplifier.

This receiver picks off not more than one per cent of the power being transmitted in the pair and amplifies it to a useful level for use in the frame. This circuit adds a small loss in the balanced pair when operating properly, and nothing which can happen in the output side of one or two pick-offs will seriously damage transmission to other frames. Under normal conditions each pick-off inserts a balanced impedance, less than one ohm, into the line. If the secondary of one transformer is either shorted or opened, the effect on transmissions is a small change in the inserted impedance. In the worst case (open secondary) the inserted impedance in the line is 3 μh. This is harmless if only one or two are open circuited.

3.4 *Cable Drivers*

The standard pulse on the pair is produced by a cable driver. As may be noted in Fig. 16, this is a two-input logic circuit.[5] It is designed to

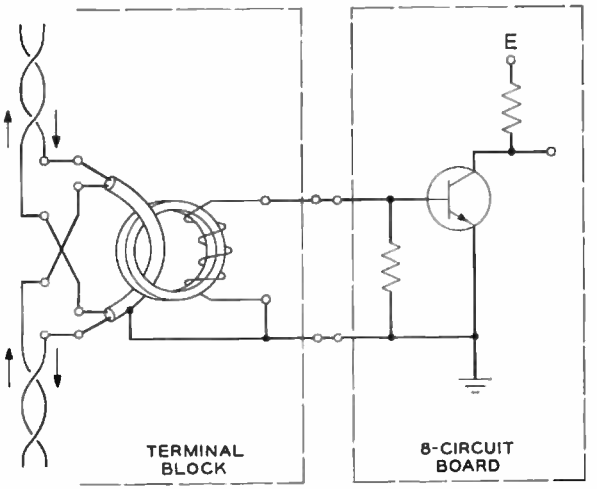


Fig. 15 — Pick-off transformer and cable receiver amplifier.

operate at a higher level than the normal logic circuitry. The two inputs are driven in coincidence by a pulse from standard logic circuits and a timing, or gate, pulse. The output transformer, designed to work into 50 ohms, is connected across the transmission line, driving the bus in both directions.

The requirement that two or more drivers be connected to the same pair and drive at a level adequate for operation of many receivers made the shunt connection the only practical one. The transformer design was the most exacting in this circuit. It was necessary that a physically small

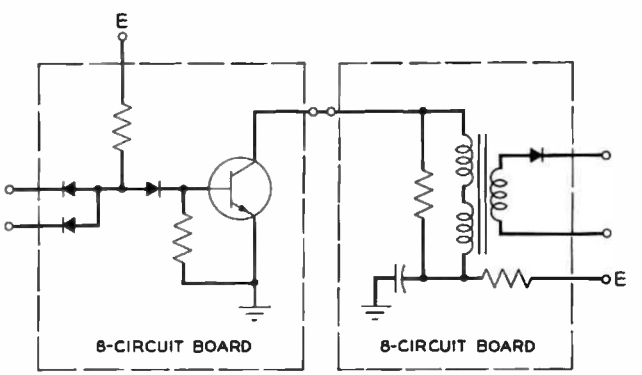


Fig. 16 — Cable driver amplifier and output transformer.

unit (mounted on a standard printed wiring board) meet the requirements: drive a 200-ma, 0.5-μsec pulse into a 50-ohm load on a 10 per cent duty cycle. A 2:1 turn ratio is used, with 3 equal windings, to limit the transistor current to 100 ma. The inductance per winding is 1.35 mh. There were two serious problems. The impedance looking back into the transformer was not large enough to prevent deterioration of the transmission when several circuits were across the same line. This was solved by means of a series isolating diode. A second problem was a sharp reverse spike, caused by leakage inductance, at the end of an output pulse. This spike temporarily increased the transistor collector voltage above breakdown and, in extreme cases, produced noise in other channels where amplifiers were on the same board. These defects could not be cured by decreasing the magnitude of the primary damping resistor without increasing the transistor load and lengthening the transformer duty cycle. The problem was solved by trifilar winding. The result was a slightly increased capacitance (which was desirable here), a leakage inductance cut to about one-third its previous value, and a unit which could be much more readily manufactured to specifications.

3.5 *Noise, Distortion, and Crosstalk*

Before going to more specialized circuitry which is associated with the bus system, the situation on distortion, noise, and crosstalk may be summarized.

As noted previously, delay requirements limit cables to less than 450 feet long. Over that distance the transmission distortion is small. There is one difficulty. While the amplifiers regenerate and transmit a good pulse, they also tend to lengthen it. This becomes serious in a large office where more amplification is used. It will be discussed later.

Putting the buses in a separate section of the cable rack, shielded from the rest of the world, is, in a sense, extra insurance against noise as far as the balanced line itself is concerned. Extensive tests have been made simulating noise (such as that from unprotected relays) which might be encountered. No trouble was detected from the transmission line itself. Longitudinal noise did, however, leak into the cable receiver. The cable receiver transformer transforms from balanced to grounded transmission. At high frequencies the parasitic capacitance ruins the balance, and so a high-level, sharp spike of longitudinal noise could be transmitted through the cable receiver. This problem was solved by shielding the primary winding as shown on Fig. 15.

The greatest remaining source of noise and crosstalk is within terminal

circuitry rather than the bus itself. The circuitry is closely packed, eight circuits to a board on closely packed boards. All circuits under such conditions are prone to noise and crosstalk because of coupling and common ground connections. The amplifier circuits, because of the large currents involved, are particularly so. The solution required careful design (and redesign!) to keep common ground leads short and isolate noisy output leads from sensitive input ones.

3.6 *Grouping of Drivers and Receivers*

Closely analogous to noise problems are those concerned with multiple drive points and multiple reception. As described, an address pair has two or more cable drivers, grouped in the control area, any one of which may drive a multiplicity of receivers in the peripheral area. In a large office there may be up to 50 receivers scattered along each half of the pair with the drivers grouped together in the middle.

There are also answer buses which transmit from a multiplicity of peripheral points to a few receivers in the control area. Here a one-sided bus must be used. Drivers cannot be put on both sides of a receiver. The necessary polarities are such that a pulse from a driver on one side passes through the isolating diode of a driver on the other side in the forward direction. It can temporarily break down the corresponding transistor, or turn it on, producing noise and distortion on the pair. In the case where the address bus goes two directions, two answer buses must be used, going to separate receivers.

3.7 *Central Pulse Distributor*

There are two distinct parts to the central pulse distributor. One[6] decodes an enable address and sends an enable signal on private pairs to other frames. The other part is the one to which the peripheral address pair goes. This section contains a group of amplifiers which regenerate the address pulse and can transmit it simultaneously to a maximum of four outputs. In a small office this amplifier is not needed and this section is wired but not equipped. The address bus is brought only to the CPD terminals so that it will be available if the office grows.

If the expansion in the number of receivers or length of bus exceeds the capacity of one bus, then the CPD is equipped. The pulse is picked off through a standard cable receiver which drives a bus fan-out circuit (see Fig. 17). This circuit simply transformer-couples the input of two transistor amplifiers in parallel to the output of a receiver amplifier. These amplifiers, through standard transformer circuitry, drive two pairs.

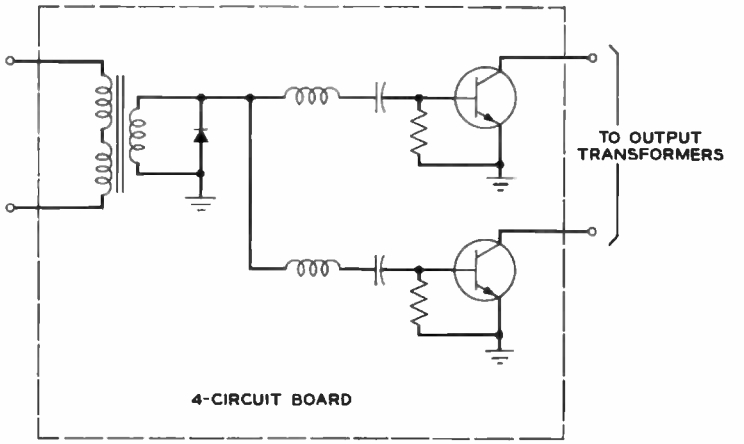


Fig. 17 — Bus fan-out circuit.

Since each end of each bus can be used, this gives fan-out from one bus to four in parallel.

The addition of this amplification results in a difficulty previously mentioned. Each time the pulse is regenerated, the output lengthens a little. At this point there is danger that the accumulated stretching will give a pulse longer than the 0.75-μsec maximum permitted in the peripheral area. To avoid this, the pulse width as well as amplitude is regenerated. *LC* circuits in series with the transistor inputs convert the input pulses to a half sine wave with 0.5-μsec duration. The amplifier clips this, producing a new 0.5-μsec pulse.

One further note may be made on bus length. The delay which may be tolerated is measured from the first cable driver in the control area to the farthest cable receiver. The 450-foot maximum length must be measured in the same way from the control center, not the CPD. Since the corresponding enable pulse must coincide with the address at the receiver, the buses involved must be of equal length, within 50 feet. To insure this, they follow the same path, as far as possible.

IV. OTHER BUSES

4.1 *Answer*

The answer bus sends signals between the same units as does the address bus, but in the reverse direction, so it has many cable drivers scattered along it and a few receivers in the control area. As noted, the

characteristics of the cable drivers make this a one-sided bus — drivers cannot be located on both sides of the receivers. It uses exactly the same circuitry as the address bus, with one exception. In the large office the answer bus must fan in four-to-one. This is done using a combination (Fig. 18) of receivers, logic circuits and drivers in an OR configuration.

4.2 *Enable Address*

The enable address is decoded in its section of the CPD, and the terminating circuitry is somewhat different. The input from the enable address bus to the CPD's is picked off in the manner already described, but the outputs are on a one-to-one basis, each enable pair going to one receiver. These pairs are transformer-coupled in special circuitry[6] within the particular frames. As discussed in the reference, there are several types of these pairs and they may transmit the standard pulses or a bipolar pulse. The enable verify is an exception to the rule that buses are

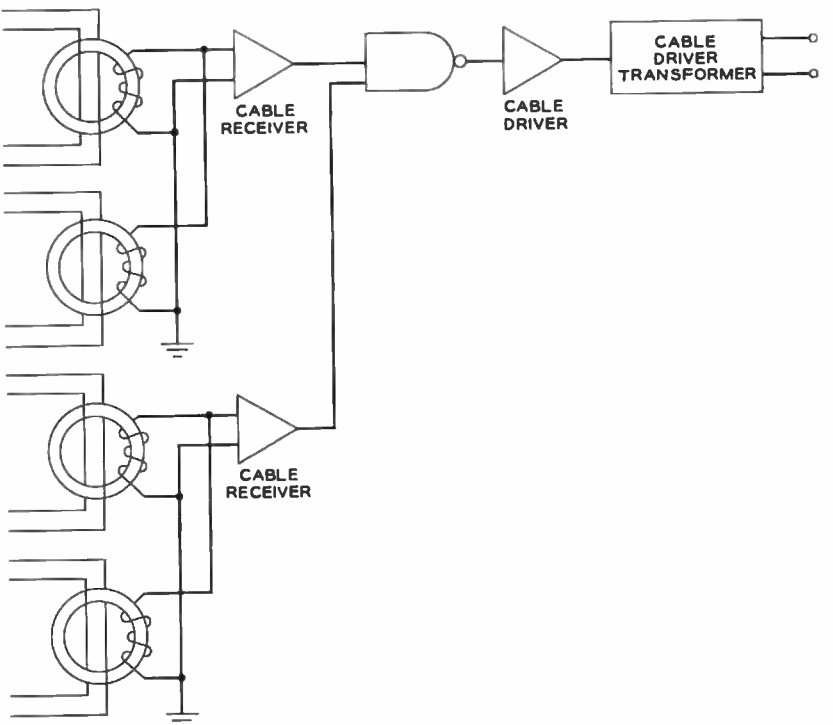


Fig. 18 — Bus fan-in circuit.

all one-way. When an enable pulse is transmitted, a verify is returned (after a brief delay) to the CPD over the same pair, using special circuitry.[6] From the CPD the verify pulse gets to the control center over an enable verify bus in the standard manner.

4.3 *We Really Mean It and Set Manual Pulse Bus*

There are many private pairs, similar to the enable, which are used to enable or disable various sections of the office — generally by operating flip-flops. These are operated by pulses generated in the enable section of the CPD's and the coincident gate (WRMI), which is a simultaneous pulse produced synchronously from a special fan-out circuit. Three of the four outputs are always used even in a small office:

WRMIA goes to master control center, master scanner and central control

WRMIC goes to call stores

WRMID goes to program stores.

All use standard drivers and receivers.

A special fan-out circuit is used because the WRMI pulse must be stretched when it arrives at the CPD before being gated out.

The pulse is received by means of a cable receiver pick-off and amplifier (Fig. 19). This drives a dynamic register[6] which stretches the pulse to 2 μsec. During this interval the four WRMI output pairs may be pulsed coincidentally with the pulse from the enable section of the CPD. This is done by standard cable driver circuitry. The additional cable driver shown on Fig. 19 is needed to correct the polarity.

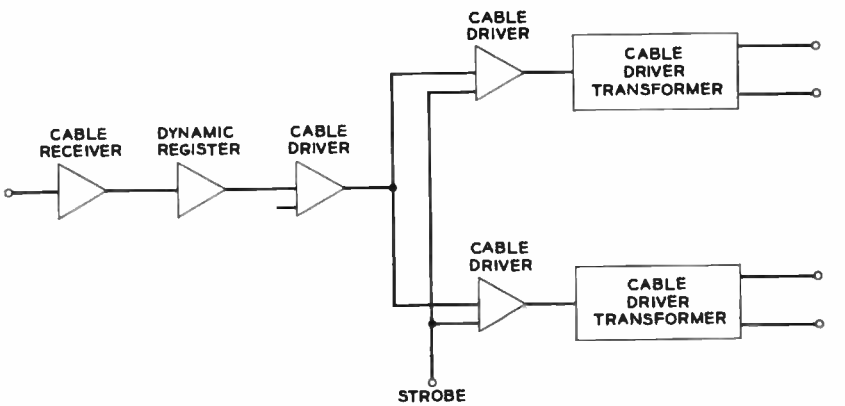


Fig. 19 — WRMI fan-out circuit.

The SMPB pair is discussed with WRMI only because it also is a lone pair — a gate or execute driven from the master control center and having receivers at central controls and program stores.

4.4 *Stores*

The individual enable and status connections from the CPD's to the stores have been mentioned. The other major interconnections in this area are the call store or program store address and answer buses. For timing reasons these must be kept short — not more than 100 feet in length or more than 24 receivers per bus. There is the same sort of limitation as on the peripheral address and answer. Drivers in the control area can operate both ends of the address bus, but the answer bus must be one-way. Drivers at the stores must all be on the same side of the receivers in the control area. Standard cable drivers are used for these buses. Standard receivers are used at the central control. At the stores the cable receiver current transformers are standard, but the amplifiers are part of a more complex circuit.

4.5 *Private Buses in the Control Area*

Most of the pairs between the central controls for maintenance and diagnostic purposes use standard drivers and receivers. There are also timing (clock pulse) pairs which have carefully controlled transmission characteristics. These connect directly to special circuitry within the central controls.

V. DC CONNECTIONS

There are three types of these:

5.1 *Master Scanner*

This collection of connecting pairs is used to observe the status of test points scattered throughout the office. The scanner itself is not very sensitive to noise but, because the leads are long and are exposed to many sources of noise, there is a danger that they might transmit noise to another sensitive circuit. For this reason, twisted pairs are used and the leads are carried in their own shielded section in the cable rack. They are of course isolated from the frame ground, the ground return being brought back to the power ground for that circuit.

5.2 *Trunk Circuitry*

The interconnections for the trunk circuit relays are also twisted pairs grounded in the same manner as the master scanner pairs. They are also, as far as practical, balanced to ac to suppress relay noise.

5.3 *Maintenance and Control Circuitry*

There are a number of relays used for maintenance and control purposes. These are operated by dc connections similar to those of the trunk circuitry.

VI. OPERATING EXPERIENCE

As previously noted, extensive laboratory tests were made, simulating the worst predictable conditions. Troubles were detected (such as the need for shields on the pick-off transformers) and eliminated.

The central control and associated buses were operated in and close to an operating, electromechanical central office (even with protection taken off adjacent relays) without failure.

The most extensive and significant experience is with the actual offices — with the Holmdel, N. J., systems laboratory, which is a small office with a complete interconnecting bus system, and with the first commercial office, installed and now under test at Succasunna, N. J.

Testing of an assembled office was begun in Holmdel in April, 1963. It has progressed to the state that calls have been set up through it. Testing of the Succasunna office started in August, 1963, and is continuing. Most of this testing uses the interconnecting buses, so this part of the office has been thoroughly tested.

These tests prove that this is a working system which should operate satisfactorily in commercial service.

VII. SUMMARY

The complex network of transmission lines described here is organized to meet the objective — the rapid exchange of a large amount of information (in digital form) between many functional units. The organization and operational modes result in flexible, dependable and efficient use of hardware. Simple, reliable hardware is used with a high degree of standardization, resulting in over-all economy and reliability.

REFERENCES

1. Keister, W., Ketchledge, R. W., and Lovell, C. A., Proc. I.E.E., **107**, Part B, Supplement, Nov. 20, 1960.

2. Ault, C. F., Gallaher, L. E., Greenwood, T. S., and Koehler, D. C., No. 1 ESS Program Store, B.S.T.J., this issue, p. 2097.
3. Genke, R. M., Harding, P. A., and Staehler, R. E., No. 1 ESS Call Store — A 0.2-Megabit Ferrite Sheet Memory, B.S.T.J., this issue, p. 2147.
4. Chevalier, J. G., and Eisenhart, R. K., No. 1 ESS Circuit Packs and Connectors, B.S.T.J., this issue, Part 2.
5. Cagle, W. B., Menne, R. S., Skinner, R. S., Staehler, R. E., and Underwood, M. D., No. 1 ESS Logic Circuits and Their Application to the Central Control, B.S.T.J., this issue, p. 2055.
6. Freimanis, L., Guercio, A. M., and May, H. F., No. 1 ESS Scanner, Signal Distributor, and Central Pulse Distributor, B.S.T.J., this issue, Part 2.

No. 1 ESS Logic Circuits and Their Application to the Design of the Central Control

By W. B. CAGLE, R. S. MENNE, R. S. SKINNER, R. E. STAEHLER and M. D. UNDERWOOD

(Manuscript received January 9, 1964)

The central control is composed of thousands of relatively simple logic circuits, intricately interconnected to perform the primary data processing functions of the No. 1 electronic switching system. This paper discusses both the circuit development of the basic modules and their application to the logic design of some of the high-speed arithmetic circuits required in the central control. Special electrical design features, wiring rules, and circuit pack assignment procedures are also described.

I. INTRODUCTION

Companion articles in this issue[1,2,3] discuss the logical organization and diverse responsibilities of the central control (CC) as the primary information processing unit of the No. 1 electronic switching system.

This paper has the objectives of describing the basic building-block circuits which constitute the framework of the central control and illustrating their logic design applications in some of the major functional portions of the unit.

The presentation is divided into three parts:

Sections II–VI deal with the parameters of the semiconductor devices and the characteristics of the basic circuit in which they are integrated.

The logical effectiveness of the resultant configuration is demonstrated in Sections VII and VIII, which trace its applications from fundamental designs to some of the more complex, specialized functions required in the CC.

Section IX discusses circuit packaging techniques and over-all equipment development as applied to the central control.

II. LOGIC CIRCUITS

2.1 *General*

Numerous factors ranging from system philosophy to device physics were considered in the process of selecting the basic semiconductor logic circuits for No. 1 ESS. The paramount objective in this process was to achieve versatile and reliable circuit performance at an over-all minimal cost. In such a large, complex information processor this can be realized only by a concurrent minimization of costs in the broad areas of design, manufacture and maintenance.

2.2 *Building-Block Approach*

The search for the common denominator of minimal cost in these broad areas highlighted the need for a standardized set of building-block logic circuits, not only for use in the central controls, which have the largest concentration of logic circuitry, but also for use throughout the entire system.

Standardization provides the system logic designer with a set of "black boxes" that can be easily interconnected to perform any complex logic function. This facilitates design by reducing the time and cost involved in specifying the logical fabrication of the system.

The minimization of the varieties of building blocks, with their correspondingly higher levels of production, translates to reduced manufacturing costs. Concurrent characterization of devices suitable for many different applications within the system reduces the number of device codes and further benefits the economics of production.

Finally, the advantages to the system user in terms of trouble diagnosis and reduced maintenance costs, particularly in the stockpiling of replacement packages and devices, is considerably improved.

2.3 *Reliability*

The electronic circuits in No. 1 ESS must be intrinsically reliable if the over-all system is to provide uninterrupted telephone service. This means that the components used in the logic circuits must be inherently reliable of themselves and used within conservative limits.

Duplication of major system units is the most important safeguard of reliable, continuous service. To prevent the loss of system control, the central control is fully duplicated. Both units of a duplicated pair contain a large number of circuits whose functions are devoted to continually monitoring all internal data processing and cross-comparing

their operations. Even so, the requirement[1] of a 40-year mean time to the simultaneous failure of duplicated units imposes stringent upper bounds on the allowable failure rate of the basic logic circuits.

To help visualize the circuit failure rates implied by these requirements, the required mean repair time in hours has been plotted against the number of failures per 10^9 circuit hours in Fig. 1.

To achieve what has been established as a reasonable mean repair time for a CC of from one to three hours (including both fault location and replacement times) it is apparent that there must be in the order of 50 to 100 failures or less per 10^9 circuit hours. For individual logic circuits, the semiconductor failure rates must then fall in the range of 1 to 10 failures per 10^9 device hours to meet these objectives.

2.4 *Switching Speed*

The No. 1 ESS system organization requires a number of units to function in real time and share a single control unit. This necessitates the use of fast switching circuits in the control unit in order not to limit the system capability. In switching circuit design, trade-offs are often made between long logic chains which require a small number of logic elements and larger, more complex circuits which have less delay.

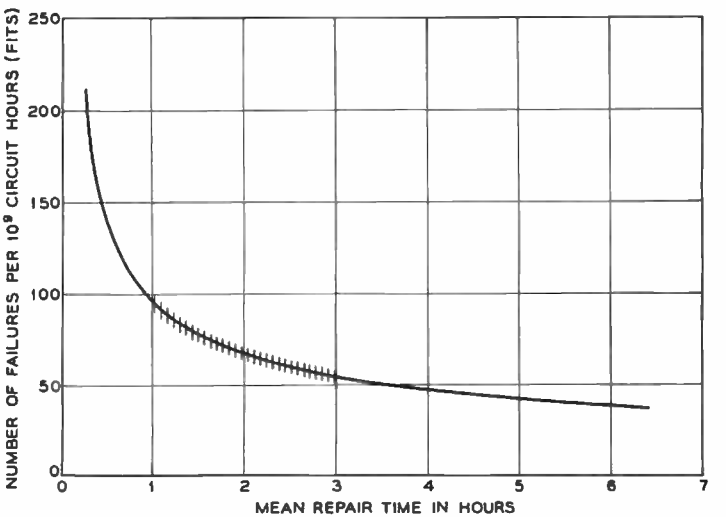


Fig. 1 — Logic circuit failure rate vs mean repair time. Assumptions: duplicated control systems, each containing 12,500 circuits; 40-year mean time to simultaneous failure of duplicated units. Typical circuit elements: 1 transistor, 4 diodes, 4 resistors, 16 soldered connections, and 4 pressure contacts.

Faster switching speeds in the individual devices alleviate the problem considerably, but this is countered by the increasing cost of the devices. These factors are all weighed in deciding upon a satisfactory operating speed.

2.5 *Environment*

For over-all system economy, the logic circuits are designed to utilize the common central-office battery power sources. Since these voltages can decrease by as much as 20 per cent during commercial ac power failure, the logic circuits must be relatively insensitive to supply voltage variations.

The logic circuits are designed to operate reliably over the temperature range of 0–55°C.

2.6 *Silicon Semiconductors*

The low leakage currents and high allowable junction temperatures of silicon devices as compared to germanium have resulted in the specification of silicon semiconductors for all circuits in No. 1 ESS. This eliminates the necessity for an air-conditioned environment. The greater tolerance to temperature variations results in smaller devices for the same power rating, allows higher packing densities for the equipment with a corresponding reduction in lead lengths and floor space, and allows the use of wave soldering on printed wiring boards. Low storage times permit high-speed operation with relatively simple circuit designs. The slightly higher voltage drop characteristic of silicon devices is easily circumvented.

III. SELECTION OF A BASIC GATE

3.1 *Type of Function*

There are many different types of semiconductor logic gates which fulfil in varying degrees the basic requirements outlined in the preceding sections. The selection of the optimum gate for a particular application depends on the relative weight given to factors such as switching speed, component and assembly costs, reliability, allowable tolerances of components and voltages, minimization of device and package codes, logical flexibility, noise margins, and many others.

The building-block logic gates for a system must, first of all, be capable of being interconnected to perform any complex logic function which may be required by the logic designers. This requires that a basic set

of functions be provided. Historically, AND and OR gates together with both inverting and noninverting amplifiers and memory cells (flip-flops) were first used. This was the case in the experimental Morris, Illinois, electronic switching system.[4]

Recently, more complex functions have been combined in the basic block — for example, the various forms of AND-NOT and OR-NOT. These are more efficient in that any one can be used exclusively to construct a complete logic system. A further benefit is that these circuits usually insert voltage or current gain in every logic stage, reducing the variations in impedance levels in a complex circuit and thereby improving speed and crosstalk performance.

3.2 *Low-Level Logic*

After studying the characteristics of many of the widely used gate circuits, the diode-coupled AND-NOT gate[5] was chosen for use in No. 1 ESS. The circuit is generally referred to as low-level logic (LLL). As shown in Fig. 2, the LLL gate consists of a diode AND gate, a biasing circuit to overcome the combined voltage drop of an input diode and the driving transistor's saturated V_{CE}, and an inverting amplifier. The combination of AND plus INVERSION derives the logical AND-NOT function. This circuit has an individual source of base current to avoid the unequal base current problems present in other logic configurations. The diodes are poled in such a manner that large reverse base drive currents can be obtained. These properties result in full utilization of the switching speed capabilities of the transistor. The voltage margins are designable and do not depend solely on device characteristics. All in all, LLL is a high-performance gate yielding a

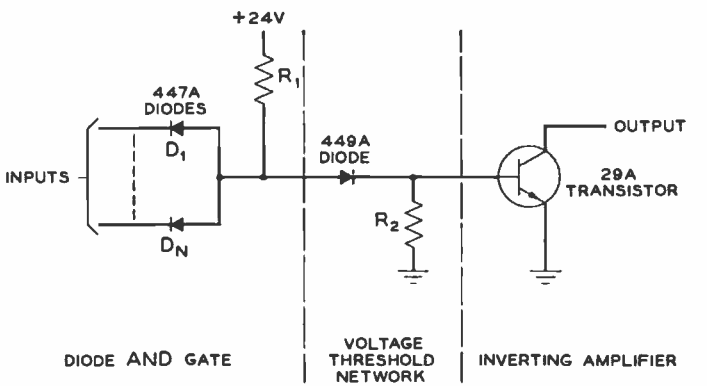


Fig. 2 — LLL circuit.

combination of fast switching speeds, large fan-in and fan-out, and excellent margins.

IV. LLL CIRCUIT DESIGN

4.1 *Worst-Case Design*

The worst-case design philosophy was used for the LLL circuit. This means that the expected characteristics of the gates were predicted on the basis of all circuit parameters being simultaneously at their extreme limits in such a manner as to degrade performance. This approach enhances circuit reliability by insuring that the margins of typical circuits are well within the design limits. For the LLL gate, worst-case design still yields very fast operation with moderately high fan-in and fan-out for relatively wide resistor and supply voltage tolerances.

4.2 *Saturated Operation*

An important consideration in the design of the LLL circuit is the use of a saturating or nonsaturating amplifier design. The primary advantage of the nonsaturating design is the elimination of the delay caused by the storage time of minority carriers in the transistor during turn-off. The significance of storage time is very much a function of the particular transistor employed and the need for speed in the system. Nonsaturating designs are complex, costly circuits which result in larger output voltages when the transistor is conducting. Also, biasing of the transistor in its active region not only increases power dissipation in the device, but also causes the circuit to be more sensitive to noise and prone to oscillation.

Part of the apparent speed advantage of the nonsaturating circuit is lost because of the necessarily higher threshold voltages. For many high-speed silicon switching transistors the storage times are so short relative to their rise and fall times as to be almost negligible in an LLL-type circuit. Furthermore, since approximately half of the stages in a logic chain are turning ON and half are turning OFF at any one time, only half the improvement in turn-off delay can be deducted from the average delay per stage. These considerations led to the selection of a saturating amplifier design.

4.3 *Biasing Network*

The major item for design in the LLL gate is the biasing network. Once a method for performing this function is selected and the element

is properly designed, the characteristics of the gate are largely determined by the inherent capabilities of the semiconductor devices, ambient operating conditions, and the tolerances of the resistors and power supply voltages.

The primary function of the biasing network in an LLL gate is to insure that its transistor will not conduct when any input to that LLL is held low by a conducting transistor in the prior stage. This requires that the biasing circuit keep the base-emitter voltage below the minimum threshold of the transistor for the maximum sum of the diode voltage drop and the V_{CE} of the saturated transistor.

The biasing network has three other important functions. First, the design should cause relatively large reverse base current flow when the transistor is switched from ON to OFF. This is necessary to obtain minimum turn-off time. Second, the biasing network should provide a noise threshold to the amplifier to prevent small transient voltage perturbations from falsely turning on the transistor. Third, the network should be designed so that the voltage across it remains as low as possible when current flows from R_1 into the base of the transistor (see Fig. 2). This minimizes the voltage swing required at the gate input.

The voltage biasing element in the LLL circuit consists of three silicon junctions in series. The three junctions provide approximately a two-volt threshold to the amplifier without the need for a second power supply. Resistor R_2 across the base-emitter junction of the transistor provides a relatively low-impedance shunt for the small forward current in the multijunction diode when the input to the circuit is low. This improves the noise margin by insuring a low potential at the base of an OFF transistor. The signal voltage swing is minimized since the difference between the threshold voltage and the forward-biased voltage drop of the multijunction diode is small. The reverse breakdown voltage of the three junctions is very high to prevent any reverse dc current. The necessary reverse transient drive is obtained by designing the multijunction diode to have a relatively long recovery time. Thus if an input to an LLL circuit whose transistor has been conducting suddenly goes low, the voltage across the diode cannot change instantaneously. For a brief time the base of the transistor is made negative and reverse base current is drawn through the temporary low reverse impedance of the multijunction diode. After recovery the diode reverts to a high impedance and prevents noise from being transmitted into the base node. To further insure this, the net capacitance of the three junctions in series is kept small.

4.4 *Dummy Load Resistor*

The LLL circuit shown in Fig. 2 has one serious deficiency. In the OFF state the collector node is at an indeterminate potential between two high impedances which are presented by the OFF collector and the nonconducting input diodes in the driven circuits. Not only does this make the node sensitive to noise, but it also slows down the turn-on of the following stages, since collector node stray capacitance must be charged by the internal gate resistors of the driven LLL circuits. The addition of a dummy load resistor from the collector node to a voltage above the threshold of the driven inputs provides a charging current for the capacitance, defines the OFF state voltage and provides a noise margin in this state, and limits the depth of saturation by establishing a lower bound for collector current.

4.5 *Operating Levels*

The choice of current and voltage levels for the LLL circuit is influenced by many of the parameters discussed earlier and ideally should be made for each system and device selection. In general, the selection of low impedance levels improves speed and capacitive crosstalk performance while higher levels result in lower power dissipation (allowing higher packing densities), improved device reliability, and fewer inductive interaction problems. In high-speed circuits the mechanical embodiment of individual circuits and of whole equipment units is significant because of its primary effect upon stray capacitance and inductance.

The impedance level chosen for the logic circuits in No. 1 ESS was largely determined by the stray capacitance anticipated at collector nodes and by estimates of wiring inductance. To charge an estimated maximum node capacity of 150 pf to the threshold of an LLL circuit in 50 nsec implies a charging current of about 10 ma. In the worst case (fan-out of one), approximately 75 per cent of this current is assigned to the dummy load and 25 per cent to the internal gate of the driven circuit. This guarantees the 10 ma for the minimum fan-out case and keeps the ratio of maximum to minimum collector current low. The choice of a collector supply voltage in the 4- to 5-volt range provides sufficient margin for noise and establishes the OFF collector impedance at about 500 ohms. The current and voltage levels chosen assure fast switching speeds for the low-power semiconductor devices described in Section V. In addition to these considerations, the effects of stray capacitance and inductance on interaction between circuits (crosstalk) had to be evaluated for the packaging and wiring arrangements used in No. 1 ESS.

4.6 *L and C Crosstalk Effects*

In addition to degrading rise times, stray capacitance between signal leads can cause interference between circuits. At the impedance level chosen, these effects do not significantly alter performance in this system.

Wiring inductance can affect circuit performance in two ways. Series lead inductance causes longitudinal voltage drops due to the rapid change of signal current. The allowable length of wire depends upon its inductance, the noise thresholds of the circuits, the operating current levels, and the required switching speeds. The wire inductance can be reduced by using twisted-pair wire for long interconnections. In addition to the series inductance problems, the mutual inductance occurring between parallel wire runs can cause interference between circuits.

The interactions of the L and C effects, together with the complex wiring patterns and nonlinear waveforms occurring in logic circuits, make an exact analysis of the net effects very difficult.[6] Analytic studies of simplified models in conjunction with laboratory measurements were utilized to formulate the following set of wiring rules to insure that the physical wiring patterns would limit the crosstalk to acceptable levels.

4.7 *Wiring Rules for Interconnecting Logic Circuits*

(1) Leads must be as short as possible, using horizontal and vertical paths

(2) wires following the same path should be loosely constrained, i.e., not laced into a cable

(3) a single wire should be used to tie the points of a node together and to a dummy load resistor where:

(a) the total node wiring is less than 6 feet and

(b) parallel runs do not exceed 3 feet

(4) twisted pairs must connect points of a node together and to three dummy loads in parallel at the driven end where:

(a) the total node wiring exceeds 6 feet and

(b) parallel runs exceed 3 feet but in no case exceed 10 feet

(5) when two LLL circuits are interconnected to form a flip-flop, they must be on the same circuit pack.

V. DEVICE CHARACTERISTICS

5.1 *General*

The characteristics of each of the semiconductors used in the LLL circuit are shown in Tables I–III. Both the input diode and the multi-

junction diode are specifically characterized for the LLL circuit. The transistor, on the other hand, is characterized for use in many different applications in a variety of circuits in No. 1 ESS.

The reliability of these devices when used in the LLL gate has been predicted on the basis of step-stress aging experiments.[7] The studies indicate that failure rates will be less than 10 in 10^9 device-hours for the transistor and less than 5 in 10^9 device-hours for the diodes. These rates are adequate to meet the reliability objectives discussed in Section 2.3.

The variations of specific parameters over the temperature range of 0–55°C were considered in establishing specific limits, although all limits are specified at 25°C to facilitate testing.

5.2 *29A Transistor*

The most important parameters of the transistor employed in the LLL circuit are presented in Table I.

The choice of operating level in the LLL circuit restricts the maximum collector current, collector voltage, and power dissipation to well below the maximums allowable. In general, these margins have been utilized to improve reliability in the logic circuits. In a number of cases the extra capability has been utilized in the design of scaled-up versions of the LLL circuit which satisfy the need to drive high fan-out situations within the logic and to drive cables to remote units.

Although the specified limits are important in determining worst-case performance, a knowledge of the distribution of parameters is necessary to predict typical behavior. Distributions of some of the more significant parameters are given in Figs. 3, 4, and 5. From the figures it is obvious that all of the limits specified are well within the device capability, which assures a high yield in production with resultant economies.

TABLE I — 29A TRANSISTOR SPECIFICATIONS

$BV_{CEO} \geq$ 30 v	I_C = 5 ma, I_B = 0
$BV_{EBO} \geq$ 7 v	I_C = 0, I_E = 10 μa
$h_{FE} \geq$ 30	I_C = 5 ma, V_{CE} = 1 v
$V_{CE} \leq$ 0.5 v	I_C = 50 ma, I_B = 2 ma
$V_{BE} \leq$ 1.1 v	I_C = 50 ma, I_B = 2 ma
$C_{ob} \leq$ 6 pf	I_E = 0, V_{CB} = 5 v
$C_{ib} \leq$ 10 pf	I_C = 0, V_{EB} = 1.5 v
$T_{on} \leq$ 65 nsec	I_C = 35 ma, I_B = 1.5 ma
$T_{off} \leq$ 65 nsec	I_C = 10 ma, I_B = 2.5 ma, V_{BER} = −0.9 v
$\theta_{J\text{-}C} \leq$ 0.33°C/mw	

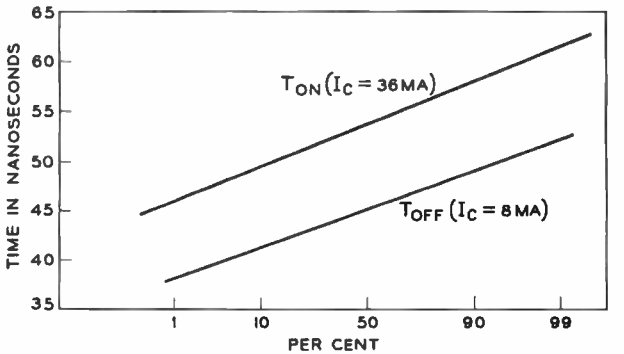


Fig. 3 — Typical switching speed distributions for worst-case conditions for 29A transistor.

5.3 *449A Multijunction Diode*

Table II lists the parameters of the multijunction diode used in the biasing network of the LLL circuit. The minimum forward voltage specification guarantees an adequate noise margin under the worst (high-temperature) conditions for the OFF state. The maximum limit assures a low drop in the conducting state, which provides an efficient transfer of current into the base of the transistor when it is turned ON.

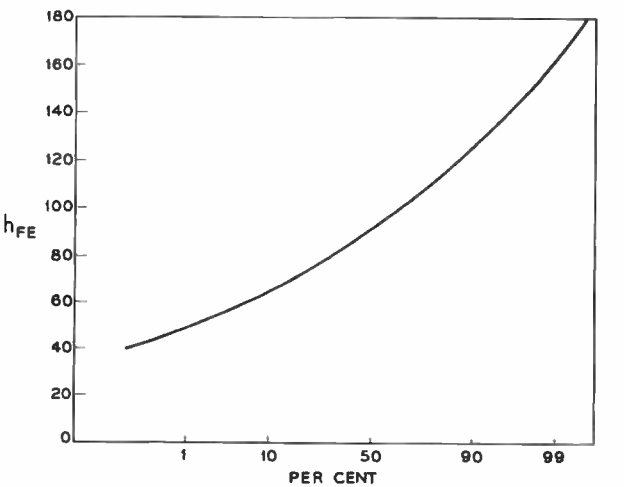


Fig. 4 — Typical h_{FE} distribution for 29A transistor: I_C = 5 ma, V_{CE} = 1 v, circuit design limit $h_{FE} \geqq 22$.

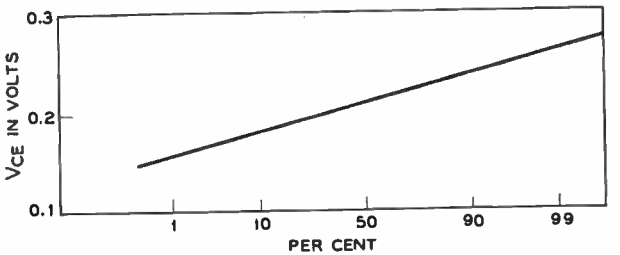


Fig. 5 — Typical V_{CE} distribution for 29A transistor: I_C = 50 ma, I_B = 2 ma specification limit ≦ 0.5 v.

Specifying the minimum stored charge guarantees that sufficient base current can be drawn during turn-off of the slowest 29A transistor. The upper limit on the total capacity across the three junctions limits noise transmission to a satisfactory level.

5.4 *447A Logic Diode*

The parameters of the 447A diode are given in Table III. The forward voltage drop, typical for a small silicon computer diode, is easily accommodated by the threshold created by the multijunction diode and the transistor base-emitter junction. The reverse breakdown is higher than necessary for the circuit, but improves reliability and allows for other applications. The reverse leakage current is so small as to be negligible for any reasonable fan-in.

The upper limits on capacitance and reverse recovery time are necessary to assure obtaining the full speed capability of the transistor. Consider the circuit in Fig. 6. If both Q_0 and Q_1 are conducting, current through R_1 will be shunted through Q_0 to ground and the current in R_2 passes through Q_0 and Q_1 in parallel; therefore Q_2 and Q_3 are nonconducting. If Q_0 is suddenly turned OFF, the shunt path to ground for I_1 will be removed, allowing Q_2 to turn ON. On a transient basis, however, diode D_2, which serves to isolate the collector nodes, can have a

TABLE II — 449A MULTIJUNCTION DIODE SPECIFICATIONS

$V_F \geqq$ 1.53 v	I_F = 70 μa
$\leqq$ 2.3 v	I_F = 2.5 ma
$\leqq$ 2.85 v	I_F = 20 ma
C 15 pf	V_F = 1 v
Stored charge $\geqq$ 0.4 × 10^{-9} coul	I_F = 2 ma, I_R = 10 ma

TABLE III — 447A LOGIC DIODE SPECIFICATIONS

$BV \geq$ 40 v	I_R = 5 μa
$I_R \leq$ 25 na	V_R = 20 v
$V_f \leq$ 1 v	I_f = 10 ma
$C \leq$ 3.5 pf	V_R = 0, f_0 = 100 kc
$t_{rr} \leq$ 4.0 nsec	

low reverse impedance for a time equal to its recovery time. As long as the low reverse impedance condition persists, the current from R_1 can "leak" through D_2 as indicated by I_3 in the figure. This interaction between nodes causes no logical error but does introduce an absolute delay in Q_2's turning on. In a fast system such delay is not tolerable. The capacitance of the diode allows a similar interaction. In addition, the capacitance can transmit noise into the internal AND gate node and also degrade rise times at this node by its shunt effect on the current from R_1. The specifications on the 447A diode are such as to make these effects almost negligible.

VI. SUMMARY OF LLL CHARACTERISTICS

The outstanding characteristics of the LLL circuit designed for No. 1 ESS are summarized in Table IV.

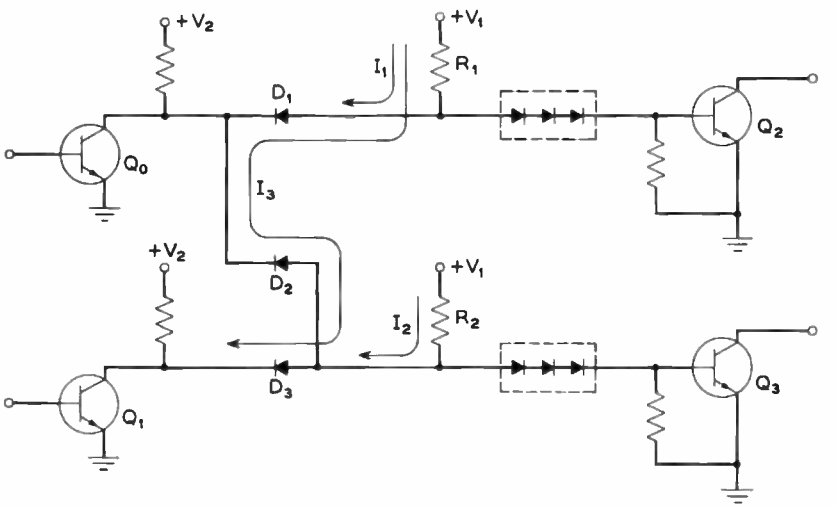


Fig. 6 — Diode recovery effect.

TABLE IV — LLL CHARACTERISTICS

Parameter	Range	Typical Value
Signal levels		
high (1)	4.0–5.1 v	4.6 v
low (0)	0.1–0.5 v	0.25 v
Fan-out	1–8	4
Fan-in	1–20	4
Turn-on delay	10–65 nsec	35 nsec
Turn-off delay	10–65 nsec	35 nsec
Noise margins		
high	1.6–3.6 v	2.5 v
low	1.0–1.5 v	1.2 v
Circuit margins		
resistor tolerance	±20%	
voltage tolerance	±10%	
Power consumption	37–140 mw	76 mw
Operating temp. range	0–55°C	

VII. LOGIC DESIGN

7.1 *Applications of LLL in Logic Designs*

It was mentioned earlier that LLL gates can be used exclusively to synthesize any logical combinaton possible with AND, OR, and NEGATION. The gate performs the AND-NOT function for the convention that a "high" or +v signal is a logical 1 and a "low" or ground signal is a logical 0.

The standard logic symbol for the LLL circuit is shown in Fig. 7 along with a function table defining its properties.

There are many ways in which complex logic circuits can be synthesized using LLL. One of the simplest approaches is to perform the original logic design in terms of the more familiar AND, OR, and NOT functions and then to substitute LLL equivalents for each of the original functions.

(a)

A	B	C (OUTPUT)
0	0	1
0	1	1
1	0	1
1	1	0

(b)

Fig. 7 — Building block logic.

The basic equivalent circuits and several other important configurations are shown in Fig. 8.

In using the substitution procedure, the following two rules should be repeatedly applied until further simplification is impossible:

(1) cancel (remove) tandem pairs of single input LLL gates since two inverters in series perform no net logical function;

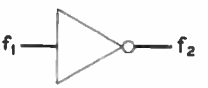
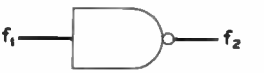
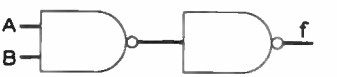
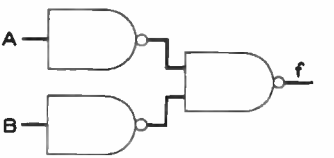
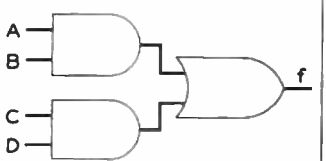
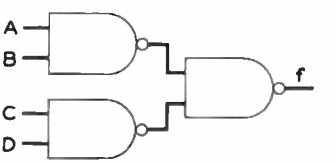
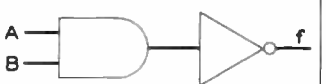
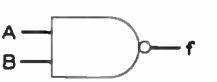
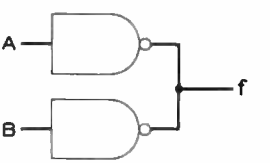
FUNCTION	AND, OR, NOT CIRCUIT	LLL CIRCUIT
INVERSION: $f_2 = f_1'$		
AND: $f = AB$		
OR: $f = A + B$		
AND – OR: $f = AB + CD$		
AND – NOT: $f = (AB)'$		
OR – NOT: $f = (A + B)'$		

Fig. 8 — Equivalent logic circuits.

(2) when two or more LLL gates drive a common gate and this gate is their only load, the outputs of all of the driving gates can be tied together and applied to a single input of the driven gate.

The example in Fig. 9 illustrates this process of simplification.

Several of the examples highlight the common connection which is

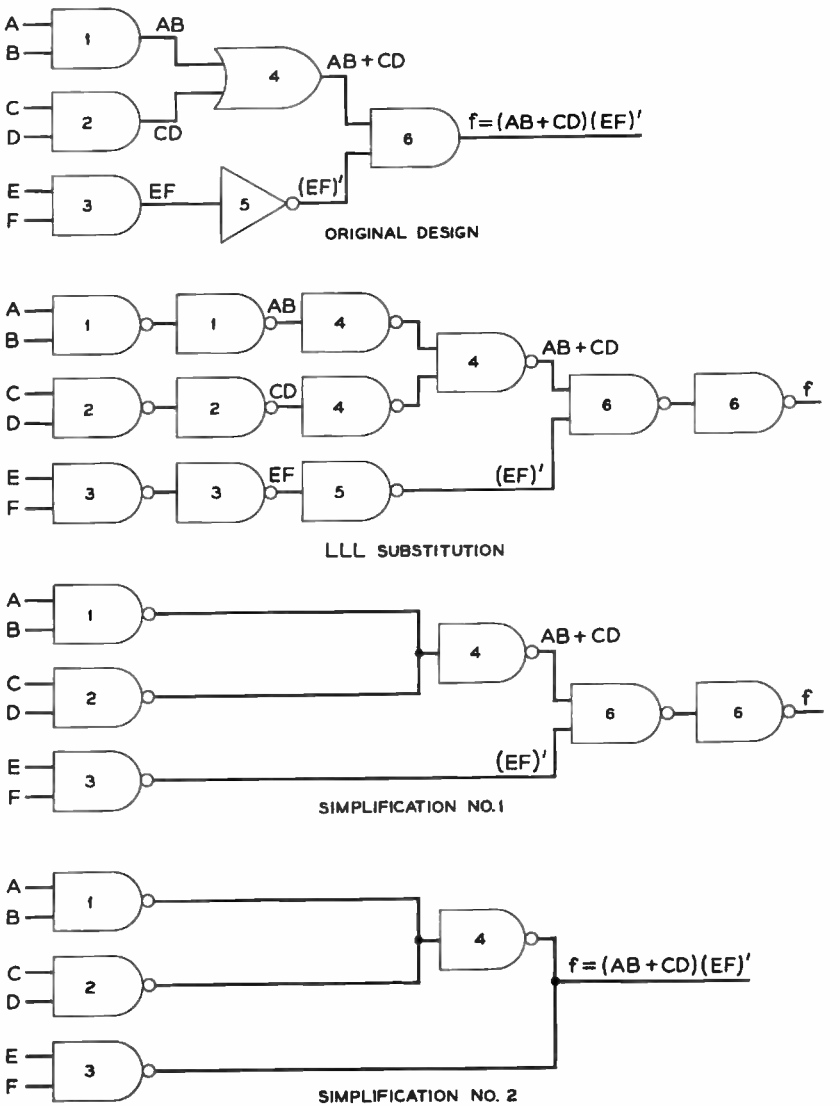


Fig. 9 — LLL simplification process.

made between the outputs of transistors which are driving a common input. An AND function is effectively formed here, since all transistors connected to the junction must be nonconducting in order that the output be high (logical 1). The use of this technique has resulted in the savings of several thousand logic diodes in the central control alone. The number of transistors which can be connected in this way is limited by the adverse effect on switching speed which is caused by the increases in inductance and capacitance.

Another approach in using LLL in logical designs is the "inhibit" technique. A simple example illustrates this approach in the design of a match circuit. In such a design, the output should be high when the two input variables (A and B) have the same state. Fig. 10(a) illustrates this solution. Conversely, one can think of the match function as requiring a high output except when A and B mismatch, in which case the output should be inhibited. This leads to a solution utilizing one LLL for each input combination which is to inhibit the output [see Fig. 10(b)]

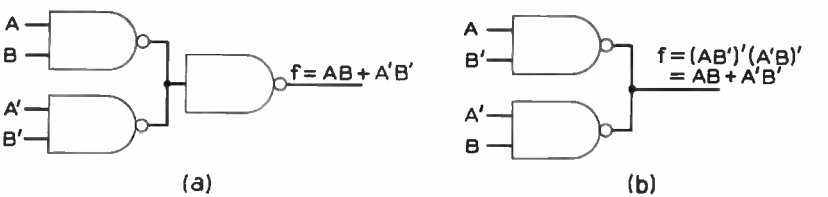


Fig. 10 — LLL match circuits.

7.2 *The LLL Flip-Flop*

Throughout the switching system, and particularly within the CC, there is the need for temporary memory facilities in the form of flip-flops. These are readily constructed by cross-connecting two LLL gates as shown in Fig. 11. The circuit operates as follows: gating functions insure that inputs A and B are both held in the high state when the

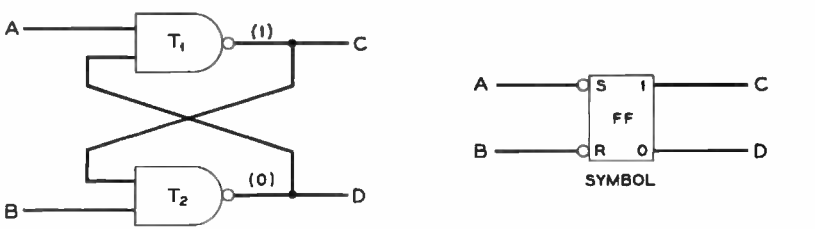


Fig. 11 — LLL flip-flop.

flip-flop is quiescent. Assume that the "set" output is high ($C = 1$) and accordingly $D = 0$. Any change in the state of A is ineffective, since the cross-connected input from D keeps T_1 nonconducting. However, if B goes to 0, the base drive for T_2 is removed, and D will go high, providing drive for T_1 and causing C to change from 0 to 1.

7.3 *Binary Counters, Shift Registers and Cable Pulsers*

In addition to the basic LLL gate and flip-flop, special circuits for binary counters and shift registers are also required. These consist of steering gates appliqued onto the LLL flip-flop to provide nonracing counting and shifting functions. "Nonracing" means that correct operation is independent of input pulse width as long as the width is greater than the minimum allowed, which here is approximately 0.25 microsecond.

The binary counter circuit is shown in Fig. 12. As can be seen, memory is provided by an LLL flip-flop, which has been modified to include direct connections to the transistor bases. Two steering networks and a

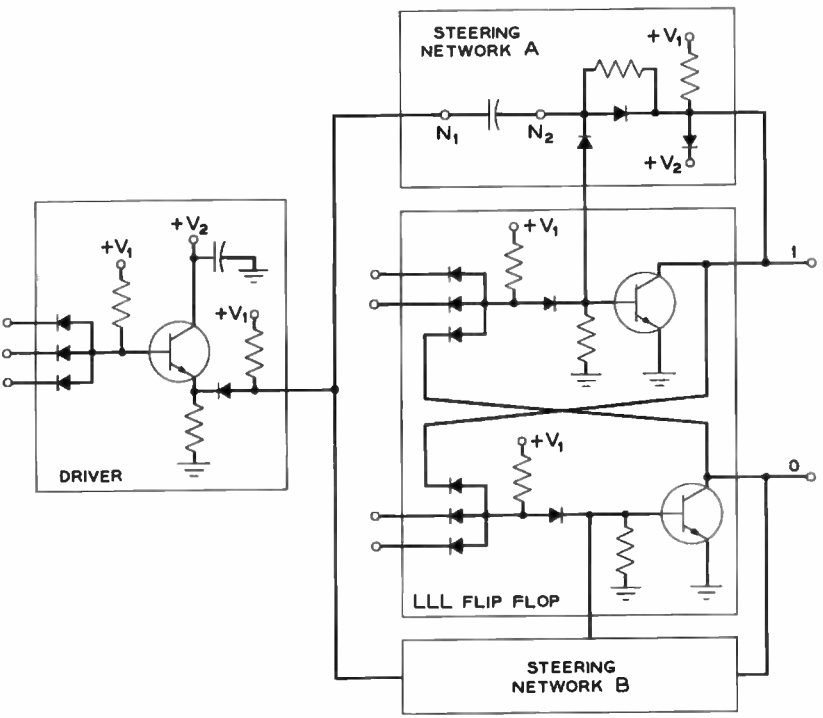


Fig. 12 — Binary counter.

special driving circuit have been added to obtain the nonracing binary counting action.

The design is such that if all inputs to the driver become high coincidently for longer than 0.25 microsecond, the flip-flop will reverse its state when any input returns to the low condition and remains there for at least 0.2 microsecond. In the steady-state condition, the N_2 points in the networks are at approximately the same potential as the output voltage of that side of the flip-flop to which they are connected (either +4.5 or approximately zero volt), while the other sides of the capacitors (points N_1) are near ground potential. When all driver inputs go high, the emitter follower drives the N_1 points to near +4.5v and the N_2 points remain at their original voltage levels. When the output of the driver goes low due to an input going low, the N_2 points attempt to drop by 4.5 volts. That N_2 which was originally at +4.5v will drop to near ground potential, while the other N_2 will go negative and forward bias the diode connected to the base of the transistor on its side of the flip-flop, turning it OFF. Since this transistor was originally ON, the flip-flop will change state. The N_2 points in both networks will return to the new output voltage states as the capacitors in the networks charge to their new voltage values. The next pulse from the driver will repeat the process, except that the opposite transistor will now be the one which turns OFF.

The shift register cell is almost identical to the binary counter, except that the two steering networks are driven by individual drivers which have one common shift control input. The two drivers have individual data inputs which are connected to the complementary outputs of the preceding stage in a multistage shift register. In this way only one driver output will be activated when the shift control input goes high and the cell will be forced into the original state of the preceding stage when the shift signal goes low.

The LLL gate has one other important application in No. 1 ESS. It is used (at a higher current level) as the basic cable pulser for the interframe bus communication system (see Fig. 13). The bus receiver circuits which are designed to drive standard LLL gates are also shown. This application is described in detail in a companion paper.[2]

VIII. LOGIC DESIGN IN THE CENTRAL CONTROL

8.1 *General*

By far the largest use of logic circuits in No. 1 ESS is made in the central control (see Fig. 14). Some feeling for the size and complexity

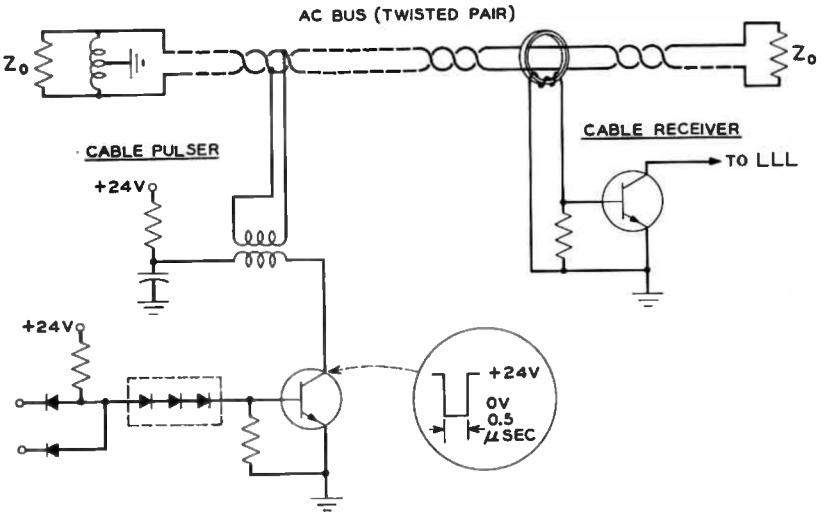


Fig. 13 — LLL gate used as a cable pulser.

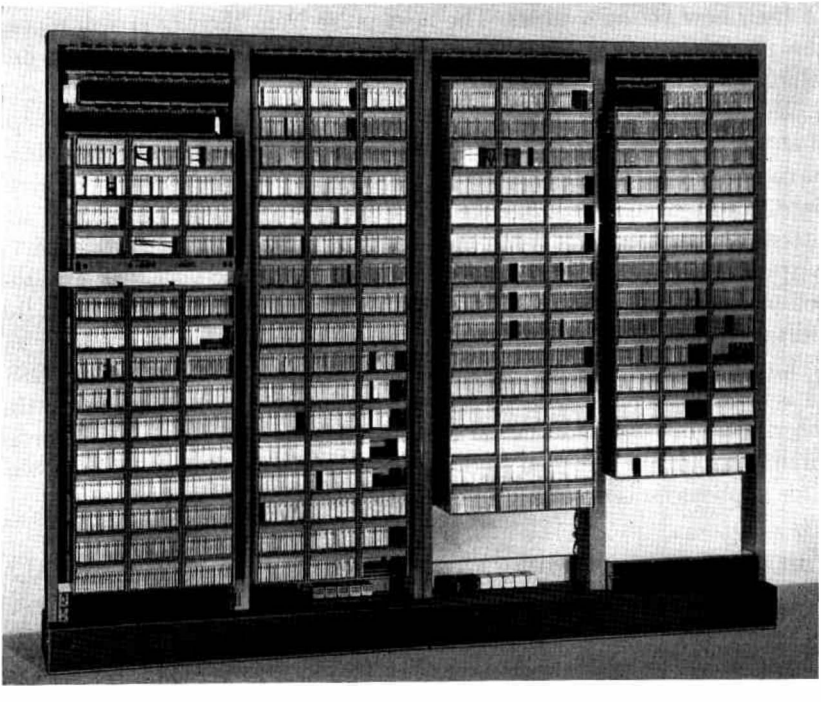


Fig. 14 — Front view of central control.

of the unit is found in the fact that the circuitry for a pair of duplicated CC's, while contained within about 5 per cent of the volume of a typical 10,000-line telephone office, contains about 50 per cent of the semiconductor devices. The statistics of Table V show the CC in perspective with the rest of the system.

Because of its size, it is impractical to discuss all of the details of the basic data processing circuits of the CC within the scope of this paper. The majority of circuits are used in translating program instructions, providing internal flip-flop registers between which data routinely shuttle, communicating with peripheral equipment, and performing routine maintenance checks. These circuits contain a high degree of design repetition and their implementation is fairly straightforward. Most of the real processing of data takes place in a small portion of the machine known as the "arithmetic system." The functions of this unit will be developed in detail, as an illustration of the over-all logic design and equipment layout techniques that have been employed.

8.2 *Arithmetic System*

In the process of completing telephone calls and diagnosing system troubles, the central control is called upon to manipulate both arithmetic quantities and logical quantities which are characterized in binary form. Arithmetic numbers contain 23 bits, labelled 0 to 22 in ascending order from the least significant at the right to the most significant at the left. Positive and negative numbers are permissible and are distinguished by the state of bit 22, the sign bit. Since the central control has a capacity of 23 bits, including the sign, the largest number therefore cannot exceed $2^{22} - 1$ ($= \pm 4{,}194{,}303$). For positive numbers, the sign bit is 0. Negative numbers are the "ones" complement of the equivalent positive value and contain a 1 in the sign bit. For example, $+1$ is $000 \cdots 001$, and -1 is $111 \cdots 110$. Both $+0$ ($000 \cdots 000$) and -0 ($111 \cdots 111$) can exist in this number system.

One of the chief constituents of the arithmetic system is a parallel

TABLE V — RELATIVE COMPLEXITY OF CENTRAL CONTROL

Item	No. of Items in Two CC's	% of Total	No. of Items in 10,000-Line No. 1 ESS
Circuit packs	4,716	37.0	12,600
Transistors	27,142	49.0	55,000
Diodes	89,934	54.0	160,000
Wire-wrapped connections	173,200	13.0	1,300,000
Equipment bays	8	5.0	147

binary adder circuit which generates the algebraic sum or difference of two or more 22-bit signed numbers in an interval of less than 1 microsecond. Addition is performed conventionally by summing corresponding bits of a number and combining them with carries that might have been generated in less significant bit positions. Subtraction is performed by the use of addition of complements with end-around carry. That is, the subtrahend is first complemented and then added to the minuend. A carry from the most significant bit sum is returned to be re-added to the lowest-order bits.

Another function of the arithmetic system is the ability to perform the purely logical functions of AND, OR, and EXCLUSIVE-OR upon two 23-bit quantities.

The arithmetic unit also has provision for shifting or rotating the positions of the bits of a binary word within a register. A shift register circuit is available which can move a word in either the right or left direction by any integral displacement of from 0 to 23 positions. A special mode rotates only 16 bits of the word, while leaving the others undisturbed. A complete shift cycle requires approximately 3.5 μsec.

8.2.1 *Block Diagram*

A simplified block diagram of the arithmetic system is shown in Fig. 15. The elements divide into two categories, namely, those concerned with the additive-type functions and those relating to rotational functions.

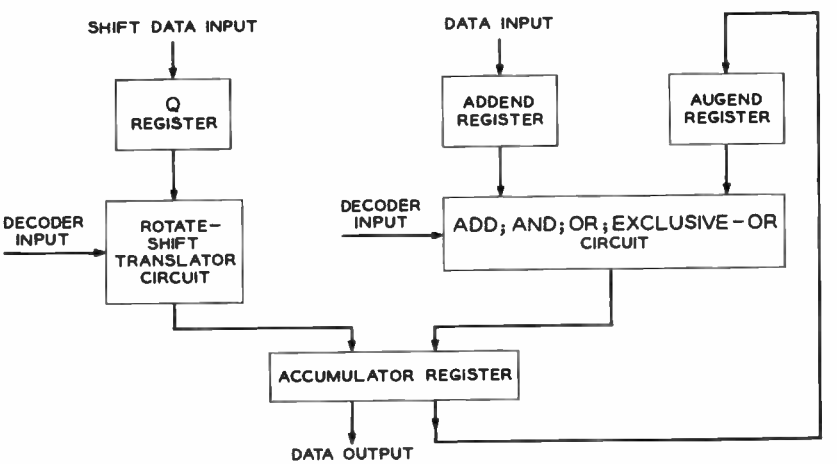


Fig. 15 — Block diagram of arithmetic system.

The circuit requirements for the additive functions are quite simple. Two flip-flop registers, the addend and augend, act as transient memories into which are inserted the incoming operands. The block labelled ADD, AND, OR, EXCLUSIVE-OR contains the logic to produce all of these functions. Finally, a register is required to hold the resultant. This is the accumulator.

The accumulator is constructed of LLL shift register circuit packs (Section 7.3) to provide greater flexibility than mere flip-flop memory. Steering networks associated with each side of the internal flip-flop allow tandem, race-free connections to other bits in the register. Upon command, the value of any bit in the register can be transferred to any other bit position to which it has been wired. The logic to interpret the shift-rotate commands and to translate the amount of displacement is contained in the rotate-shift translator. The Q register is a 6-bit flip-flop group which holds the shift-rotate translator data while the operation is taking place.

8.2.2 *Accumulator Input Registers*

The addend and augend registers consist of 23 LLL flip-flops with associated input gating. All new data are introduced into the arithmetic system by way of the addend register, which connects to the central control's main transmission artery — the masked bus. The augend register has two input options. It may be "reset" to zero by a signal from the order decoders, or it may be set up to the contents of the accumulator register. The latter path provides the ability to accumulate a sum by the process of repeatedly adding new numbers to the earlier totals. Since both input registers are connected to the adder circuit without gating, the adder continually monitors the sum of the register contents.

8.2.3 *Adder Circuit*

The inherent slowness of binary adders arises from the fact that "carries" which are generated in low-order bits create an appreciable delay by propagating serially toward the more significant bit positions. A complete sum cannot be available until all carries have been terminated. An adder circuit was developed which partially overcomes this problem and achieves the required speed without an unreasonable increase in circuit components. The circuit makes use of the group-carry concept, whereby all the carries within a prescribed group of bits are generated simultaneously and propagation is restricted to occur only between the adjacent groups.

When adding the corresponding bits of two binary numbers X and Y, a carry is obtained whenever both bits x_i and y_i are 1; or when either x_i or y_i is 1 in conjunction with a "carry in" from the adjacent lower-order bit. The relation can be expressed as:

$$C \text{ out} = X_i Y_i + (X_i + Y_i)C_{i-1} .$$

In order to take advantage of the inversion property of the LLL circuit, it is convenient to complement the expression to obtain

$$C' \text{ out} = X_i'Y_i' + (X_i' + Y_i')C_{i-1}'.$$

Now, the "carry out" from the ith position will be the "carry in" to $i + 1$, so that C_{i+1}' can be expressed as

$$C_{i+1}' = X_{i+1}'Y_{i+1}' + (X_{i+1}' + Y_{i+1}')C_o'.$$

Substituting the value of C_o and simplifying yield the product-of-sums expression:

$$C_{i+1} = (X_{i+1} + Y_{i+1})(X_i + Y_i + X_{i+1}Y_{i+1})(C_{i-1} + X_iY_i + X_{i+1}Y_{i+1}).$$

Let

$$A_i = X_i + Y_i$$

$$B_i = X_iY_i .$$

Then

$$C_{i+1} = A_{i+1}\,(A_i + B_{i+1})(C_{i-1} + B_i + B_{i+1}).$$

If the process is continued for successive carries, a general expression for the nth carry is obtained in terms of the input:

$$C_n = A_n\,(A_{n-1} + B_n)(A_{n-2} + B_{n-1} + B_n)(\cdots)\cdots$$

$$\cdots (A_2 + B_3 + \cdots + B_{n-1} + B_n)(C_{in}\,A_1 + B_1 + B_2 + \cdots + B_n).$$

The equation consists of n product-of-sum terms whose complexity increases in a pyramidal pattern. For example, the term $(A_1 + B_2)$ would appear as the second factor in generating the carry C_2. It re-appears, in union with B_3, as the third factor of C_3. This new factor, $(A_1 + B_2 + B_3)$, combines with B_4 as the fourth product of C_4, and so on. Each product term in every carry expression follows a similar growth pattern. These combinations are suited to LLL circuitry in that the highly repetitive B terms need only be generated once and can be combined successively (within fan-out limitations) to form a series of carry outputs.

Although any number of carries can be generated simultaneously in a two-stage logic circuit by implementing the above equations, the number of LLL gates required quickly becomes excessive when more than about 15 carries are generated. The curve of Fig. 16 summarizes the circuit requirements for multibit adders of any size up to 25 bits. It is useful in predicting the circuit complexity for tandem arrangements of carry groups to obtain a desired addition speed. The curve is divided into two areas. The portion with the larger delay occurs for group sizes greater than 11 carries and is occasioned by the fact that an extra logic stage is necessary to satisfy the fan-out limitations of the LLL circuit. The minimum delay of four logic stages includes the two stages for addition in the sum circuit, and applies to the first group of a tandem arrangement only, since the carries in adjacent groups are completed concurrently with the preceding sum. Therefore, each subsequent carry group after the first contributes only two delays to the total addition time. On this basis, an adder with two group carries requires six delays, one with three groups requires eight delays, etc.

Fig. 17 compares the circuit component requirements with the delay for several possible 23-bit end-around carry adders. Point A on the curve indicates that approximately 800 LLL gates are needed to design a circuit which generates all bits simultaneously and produces a sum in 5 stages of delay, ≈300 nsec. At the other extreme, point D describes the conditions of a quasiparallel adder where the carry propagates on a bit-by-bit basis. B shows the condition for which the carries are divided

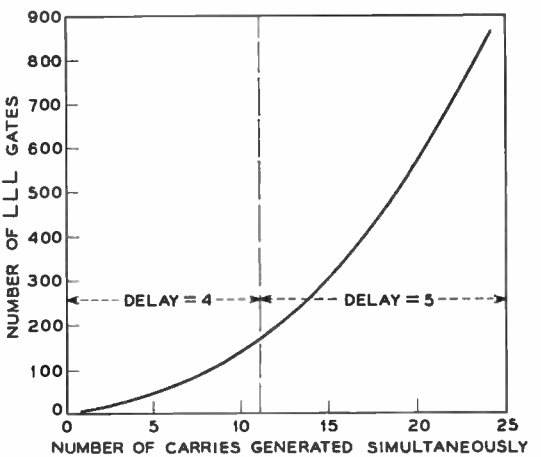


Fig. 16 — LLL circuit requirements for parallel carry adders.

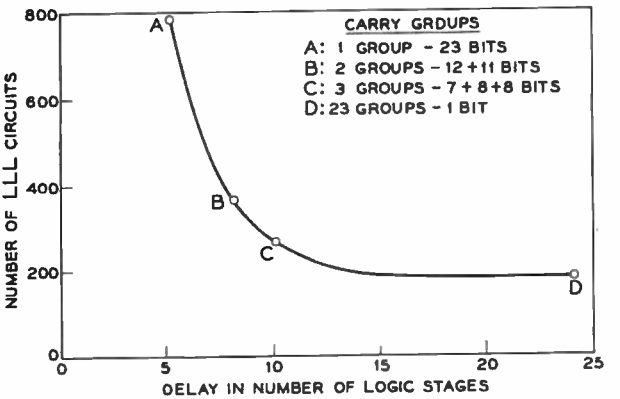


Fig. 17 — Circuit requirements and delay of 23-bit adders.

into groups of 11 and 12 bits, and indicates better than 2:1 savings in equipment over the circuit in A at the cost of ≈200 nsec. Point C defines the central control adder. It requires 270 LLLs and has a delay of 650 nsec, determined on the basis of 65-nsec worst-case transistor switching speed.

The block diagram of Fig. 18 shows the adder circuit used in the central control. For maximum economy in satisfying the speed requirements, the carry circuit is divided into three groups: one produces

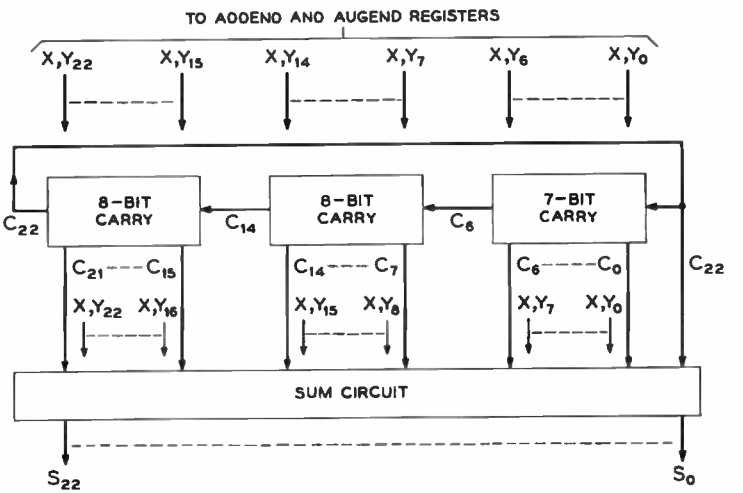


Fig. 18 — Block diagram of adder.

simultaneous carry outputs from seven bits, the other two groups from eight bits apiece. A single carry propagates from one group to the next in tandem, with the third group rejoined to the first in order to provide the end-around-carry connection. With each group after the first contributing the delay of two logic stages, and an added two stages of delay caused by the end-around-carry provision, the maximum propagation time is that of ten stages or, at worst, 650 nsec. Maximum propagation delay occurs when a carry terminates in the same group from which it had originated, after having propagated through the other groups. The addition of $+1$, $(00 \cdots 01)$, to -0, $(11 \cdots 11)$, is an example where the initial carry in the least significant bit causes a carry to ripple through all succeeding bits, until the end-around connection returns it to the origin where it terminates. The correct result is, of course, $00 \cdots 01$, or $+1$.

The end-around-carry connection between carry groups constitutes a feedback loop which can give rise to oscillation when two binary numbers, one of which is the exact complement of the other, are to be added. The problem exists because the carry circuit reflects the momentary conditions of its input registers at all times. The closed loop constitutes a delay line which can inject a remembered carry indication at the exact moment when a register is changing state. Normally, the circuit stabilizes when the register state has settled down due to the further generation of new carry terms, but when the numbers are complementary, no carries are initiated and the remembered carry could continually circulate or at least introduce an error. This problem is eliminated by inhibiting all carries (opening the feedback loop), during the transition period of either of the input registers.

The sum circuit combines the carries with the operand bits in the two-stage logic circuit shown in Fig. 19. The first stage of the circuit combines the inputs in four LLL gates as a sum-of-products which when inverted yields:

$$\text{SUM}' = X_i Y_i' C_{i-1} + X_i' Y_i C_{i-1} + X_i Y_i C_{i-1}' + X_i' Y_i' C_{i-1}'.$$

The output of this stage remains at the common-collector potential ($+4.5$ v) when at least one input to every product term is at ground potential. Logically, the equation states that the sum is 0 whenever any two of the operand bits or the input carry are both equal to 1, or when all three are zero coincidently.

An inverter amplifier stage is used to make the complementary sum output available to the connecting logic.

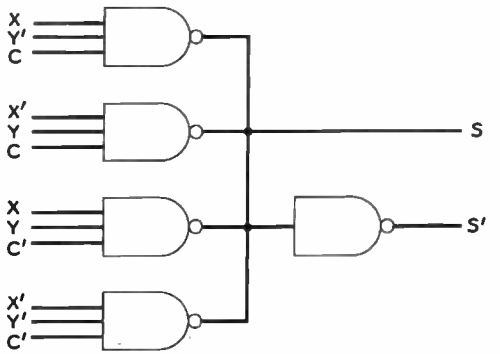


Fig. 19 — Logic diagram of sum circuit.

8.2.4 *Special Logic Functions*

The sum circuit logic contains each of the four possible combinations of the input variables X and Y in product with a carry term or its complement. By forcing the carry terms to become high ($C = C' = 1$), the equation for the sum degenerates to

$$\text{OUTPUT}' = X_iY_i' + X_i'Y_i + X_iY_i + X_i'Y_i'.$$

When appropriate control functions are introduced into each product term, the result can be expressed as

$$F' = f_1X_iY_i' + f_2X_i'Y_i + f_3X_iY_i + f_4X_i'Y_i'.$$

This is the familiar expansion of two variables; it can generate any of the sixteen functions of X and Y, depending on the values assigned to the control functions.

Control inputs were added to the LLLs in the sum circuit to generate the logical combinations required in the CC. The condition that $C = C' = 1$ was satisfied by grounding inputs to the appropriate logic packages in the carry circuit. Fig. 20 shows the sum circuit with the control inputs to generate logical AND, OR, and EXCLUSIVE-OR. None of the functions here required the condition of $f_4 = 0$; therefore it was not provided. To obtain a high output when performing the AND operation, we let $f_3 = 0$. Then

$$F'_{XY} = f_1XY' + f_2X'Y + X'Y'.$$

When $f_1 = f_2 = f_3 = 0$:

$$F'_{(X+Y)} = X'Y'$$

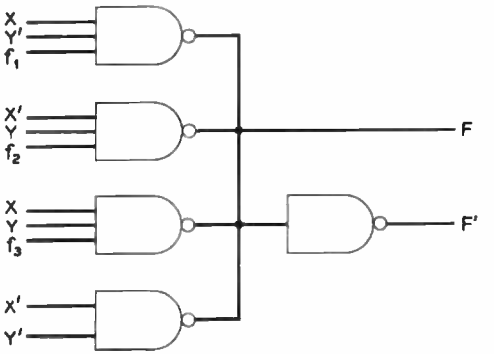


Fig. 20 — Modification of sum circuit to generate special logic functions.

which is high only when the bits satisfy OR logic. Finally, to obtain EXCLUSIVE-OR, let

$$f_1 = f_2 = 0:$$

$$F'_{(x \oplus y)} = f_3 XY + X'Y'$$

which is accordingly high for $XY' + X'Y$ as required.

Since the carry circuit is inoperative when forming the logical combinations, the delay time is merely that of the two-stage sum circuit or 130 nsec maximum.

8.2.5 *Shift-Rotate Functions*

When manipulating data in the central control, it frequently happens that particular bits of information are not located in the bit positions that are most convenient to use. This commonly comes about, for example, when several small data words are packed together into adjacent bits of a larger memory word. Product masking is effective in isolating the desired data word from the rest of the group, but does not affect the relative bit positions. To provide the ability to relocate all bits of a word in some specified manner, shift register features have been designed into the accumulator.

The two basic types of operation in the accumulator register are rotation and shifting. Full rotation indicates a displacement of the 23 bits of a word by any instructed amount from 0 to 23 positions. Those bits which pass through one end of the shift register are reinserted into the other end so that no information is erased in the process. A modified,

form of rotation is confined to be effective only within a selected group of sixteen bits. In both cases, the rotation can be specified to be in either a left or right direction. Shifting is also a displacement of all 23 bits in either direction, but here the bits are forced out of the end of the register and the vacated spaces at the opposite end are replaced with zeros.

The accumulator is activated by synchronous commands from the rotate-shift translator, which receives the basic instruction from the order word decoder and a 6-bit descriptive word from the Q register. The five least significant bits of this word are the binary equivalent of the amount of displacement; the sixth, the sign bit, determines the direction of motion. When the sign is 0 (plus), motion is to the left; when it is 1 (minus), motion is to the right. In the latter case, the accompanying five bits are in complement notation and signify a negative number. To obtain the true displacement of a negative quantity, the rotate-shift translator must recomplement these five bits.

The program instruction itself can also specify that the shift or rotation take place in complemented form. That is, under the influence of a "shift complemented" instruction, not only will the magnitude of the displacement be complemented, as in the case for negative numbers, but the direction of the motion will also be reversed. Table VI summarizes the various forms of motion which are obtained.

8.2.6 *Derivation of Shift Mechanism*

Only 4.0 microseconds may be allotted to the shifting operation, so that the output of the accumulator can be available for some other operation in the cycle following the shift instruction. This interval includes the time absorbed in receiving and translating the instruction and the description of motion, as well as that allowed for the mechanics of shifting and register settling. Conventional series shift registers, in

TABLE VI — TRANSLATION OF SHIFT-ROTATE INSTRUCTIONS

Instruction in Order Word Decoder	Q Register Shift Code		Bit Motion in Accumulator
	Sign Bit	Magnitude	
Shift	0	00001	1 pos. to left
Rotate	1	11110	1 pos. to right
Rotate 16 bits only			
Shift complement	0	00001	1 pos. to right
Rotate complement	1	11110	1 pos. to left
Rotate 16 complement			

which a single position shift of all bits is triggered by successive clock pulses, were thereby ruled out. Instead, a semiparallel operation was used which allows all bits to be relocated by discrete groups. This reduces the number of shift operations.

Shifts involving any number of positions, within the 23-bit capacity of the accumulator, can be formed by successively adding together any three of the integers 0, 1, 4, 7, and 8, if we include the possibility of using an integer repeatedly. For example: $19 = 4 + 7 + 8$; $9 = 7 + 1 + 1$, or $9 = 8 + 1 + 0$; $3 = 1 + 1 + 1$; and so on. Based on shift groups of these values, all accumulator movements are realized as a combination of, at most, three jumps. The 0 group is trivial, of course, since it requires no shifting action.

After determining the direction of motion and performing any resultant complementation, the shift circuit translates the binary magnitude of the remaining five bits and sorts the results into the four categories. Each bit in the accumulator shift register is connected to its neighbors on both sides which are 1, 4, 7, and 8 positions away. Fig. 21 is a sketch of the bit positions of the accumulator showing the shift register connections for bits 10 and 17. Each bit provides an input to eight other positions, four on each side, to allow bidirectional motion. Bit 17 illustrates the circular nature of the rotation process, whereby a left rotation of 7 positions causes bit 1 to assume the value of bit 17. All bits of the register would be connected similarly for the shift orders and 23-bit rotation orders.

Further interconnections are required for the 16-bit rotation instructions that confine the rotation process to the particular 16 bits between positions 6 and 21. As shown in Fig. 22, a left rotation of 7 positions from bit 17 in this mode requires a connection to bit 8. Similarly, bit 10 is connected to bit 19 as well as to its other destinations. Double connections are shown between bits 10 and 18, and bits 17 and 9, to indicate that these destinations are equidistant from the source bits in

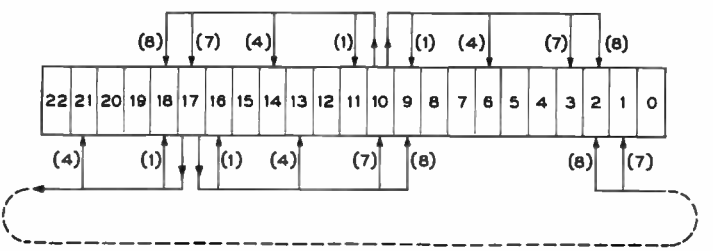


Fig. 21 — Interconnection of accumulator bits for rotate-shift functions.

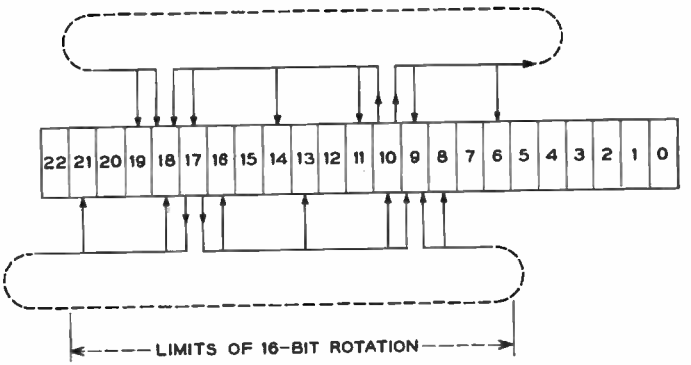


Fig. 22 — Bit interconnections for 16-bit rotation mode.

both directions. Bit 18, for example, assumes the state of bit 10 either in response to the instruction "shift (or rotate) all bits 8 positions to the left," or in response to "rotate 16 bits a distance of 8 positions to the right." These and similar patterns for other bits are redundant entry conditions and have been logically simplified in the final design.

Both the translated shift description and the bit interconnection logic are assembled in the final stages of the shift-rotate circuit. The results are combined with the three required clock pulses to trigger the synchronous operation of the shift register. Fig. 23 is a timing diagram illustrating the sequence of events which occur as a result of a shift or rotate instruction. During the initial 0.75-μsec interval, the instruction is recorded and the Q register is loaded with the 6-bit word that defines the direction and amount of shift that is to be applied to the binary word in the accumulator. Suppose that a shift instruction had been given, specifying that all accumulator bits be displaced 13 positions to the left. In binary form, the 6-bit description is 001101,

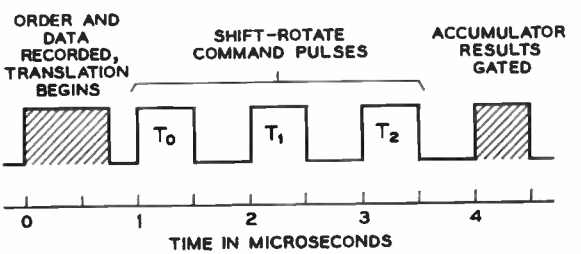


Fig. 23 — Timing diagram for shift register.

where the most significant bit signifies that motion is to be to the left. The shift translator decomposes the binary code into commands of jumps of 1, 4, and 8 positions. The first clock pulse of the chain, T_0, combines with the "shift 1" command, and the circuits in the accumulator prepare to respond. Activity commences on the downward swing of the T_0 pulse, causing all bits to move simultaneously one position to the left. At the same time, a 0 is inserted into bit 0. The ensuing 0.5 μsec interval allows the shift register stages adequate time to settle down in their new states. T_1 is combined with the "shift 8" signal. At its conclusion, all bits move 8 positions to the left, with 0's being inserted into the vacated spaces at the right. Finally, T_2 combines with "shift 4" to complete the action. Within 0.5 μsec after T_2, the new accumulator contents are gated out and the register is available for other use.

Table VII shows the translation of the shift magnitude from binary to the 1, 4, 7, 8 code and indicates the interval when each jump occurs. There are several limiting conditions which have not been previously

TABLE VII — TRANSLATION CODE FOR SHIFT-ROTATE

Shift-Rotate Code	Displacement	Jumps/Unit Time		
		T_0	T_1	T_2
0 0 0 0 0	0	0	0	0
0 0 0 0 1	1	1	0	0
0 0 0 1 0	2	1	1	0
0 0 0 1 1	3	1	1	1
0 0 1 0 0	4	0	0	4
0 0 1 0 1	5	1	0	4
0 0 1 1 0	6	1	1	4
0 0 1 1 1	7	0	0	7
0 1 0 0 0	8	0	8	0
0 1 0 0 1	9	1	8	0
0 1 0 1 0	10	1	1	8
0 1 0 1 1	11	0	7	4
0 1 1 0 0	12	0	8	4
0 1 1 0 1	13	1	8	4
0 1 1 1 0	14	0	7	7
0 1 1 1 1	15	0	8	7
1 0 0 0 0	16	8	8	0
1 0 0 0 1	17	8	8	1
1 0 0 1 0	18	7	7	4
1 0 0 1 1	19	7	8	4
1 0 1 0 0	20	8	8	4
1 0 1 0 1	21	7	7	7
1 0 1 1 0	22	7	7	8
1 0 1 1 1	23	8*	8*	8*
1 1 0 0 0	None			

* Results in resetting accumulator on shift instruction. No action on rotate instruction.

discussed. When either 0 displacement or a magnitude greater than 23 is specified, the shift-rotate mechanism is inhibited. An exception is the 16-bit rotation instruction, which rotates the bits "modulo 16," i.e., $n - 16$ positions.

IX. EQUIPMENT DESIGN

9.1 *General*

Apparatus counts based on early block diagrams showed that the central control would be so large as to require precise control of logic circuit locations and their interconnections. With these concerns in mind a "breadboard" version of the central control's arithmetic system was built and tested. This unit, dubbed SPADE (Stored Program Arithmetic Digital Exerciser), contained wired LLL circuit packs tailored to meet many different combinational patterns and utilized point-to-point wiring between connector terminals.

SPADE provided the testing ground for prototype semiconductors, LLL combinational patterns, and many system concepts. The four-bay central control reflects the circuit pack designs, wiring patterns, and logic pack arrangements resulting from these early studies.

9.2 *LLL Circuit Packs*

Plug-in circuit packs[8] to accommodate the various combinations of gates, flip-flops, etc., were designed to allow rapid isolation and replacement of suspect components, thereby facilitating system maintenance. The design approach was to provide accommodations for various quantities of separate, independently accessed LLL gates on specific packs. Logic combinations are formed by interconnecting suitable gates at the plug-in connector terminals. This approach is in contrast to that of designing a large variety of low-usage packs in which the gates are internally wired to generate specialized functions.

To achieve high density, it is important to place as many logic gates as possible on a circuit pack and to use them with the greatest efficiency. The controlling variables in the design are the circuit pack area and the number of connector terminals available (28). With these limitations, the most densely packed logic element contains eight two-input gates. This pack, shown with its apparatus mounting in Fig. 24, has the largest usage in a No. 1 ESS office. Table VIII contains a listing of other LLL-type circuit packs used in the system, predominantly in the central control.

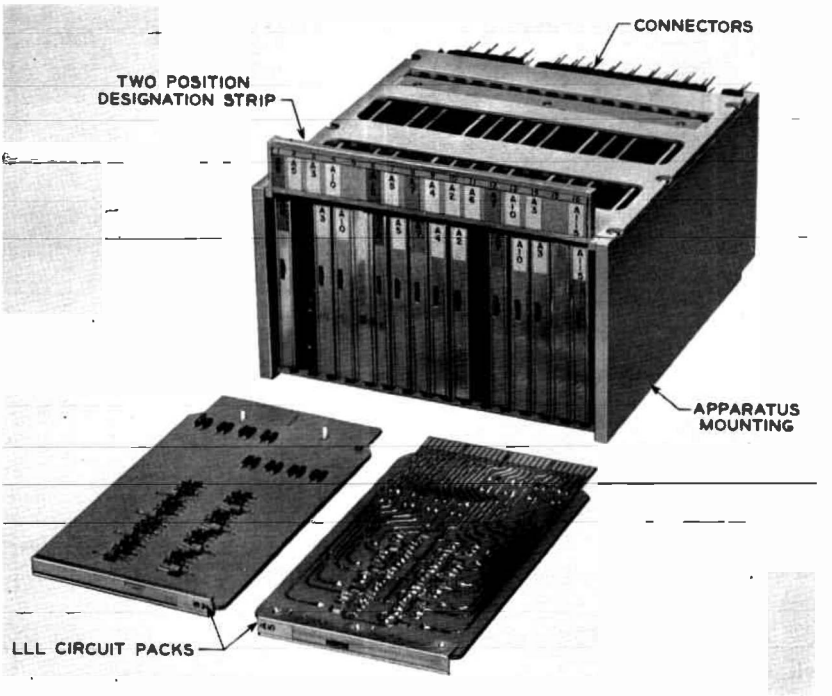


Fig. 24 — LLL circuit pack and mounting.

A measure of the success of logic circuit standardization is best illustrated by the fact that sixteen logic circuit packs, making up about 10 per cent of the entire No. 1 ESS catalog, account for more than 50 per cent of the packages and 70 per cent of the semiconductors used in a typical office. In the No. 1 ESS office at Succasunna, New Jersey, 6,675 of these sixteen types are required, the duplicated central controls alone accounting for 4,036.

Previous sections have pointed out that LLL gate outputs can be connected together (with certain restrictions) to form a common driving AND node. When this is done, a single dummy load resistor (Section 4.4) is sufficient to maintain the proper operating conditions for the combined circuits. To facilitate this use, the resistors are omitted from the logic packs and are assigned instead to resistor packs. These contain twenty-four separate collector resistors commonly connected to the +4.5-v collector source. A single LLL gate then receives its base drive through its own internal base resistor supplied from a common +24-v

TABLE VIII — LLL-TYPE CIRCUIT PACKS

Code		Description
A1	2	9-input LLL gates
A2	3	7-input LLL gates
A3	4	5-input LLL gates
A4	6	3-input LLL gates
A5	8	1-input LLL gates
A6	8	2-input LLL gates
A15	8	1-input LLL gates (individual power connections)
A7	8	1-input high fan-out LLL gates
A8	8	2-input high fan-out LLL gates
A11	1	binary counter stage
A13	2	flip-flops with input gating
A14	1	shift register stage
A18	8	high-sensitivity bus receivers
A19	8	bus receiver amplifiers
A21	8	bus driver transformers
A177	8	bus driver LLL gates

power terminal on the connector and its +4.5-v collector source through an assigned external resistor at the collector output terminal.

9.3 *Circuit Pack Interconnection Practices*

The interconnection of logic gates to generate a circuit function is dependent upon the over-all circuit organization and the diagnostic and maintenance techniques which are applied in its operation.

Because of its word-organized design, faults and errors in a CC are most easily diagnosed to the troublesome bit position in a word. It seems only natural, then, that as many logic gates as possible associated with the same bit of a word be assigned to the same circuit pack. In effect, this approach to logic gate assignment keeps to a minimum the number of circuit packs which must be tested to isolate a particular fault. Unfortunately, assigning logic gates to circuit packs on a word-bit basis is often incompatible with the approach which assigns logic gates on the basis of achieving high circuit pack density. As a result, it is often necessary to compromise and to assign two or more bits to the same circuit pack.

Output leads from flip-flops, binary counters, and shift registers are critical because of their internal cross-connections to inputs; therefore the total wire length is restricted to a maximum of three feet. To some degree, this requirement also affects the assignment of logic gates to circuit packs. The following are examples of circuit pack assignment practices as applied to the central control.

Fig. 25(a) shows a common-collector configuration where all logic

gates can be assigned to the same circuit pack if no input lead from a register exceeds three feet.

Fig. 25(b) shows a common-collector configuration where all logic gates are not assigned to the same circuit pack. These logic gates are assigned to three different circuit packs as follows: one 7-input logic gate to an A2 circuit pack, three 5-input logic gates to an A3 circuit pack, three 3-input and two 2-input logic gates to one A4 circuit pack. In this case, the 3-input circuit pack is used where the circuit logic requires 2-input gates. Had the 2-input gates been assigned to an A6 pack, four types of circuit packs would have been required for this word bit, rather than three.

9.4 *Frame Wiring Methods*

During the brassboard development stage it became evident that the shop and field wiring practices of relay switching systems could not be

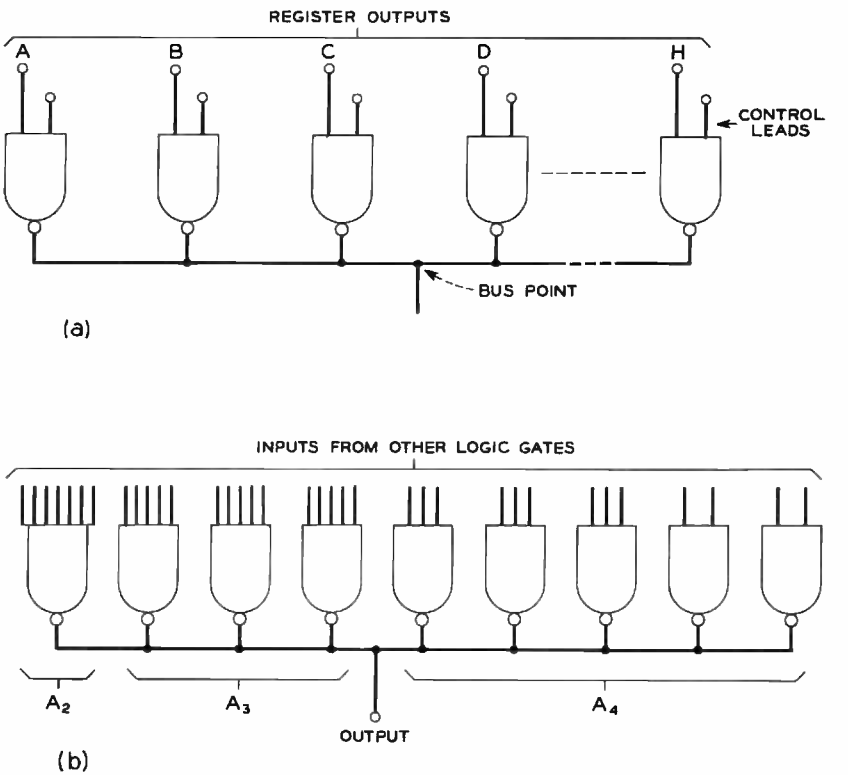


Fig. 25 — Common-collector configurations.

used for the high-speed pulse circuit wiring of central control. The lengths and parallel spacings of signal wires had to be controlled to minimize electrical interference (crosstalk) between circuits. This resulted in circuit pack placement restrictions and the development of new wiring methods suitable for this type of equipment.

Where high densities of circuit packs and wired interconnections exist, special wiring procedures are employed which dictate specific routings for surface wiring, loose wiring, and local cables. This eliminates the shop wiring variability that could appear in surface wiring by different operators.

The individual mounting plates are first surface-wired in the shop and verified with a dc test. The associated mounting plates of a unit are next surface-wired together. The loose wiring support details are then mounted on the units, and the units in turn installed on the bay or frame. A preformed loose wiring harness is mounted on the support details and the leads are terminated.

The loose wiring support details are epoxy-coated metal combs and fingers which are grounded to the mounting plate via their mounting screws. These provide an extension of the ground plane to minimize inductive coupling between the loose wire signal paths. The high wiring density shown in Fig. 26 requires the use of no. 26 gauge wire.

An analysis of the CC wiring disclosed the following distribution on the four-bay frame:

(a) surface wiring per mounting plate (average): 450 wires between connectors on the same plate, and 100 wires between connectors on different mounting plates of the same unit

(b) loose wiring per mounting plate: 50 wires between connectors on different units.

(c) local cable wiring per mounting plate: four wires to fuse panel

(d) interframe loose wires between bays 1 and 2: 750 wires.

Approximately 43,300 wires or 86,600 wrapped connections are required within the four bays.

9.5 *Grounding Techniques*

Because of the rather small voltage swings necessary to control the transistor circuits, and the speed with which they react, the grounding technique is very important. A very low impedance (inductance) ground plane is required to prevent excessive noise voltages (a few tenths of a volt) from being generated by the high-speed switching circuits in the CC. For this reason, the equipment frame is utilized as

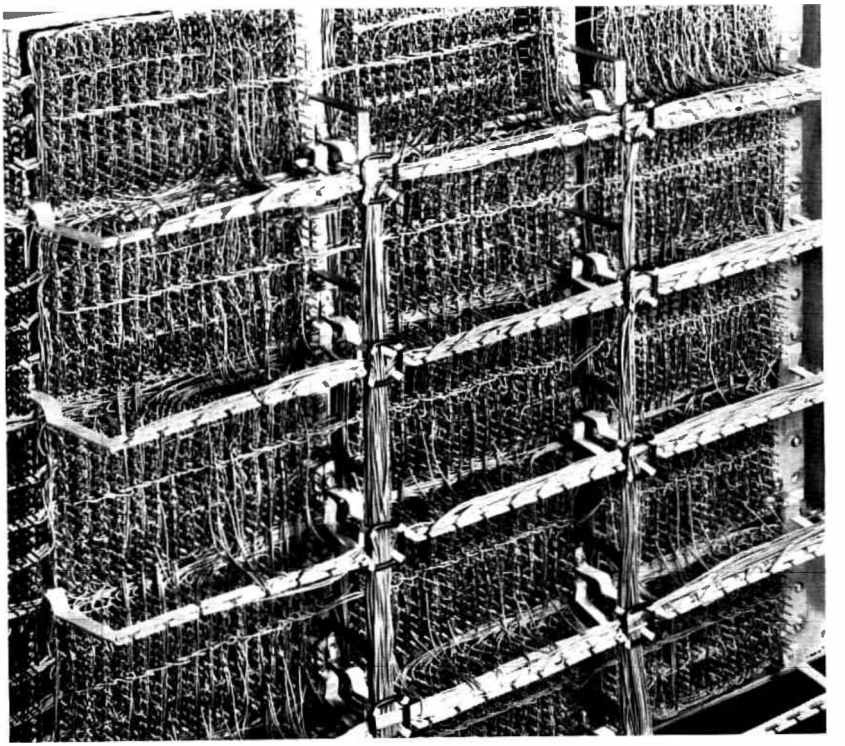


Fig. 26 — Photograph of central control wiring.

a circuit ground. A ground terminal intimately connected to the frame is located immediately below each circuit pack. Each circuit pack has its circuit ground tied into frame ground with a wire approximately three inches in length. These precautions have reduced the ground noise internally generated in the CC to a level where it is not a problem. The power and grounding arrangements in the system are relied upon to keep extraneous circulating currents from creating disturbing voltage drops in the framework.

X. SUMMARY

The design of the central control has been discussed in terms of its influence upon the design of building block logic circuits and their utilization in performing some of the specialized functions which are carried out in the process of controlling a telephone switching system. The complexity of the unit made a detailed description of each internal

circuit beyond the scope of this paper; therefore many features have had to be omitted.

The emphasis for reliability has been perhaps the most important factor dictating all CC design decisions. A major part of the circuit and equipment design effort was directed toward enhancing dependability by prescribing inherently reliable circuits, components, and equipment. The logic circuit design provides standardized but versatile building blocks with fast, low cost performance. Over 10^9 LLL gate hours have been logged during the No. 1 ESS preoperational phase. The failure rates during this phase are, as might be expected, higher than will be experienced when installation troubleshooting is completed. Based on current reliability predictions, about one component failure per operating month is anticipated in each CC. Because of the internal fault recognition and self-diagnostic maintenance features of the system, along with the provision for rapid repair by circuit pack replacement, this failure level is sufficiently low to insure the requisite system dependability.

XI. ACKNOWLEDGMENTS

The authors wish to thank their many colleagues who contributed their efforts and talents to the development of this project. Particular credit is due R. C. Buschert, S. F. Sampson, and S. M. Neville for their work in evaluating the devices and developing the basic circuits; also to R. W. Donalies, R. A. Estvander, E. Graeve, F. R. Huff, A. V. Jensen, R. Korte, A. Perretto, and J. Pindroh for logic designs; to A. C. Luebke, I. W. Morris, R. Nelsen, W. E. Smith, A. M. Wagner, and H. J. Wirth for equipment designs; and to J. B. Connell, A. H. Doblmaier, R. W. Downing, M. P. Fabisch, J. S. Nowak, F. F. Taylor, and W. Ulrich for their cooperation in relating the CC functional requirements to practical designs. The assistance of the semiconductor device development areas in the practical characterization of the semiconductors is also gratefully acknowledged.

REFERENCES

1. Downing, R. W., Nowak, J. S., and Tuomenoksa, L. S., No. 1 ESS Maintenance Plan, B.S.T.J., this issue, p. 1961.
2. Connell, J. B., Hussey, L. W., and Ketchledge, R. W., No. 1 ESS Bus System, B.S.T.J., this issue, p. 2021.
3. Harr, J. A., Taylor, F. F., and Ulrich, W., Organization of No. 1 ESS Central Processor, B.S.T.J., this issue, p. 1845.
4. Yokelson, B. J., Cagle, W. B., and Underwood, M. D., Semiconductor Circuit Design Philosophy for the Central Control of an Electronic Switching System, B.S.T.J., **37**, Sept., 1958, pp. 1125–1160.

5. Cagle, W. B., and Chen, W. H., A New Method of Designing Low-Level, High-Speed Semiconductor Logic Circuits; I.R.E. Wescon Convention Record, Vol. 1, Part 2, 1957, pp. 3–9.
6. Jarvis, D. B., The Effects of Interconnections on High-Speed Logic Circuits, IEEE Trans. Electronic Computers, Oct., 1963.
7. Dodson, G. A., and Howard, B. T., High Stress Aging to Failure of Semiconductor Devices, Proc. 7th Natl. Symp. on Reliability and Quality Control, Jan., 1961, pp. 262–267.
8. Chevalier, J. G., and Eisenhart, R. K., No. 1 ESS Circuit Packs and Connectors, B.S.T.J., this issue, Part 2.

No. 1 ESS Program Store

By C. F. AULT, L. E. GALLAHER, T. S. GREENWOOD and D. C. KOEHLER

(Manuscript received January 15, 1964)

Line and trunk translation data and operating programs for No. 1 ESS are stored in a large semipermanent memory. This memory is provided by modular units known as program stores. Each program store provides 5.8 million bits of randomly accessible permanent magnet twistor memory organized into 131,072 parallel words. The information is stored in the state of small magnets affixed to aluminum cards. Each card contains 64 forty-four-bit words.

Each store is designed to operate over a duplicated common bus system for both normal and diagnostic operations. The stores have a cycle time of 5.5 μsec. Such stores are an attractive and economical solution to the problem of providing large storage capacity for information which must be protected against accidental change.

To provide an efficient and routine method for updating the information content of such stores, offices are provided with card writing equipment. This includes both card handling equipment and card magnetizing equipment under system control.

I. GENERAL

1.1 *Storage Requirements*

No. 1 ESS is under control of a very large program. The complete program may require a storage capacity in excess of 4 million bits. In addition, the office must store a considerable amount of data about each line and trunk. Such data as directory number, class of service, special feature lists, etc. must be available internally to the system. The storage needs for these items are variable with office size and range from 1 to 14 million bits. This program and translation information together constitute (a) the knowledge necessary to perform the telephone switching function and (b) the memory of the service commitment to each customer. To ensure service continuity this information must be

protected from accidental destruction by either equipment malfunction or operator error.

The system must have direct and immediate access to all of the information. However, it will be noted that the system does not require the direct capability of altering the information. This is because the information and its changes are generated externally to the system, as by a program designer or the telephone business office. What is required is a suitable way of introducing the information changes into the system.

To meet this need of an economical, high-capacity, random-access memory, No. 1 ESS uses permanent magnet twistor modules as basic storage elements. These provide a memory that is fundamentally "read-only" while at the same time providing a simple and straightforward way of replacing old information with new. As desired, no electrical malfunction can alter the information content.

1.2 *Store Objectives*

Twistor modules, with their circuitry, form an ESS unit designated a "program store." The general objectives set for the development of program stores are given in Table I. The word size of 44 bits allows 37 bits of information and 7 bits of redundant encoding.[1] The high-reliability requirement, coupled with past experience in the Morris, Illinois, ESS trial, indicated strongly the desirability of high redundancy in program information. The coding used includes a Hamming single error correcting code plus an over-all parity, both taken over the data and its storage address. While with equipment operating normally this much redundancy may seem extravagant, the ability to operate under serious degradation of circuit or memory, as well as the enhanced ability to detect and isolate malfunction, is felt to more than repay the cost.

The capacity of 131,000 words represents the basic storage needs of a small office. It was felt that this size also represented about the largest size that could be achieved with common-access circuits. The cycle time of 5.5 μsec was essentially determined by the over-all system,

TABLE I — ESS PROGRAM STORE OBJECTIVES

Word size	44 bits
Capacity	131,000 words
Speed	5.5 μsec
Cost	low
Reliability	high
Maintenance	automatic
Power	battery
Environment	non-air-conditioned

including the twistor modules, but did not represent an attempt to build the fastest store possible. Low cost is of course an inherent objective; an over-all economic balance in the system dictated the need to achieve a cost per bit of the order of one-fifth that of the system's variable memory.

Programmed fault diagnosis is an essential feature of an ESS. In the case of a program store, the interaction between the fault and the fault-detection program can be especially severe. This requires that advance planning and coordination between the circuit, system, and programming engineers be especially effective.

The requirements of battery power and non-air-conditioned space are office-wide. They provide somewhat more of a challenge in the case of the program and call stores, where temperature-sensitive devices and nondigital circuits occur.

The over-all objectives, at the time they were established, represented a goal that would require significant device and circuit development. This article reports the successful attainment of that goal.

1.3 *Devices*

1.3.1 *Semiconductors*

As in the remainder of the system, in digital or low-level applications the store used the single code of transistor developed for No. 1 ESS. However, it was clear at the outset that a fast, higher-powered transistor and companion diode would be required for program store access. For this purpose the 20D transistor and 426AC diode were developed. The 20D transistor, which can handle 1.35 amps with $\frac{1}{3}$-μsec switching times and a breakdown voltage of 50 volts, also proved a suitable transistor for other fast, high-power needs of the system.

1.3.2 *Twistor Modules*

The permanent magnet twistor memory has been previously reported.[2,3,4,5] The twistor modules used in ESS were a further development of the module described in Ref. 4. The front and rear views of a module are shown in Fig. 1. On the rear is a 64 × 64 biased-core switch matrix. A particular core is switched by the combined action of a horizontal and vertical half-select current overcoming a common bias current. Each core is coupled to a strip solenoid referred to as a "word solenoid" (see Fig. 2). The word solenoids are attached to the surface of an insulating board over which have been placed thin sheets of permal-

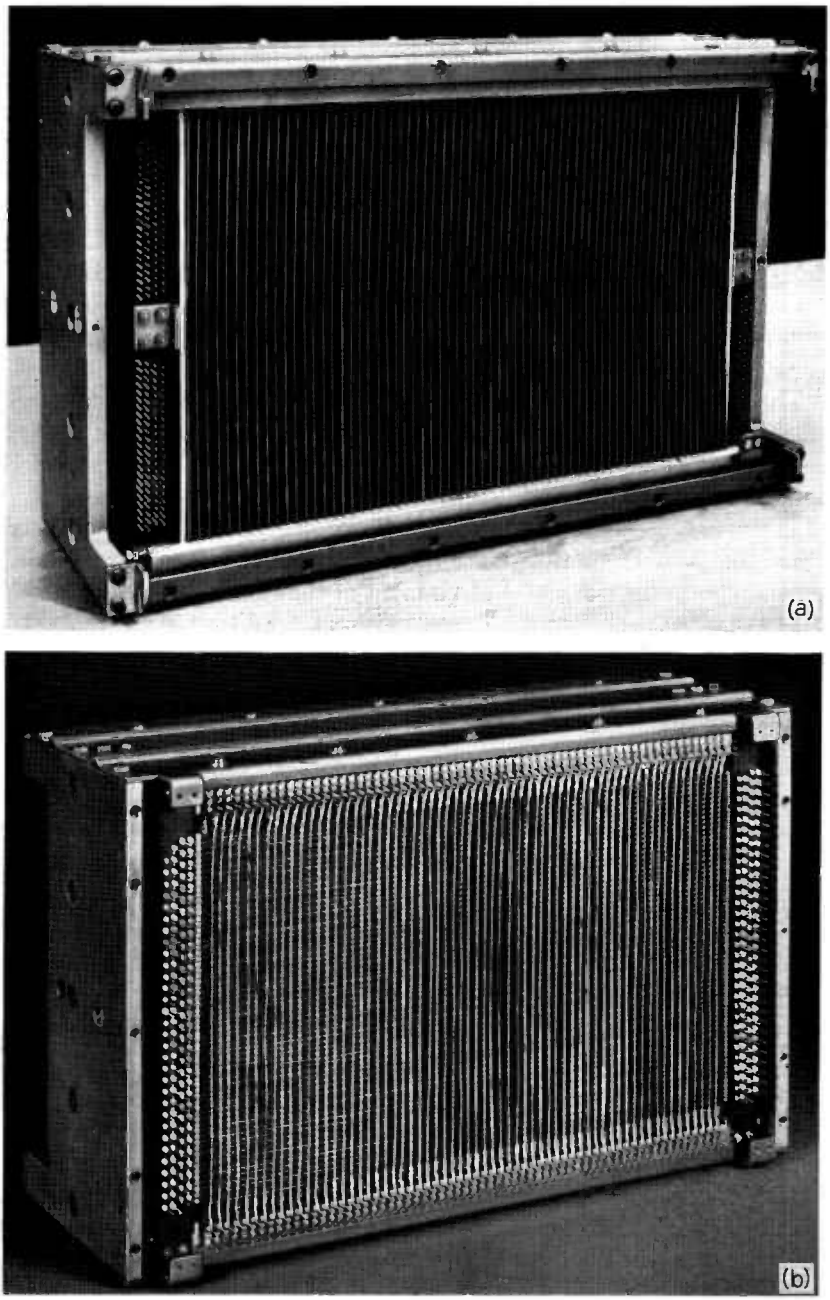


Fig. 1 — (a) Front (card insertion side) of a 1A twistor memory; (b) rear (core access side) of memory.

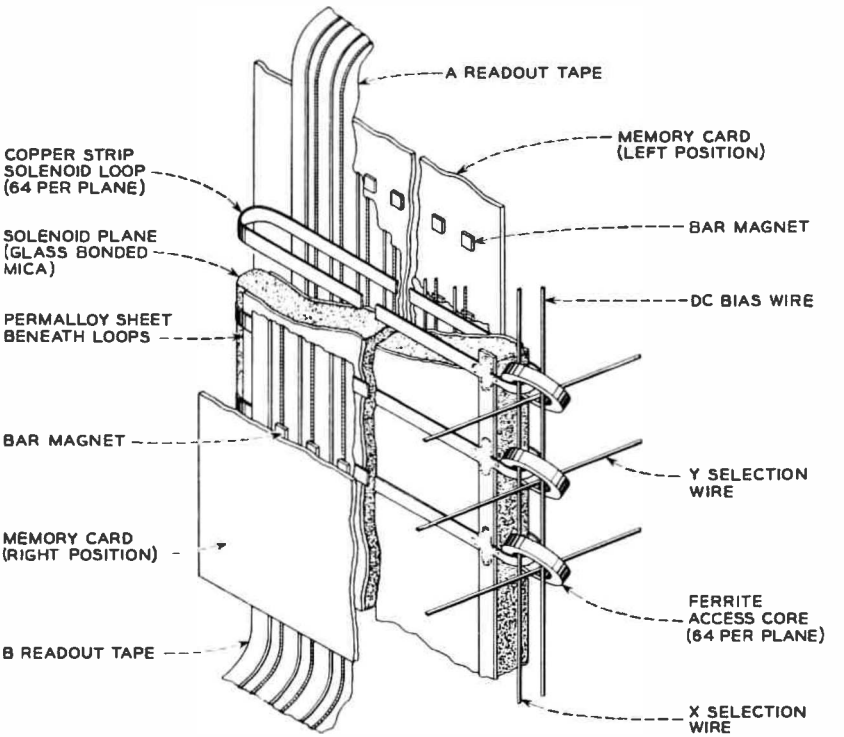


Fig. 2 — Cutaway schematic showing the basic elements of permanent magnet twistor memory.

loy. Each module contains 64 boards, each with 64 solenoids. Through the module, accordion fashion, run two flat plastic belts, each containing 44 twistor wires and 44 adjacent return wires. A section of each tape is cemented to one side of each board. Each of the modules provides space for 128 magnet cards, one of which is shown in Fig. 3. Each aluminum card carries 64 columns of 45 thin permanent magnets. Each column represents a word and each of the first 44 magnets, a bit of the word. (The 45th row is not used in this application.) When magnetized, each magnet represents a stored "zero." A demagnetized magnet represents a stored "one." When all cards are in place in the module, a magnet appears over each intersection of a twistor wire and word solenoid. If the magnet is magnetized, it fully saturates a region of the twistor wire beneath it. The magnets are always magnetized in the same direction as the initial field of the word solenoid.

When an individual word solenoid is selected by applying a half-

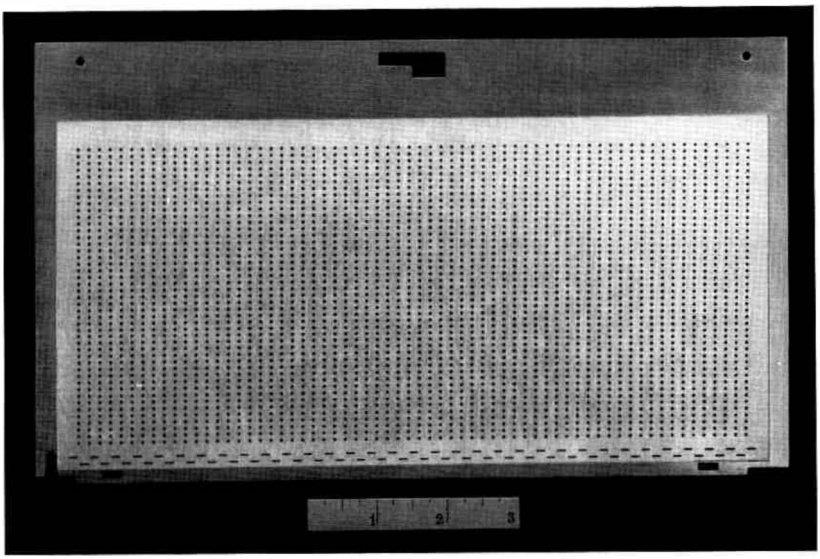


Fig. 3 — Twistor memory card.

select current to individual horizontal and vertical access wires, a current pulse is induced in the word solenoid. The resulting magnetic field acts on the twistor wire. Because of the orthogonal geometry no significant voltage is induced in the wire unless the magnetic material of the twistor wire is switched. This can occur only at the sites containing nonmagnetized magnets. When the half-select currents are removed, the common bias current switches the selected access core back to its initial state. The resulting word solenoid pulse restores initial twistor wire conditions.

It will be seen that the selection of a single word solenoid causes a readout of 88 bits. Within the readout circuits, it is thus necessary to select the 44 bits associated with the desired magnet card.

For uniform outputs from the memory, the magnetic field applied to the twistor wire should completely switch its magnetic material. In these modules the magnetic field produced by the solenoid is concentrated onto the twistor wire by two mechanisms. The first is the underlying permalloy sheet, which provides a low-reluctance return path for the field; the second is the conductivity of the magnet card, which produces, by eddy currents, a magnetic barrier above the twistor wire. These two mechanisms help to reduce the drive currents required and the interaction between bits.

In these modules, a 2.62 ampere-turn bias and half select are used. These produce a "one" output at the end of a twistor pair of 2.5 mv

across 300 ohms (far end short-circuited). For a bit with a magnetized magnet, essentially no output is generated in the twistor wire. However, because any selected access core is always accompanied by 126 half-selected cores, some "shuttle" or "delta" noise may be generated due to the summation of small voltages at each of these solenoids. This delta noise is information dependent. It may subtract from a "one" signal as well as create a "zero." The modules used are designed to insure a 2.5-to-1 ratio of "one" to "delta" for worst drive, temperature, and information pattern over the complete population of manufactured modules. Under nominal conditions of drive and temperature almost all bits will exceed 5-to-1 with worst pattern.

1.4 *Store Organization*

A complete program store is shown in Fig. 4. The store circuitry contains three major divisions: access, readout, and control. Each of these will be treated in detail in this article. The function of each is as follows:

1.4.1 *Access*

To achieve the required capacity, 16 twistor modules are used in the store. These are arranged in a 4×4 array which results in a 256×256 access core matrix. The access circuits must provide this array two pulses of accurately controlled shape, amplitude, and coincidence to cause the readout of the desired word.

1.4.2 *Readout*

The readout section must provide the amplification of the low-level twistor signals and the selection and sampling necessary to provide the binary output word. The twistor wire signals all appear on terminals on the front surface of the modules. Because of the large area of the core matrix and the low level of the twistor signals, the readout leads must be well shielded to avoid interference. Ideally, the 88 twistor pairs from each module could be paralleled before amplification, since only one solenoid is accessed at a time. For two reasons, this is not done: first, it would result in a considerable reduction in available output power; second, and more importantly, the delta noise would be several times that from a single module. On the other hand, individual handling would require 1408 preamplifiers. In the interest of economy a compromise solution is used. The modules are paralleled in groups of four. Each module in the group is chosen from a different row and column so that no delta noise penalty is incurred. The 6-db loss in power is not great enough to significantly degrade signal-to-noise performance.

The selection of the desired group of 44 pairs from the four groups of 88 pairs is made after a small amount of amplification. A single group of

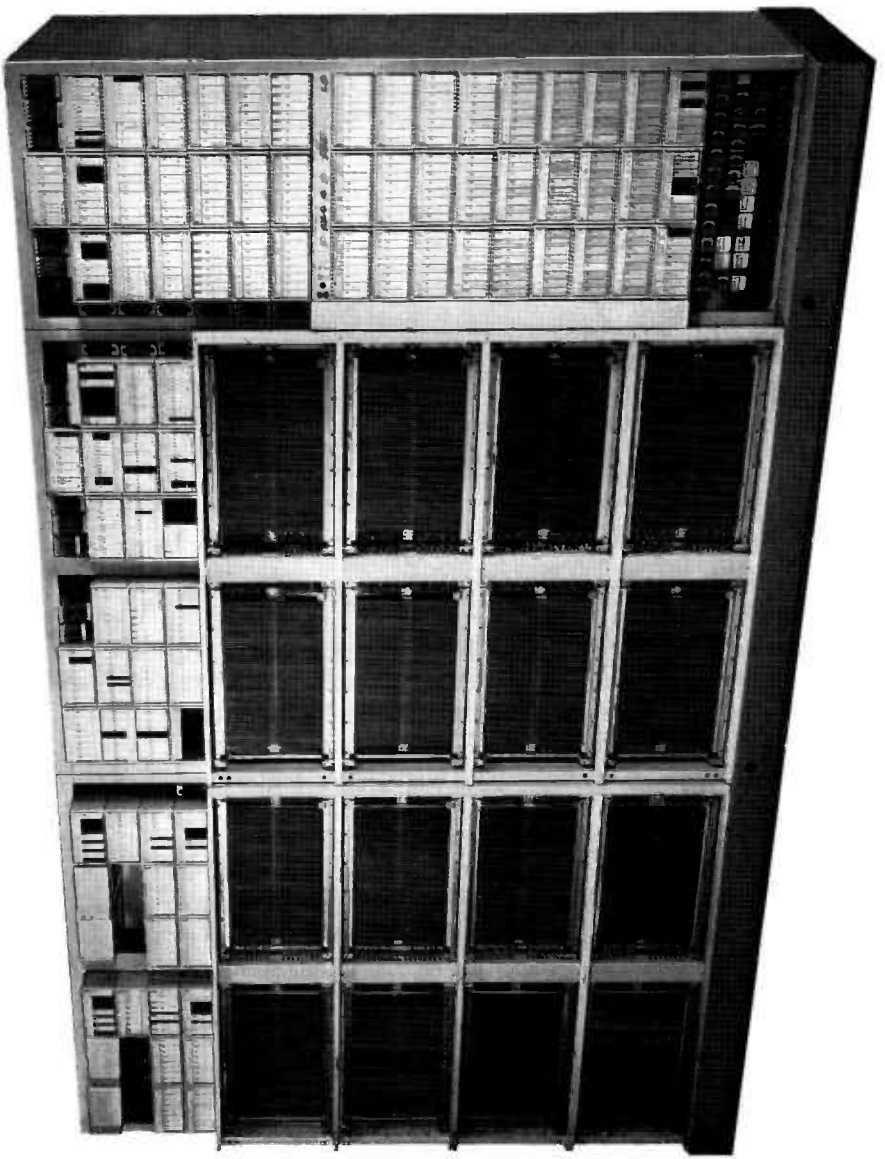


Fig. 4 — No. 1 ESS program store.

44 readout amplifiers and samplers is used to generate the final quantized one and zero signals. (See Fig. 5.)

1.4.3 *Control*

These circuits must provide all of the communications with the remainder of the system, both for normal action and maintenance. Additionally, they must make continual checks to detect abnormal conditions and initiate diagnostic action. Duplicate communication circuits to the central controls are provided. This duplication requires control circuits which must have further communication paths. To provide flexible growth a common bus system is used for all stores. Thus each store must have circuits for recognizing coded bus signals before responding. To insure that effective automatic diagnosis is possible, logical separation of circuit faults must be made possible by judicious provisions of circuit and communication redundancy. An indication of the complexity of this function is that nearly one-half of the store circuitry is in this section.

1.5 *Card Writing*

The insertion of information into the program store requires the magnetizing or demagnetizing of individual magnets. To meet the

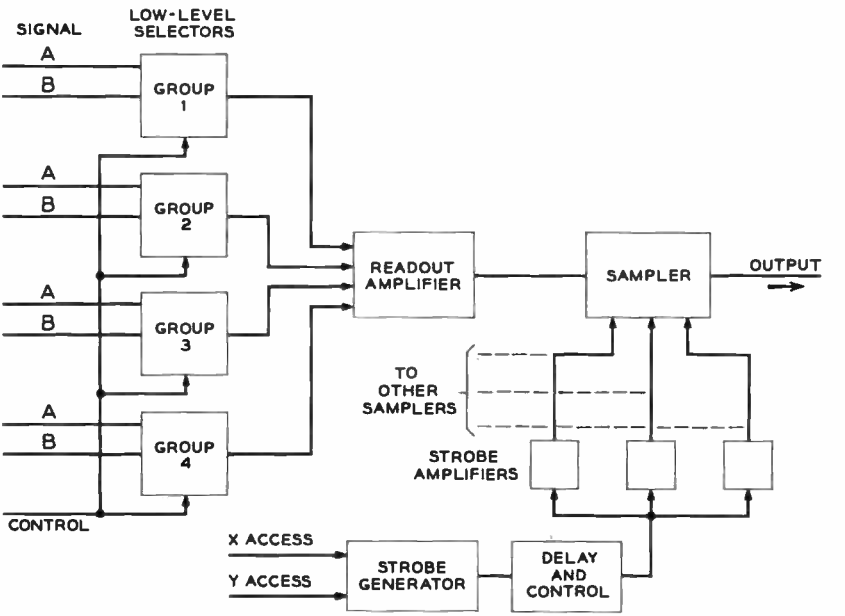


Fig. 5 — Readout block diagram.

system objective of a simple and straightforward method for accomplishing this, two auxiliary units are used: (1) the magnet card writer, which provides a mechanism for handling cards and a movable magnetic head for magnetizing or erasing magnets, and (2) a card loader which provides an automated method of removing or inserting a full module of cards from or into a store and supplying them to the card writer.

The card writer is used by the system as a peripheral unit. It is supplied information based on old program store information modified by the changes desired. Using this information the card writer magnetizes and erases a spare set of cards. Once prepared, this set may be substituted for the old set in the program store. This process of updating translation information in the office is straightforward and insures that the new information is correct before the old information is destroyed.

II. STORE ACCESS

2.1 *General Philosophy*

2.1.1 *Coincident-Current Word Selection*

Word selection is obtained in the No. 1 ESS program store by means of a coincident-current selection method utilizing a ferrite biased core switch. Normally, all of the access core switches are biased with 2.62 ampere-turns, which in the absence of other drives causes the ferrite core to be in a state of saturation. The core has a squareness ratio, B_r/B_{max},

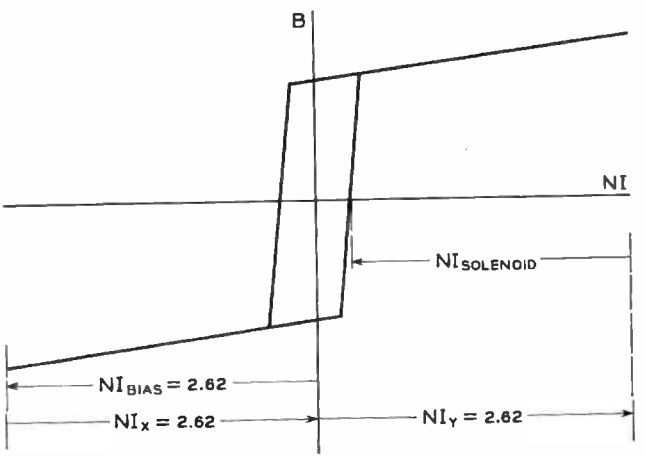


Fig. 6 — Operation of biased core switch.

of about 0.95, together with a coercive force (H_c) of about 0.2 oersted. Thus the state of the core is much as shown in Fig. 6. Also shown are the driving forces, NI_x and NI_y, applied to the biased core switch. It is obvious from the figure that either drive by itself is insufficient in magnitude to cause any switching action in the core. On the other hand, if both drives are present at the same time, the core will begin to switch. During the switching interval, a voltage will be induced in the solenoid winding which will cause a current in this winding that will flow in a direction such as to oppose the drive current. Thus the magnitude of the ampere-turns drive in the solenoid is equal to the sum of the two drives minus the bias, the coercive force, and appreciable losses due to high-speed switching of the core.

If the drives were maintained on the biased core switch, the core would eventually saturate. For the duration of the pulse required (2 μsec), a 400-mv-μsec core provides ample margin for worst-case considerations of temperature, drive currents and solenoid impedances. Normally, the core is only partially switched at the time the drive currents are removed. Upon removal of the drive currents, the bias current switches the core back to its initial, or set, state. This results in a reverse current flow in the solenoid winding that resets all the twistor wire bits. Such action points out one great advantage of the biased core switch for memory access: unipolar drive currents in the core lead to bipolar pulses in the solenoid. This provides for automatic resetting of the memory word bits after interrogation.

2.1.2 *Size of Access Matrix*

The size of the access matrix is necessarily a balance between economy and peak power requirements. Generally speaking, the cost of memory on a per bit basis is less as the size of the memory increases. This is particularly true of coincident-current access memories. The larger memories, however, require an increased drive power which results in several problems. One is the problem of generating the high-power drive pulses while meeting the system cycle time and another is the problem of interference between the access and readout signals.

The No. 1 ESS program store has a single access system. The access matrix is 256 $\times$ 256. That is, there are 256 X-drive windings and 256 Y-drive windings, together capable of selecting one of the 65,536 cores used as biased core switches. The load presented by the winding consists of three parts. The primary load consists of the air inductance of the access winding itself. The secondary load is the shuttling of the 255 cores which are not driven by the other set of drive windings. Of little con-

sequence is the load presented by the selected core. This is not to say that the core properties are not of great importance. The squareness ratio determines the load presented to drivers when the 255 cores are shuttled. The smallest core that has the requisite output is desirable, since larger cores increase the shuttling load.

The choice of the number of turns to be used on the biased core switches for the drive windings is dependent on several factors. The most important factor is the semiconductor to be used and its ratings. The 20D transistor is specified primarily for its needs in the access circuitry. The transistor has to have fast turn on and off times, with 1.35-amp current capability, and be able to interrupt a 50-v level without breaking down. These are approximately the values of current and voltage encountered with two turns per core. Operation with three turns requires excessive voltage ratings on the transistors and lowers the resonant frequency of the drive winding into the frequency band used by the pulse drivers. Single-turn operation would be desirable if an adequate transistor were available. Its advantages do not appear sufficient to warrant paralleling three 20D transistors, since this would entail added cost and circuit complications. As a result, a two-turn winding was chosen.

The circuit access matrix for one axis is shown in Fig. 7. It is basically a simple diode matrix with 16 access switches on each side of the matrix. A closure of one access switch each in the upper set and the lower set will establish a path through the matrix, thus selecting 1 out of 256 drive windings. As mentioned previously, each of the drive windings drives 256 cores. The matrix diodes prevent sneak paths from occurring which would subtract from the current pulse desired in the selected winding. They are normally back-biased to a voltage slightly higher than the peak drive voltage. This protects the access switch from a transient turn-on problem that could degrade the shape of the leading edge of the drive pulse. The diodes above the upper switches and below the lower switches also serve to protect the transistors of the associated switches from voltage pulses which could turn on a switch on a transient basis. These diodes are also normally back biased.

2.1.3 *Access Waveform*

Fig. 8 shows the basic waveforms of current and voltage as related to the twistor modules. Traces (a) and (b) represent the drive currents, I_x and I_y, that drive a particular core in the memory. Trace (c) shows the current in the word solenoid. The solenoid current shows the bipolar pulse characteristics plus the exponential recovery portion. The ex-

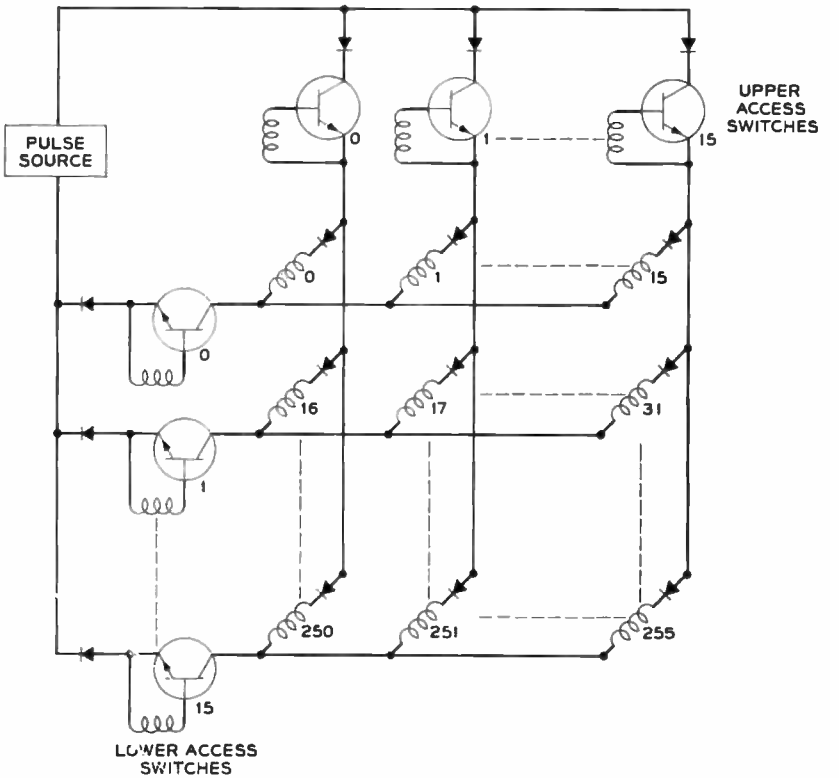


Fig. 7 — Access matrix for one axis of selection.

ponential decay occurs when the biased core switch returns to saturation. At this time the core is essentially a short circuit to the word solenoid. Thus the inductance and resistance of the solenoid determine the rate of decay of the solenoid current. This L/R time is important, since another interrogation cannot be started until the solenoid is nearly recovered. The program stores are operated at a 5.5-μsec cycle rate in the No. 1 ESS system but could be operated as fast as 4.5 μsec without degrading the solenoid current waveforms.

The drive currents are "on" for 2 μsec in the program store. This is primarily so that the readout can be detected before the transient caused by turn-off occurs in the readout circuitry. The approximate waveform of readout from the memory is shown as trace (d). It is included for timing reference only and is explained in some detail in the readout section of this article. The drive currents must be left on long enough to

switch enough flux in the biased core switch to insure a sufficient reverse current in the word solenoid to reset the twistor bits.

The rise and fall times of the drive currents are approximately 0.5 μsec. The shape of the current waveform is nearly trapezoidal in order to obtain the desired current level in a minimum amount of time without exceeding the drive transistor voltage limits.

2.2 *Current Pulse Generating System*

2.2.1 *Basic System*

The basic current pulse generating system for one axis consists of a constant-current source connected to a parallel circuit consisting of the load (access matrix) and a normally closed switch, as shown in Fig. 9. Normally the current source feeds current through the closed switch, A. When switch A is opened the current is forced to flow through the load and the normally closed B switch to ground. This presupposes that the proper access switches in the load section have been closed prior to opening switch A. Since the load is primarily an inductance, it is necessary to limit the voltage across the parallel circuit to prevent damage to switch A. For this purpose, limiter A is placed directly across switch A. Thus, when switch A opens, the limiter takes all of the current but places a fixed voltage across the load and switch A, causing a linear

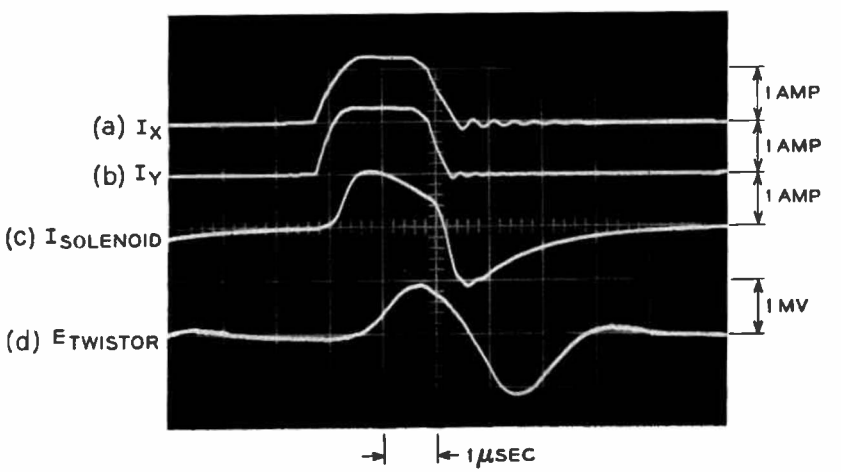


Fig. 8 — Access waveforms; (a) horizontal select current, 1 amp/division; (b) vertical select current, 1 amp/division; (c) drive solenoid current, 1 amp/division; (d) twistor output, 1 mv/division; horizontal scale, 1 μsec/division.

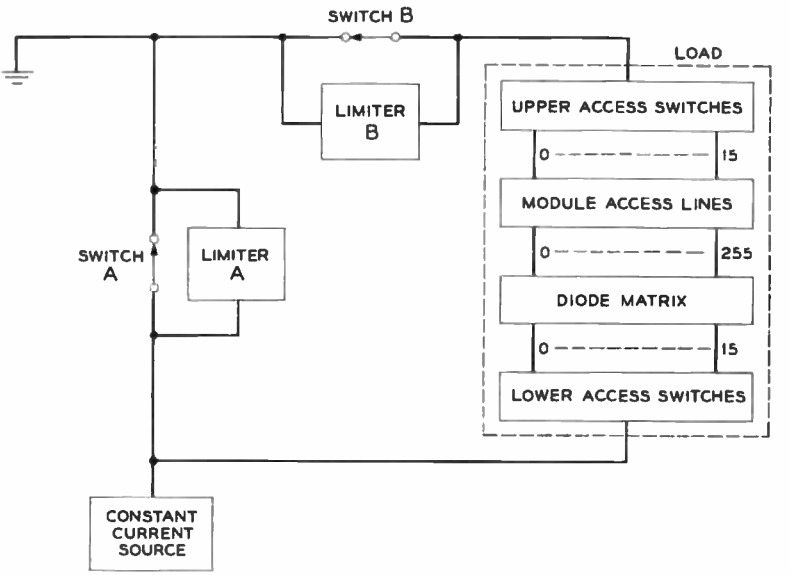


Fig. 9 — Basic access pulse generation.

buildup of current in the load. This continues until the load current equals the source current, at which time the load current remains fixed and the voltage on the parallel circuits drops to a low value determined primarily by voltage drops of the various semiconductor devices.

This mode of operation is uniquely suited to the twistor. In the twistor the inductances of the drive lines are all well matched and independent of the memory contents. Thus the application of a constant voltage results in accurate control of the rise time.

A similar method is used to effect turn-off. Again referring to Fig. 9, switch B is opened and switch A is reclosed. The inductance of the load must now drive limiter B. Thus a constant reverse voltage occurs across the load until the load current drops to zero. Switch B can then be reclosed and the access switches opened. With this method only switches A and B are used to make and break the current paths. The access switches are merely used to route the current pulse to the proper drive line.

2.2.2 *Switches A and B*

Switches A and B are identical circuits. They are both normally closed switches utilizing a pair of 20D transistors in the output stage and

operated in an antisaturation circuit to provide fast turn-off by elimination of the storage time inherent in saturated transistors. The basic circuit is shown in Fig. 10. The input stage is compatible with LLL and is driven from timing chain flip-flop circuits used as gate generators. The second stage is connected across the base-emitter terminals of the Darlington-connected output stages. This connection allows very fast turn-off even when driving inductive loads. The voltage transient that occurs as the switch is opened drives the second stage harder into saturation, preventing the power stages from turning on again.

The paralleling of the output stage is necessary in this circuit to keep the junction temperature below 100°C in the worst case, since these switches work in a nonsaturating mode and may run at 100 per cent duty factor.

2.2.3 *Limiter*

The basic limiter circuit is shown in Fig. 11. The limiter is a floating limit circuit. That is, both ends are free while the circuit is in a limiting mode and the capacitor, C_1, controls the limiting voltage. In its quiescent state, the −40-v regulator controls the voltage across C_1. The transistor used is a 20D with a resistor in its collector to lower the transistor power dissipation when the peak current of 1.35 amps is being limited. The limiter is expected to limit the voltage across its corresponding switch and carry a 0.5-μsec triangular current pulse every 5.5 μsec. A pulse transformer is provided in the collector circuit of the limiter. It produces

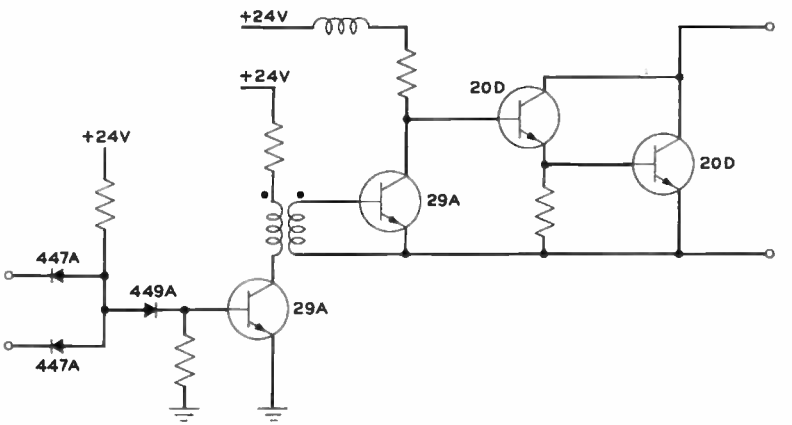


Fig. 10 — Circuit schematic — switches A and B.

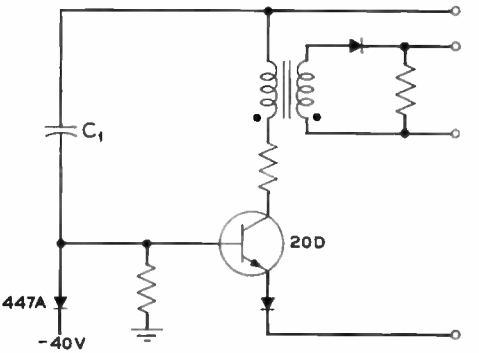


Fig. 11 — Circuit schematic — limiter.

a pulse used for maintenance checks on the access circuitry. The limiters for one axis are mounted on a common plug-in board.

2.2.4 *Access Switches, Voltage Pulser, and the Diode Matrix*

The access switch is a normally open switch used to select a current path through the diode matrix. The basic circuit for the access switch is shown in Fig. 12. The input stage is basically an LLL stage with 5 inputs. Four of the inputs are from the address register and provide the 1-of-16 selections required. The fifth is a gating input that controls the closing and opening times of the selected access switch. The collector of the input stage contains a 3-to-1 stepdown pulse transformer to drive the base of the output stage. This transformer coupling allows the output stage to float relative to the input stage. The transformer is designed to cause a 20 per cent droop in the current pulse applied to the output stage. This causes a negative drive to the base of the output stage to improve its turn-off characteristics when the input stage is turned off. The connections of the upper access switches for one axis are shown in Fig. 13. The capacitors (C_s) shown on the emitters of access switches represent the stray wiring capacitance of approximately 3000 $\mu\mu$f. The capacitance is charged to negative battery voltage to back-bias the diode matrix. Thus, when an upper axis switch is selected, it must discharge the 3000 $\mu\mu$f capacitance. To prevent this discharge current from exceeding the switch ratings, a filter is provided to limit the current surge to 1.35 amps at 50 volts. This allows the capacitance to be discharged in about 0.25 μsec. For this reason, the upper access switches are turned on 0.25 μsec before the lower access switches and switch A, which initiates the main drive pulse. Conversely, switch B is opened

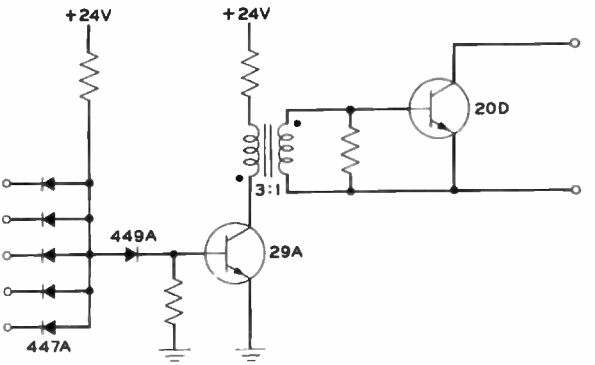


Fig. 12 — Circuit schematic — access switch.

0.25 μsec before switch A is closed, which forces the selected path to recharge its capacitance back to the level set by limiter B. The capacitance is charged the rest of the way to battery voltage by the action of the voltage pulser. The voltage pulser is identical to switches A and B; it is opened just prior to the initiation of a drive pulse and closed after the access switches have all opened. The voltage pulser connects the emitter side of each upper axis switch (and its associated capacitance) through a diode to battery.

The voltage pulser contains, as do switches A and B, a pulse trans-

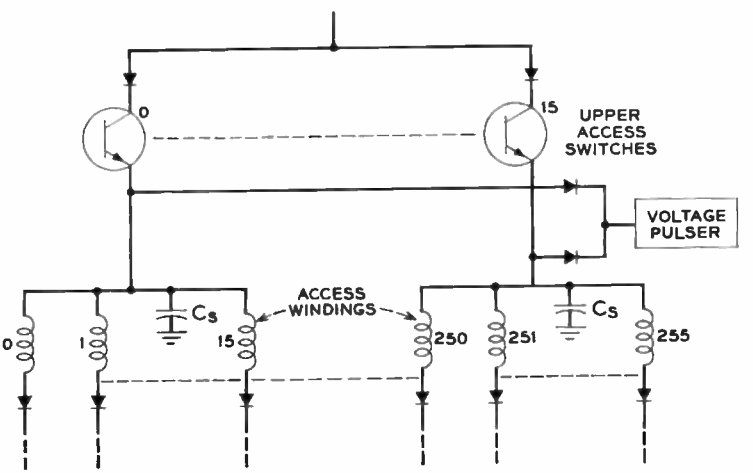


Fig. 13 — Partial schematic — access matrix showing stray capacitance load handling.

former to provide a voltage pulse representative of the switched current. This is used for diagnostic and maintenance purposes.

2.2.5 *Bias and Drive Current Regulator*

The bias and drive currents are regulated by identical plug-in current regulators. The currents at which the regulator operates, 1.31 amps for drive and 2.62 amps for bias, are determined by the socket interconnections. Because of the large amounts of power that must be handled, the regulator output stage is operated as a high-speed switch rather than in a linear mode. The average output current is controlled by adjusting the percentage dwell time of the switch closure. In this mode of operation the 20D transistors are either cut off or in saturation except for the short time required to transfer from one state to the other. Thus the power dissipation of the transistors is a small fraction of the power that is controlled.

The operation of the regulator can be seen from Fig. 14. The transistor switch and the resistor, R_s, are connected in parallel. This combination is then connected in series with the reference resistor R_R and the choke L. Feedback control is used to control the "on" and "off" times so as to generate the desired dc average current. (The resistance of R_s must be chosen large enough so that the highest steady-state voltage across it causes less than the desired regulator output current to flow into L.) Thus, the regulator's output current has an ac component; however, this component can be made quite small. The amplitude of the ripple component of the regulator's output current is controlled by the reference resistor value, the difference amplifier gain, and the Schmitt trigger's "window" or difference between "on" and "off" trigger voltage. The ripple frequency depends primarily on the above items plus the value of L_F but is also a function of the average battery and load voltage. The

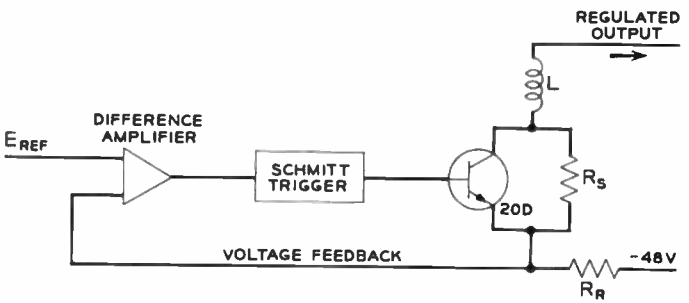


Fig. 14 — Access current regulator.

regulators operate at a nominal frequency of 50 kc with less than 1 per cent peak ripple. The choke serves to maintain constant current during transients too rapid for the regulator to handle directly.

Since the actual drive seen by a biased core switch is the sum of the X and Y drive currents minus the bias current, it is important that the bias and drive regulators do not drift in opposing directions and thus cause additive errors in the net drive. To prevent this occurrence, tracking circuits are provided that cause the output of each bias regulator to equal the sum of the X and Y drive regulator outputs within 1 per cent, ripple not included. The tracking circuit is composed of magnetic amplifiers driven with a 1-kc carrier current. A fail-safe tracking check detector is provided for diagnostic checking of the tracking circuitry.

2.3 *Access Timing*

The more important timing functions for the access portion of the program store are shown in Fig. 15. The various timing gates are generated by setting and resetting various flip-flops under the control of the access timing chain. Also included in Fig. 15 are some of the currents that are generated by these timing functions.

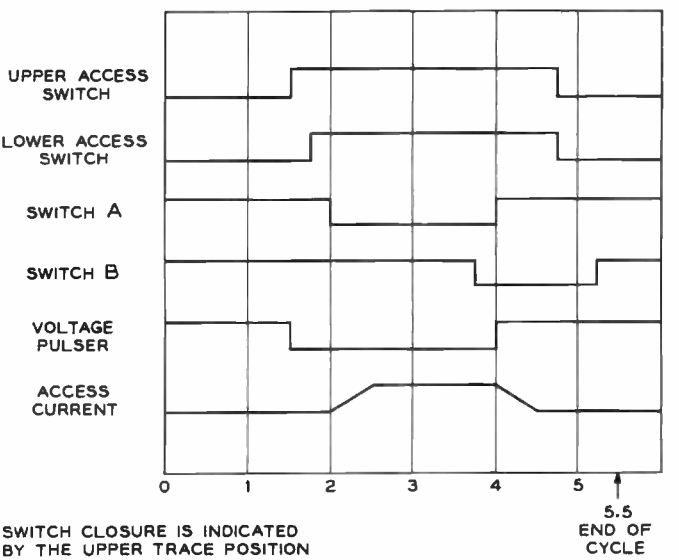


Fig. 15 — Access timing for one axis.

III. STORE READOUT

3.1 *Signal Selection*

As previously mentioned, in each group of four modules corresponding readout wires are paralleled. This results in eight groups of forty-four wires, of which only one contains the desired information. The other groups contain unwanted information or noise which must be suppressed. The selection of the desired group takes place in the low level selectors (LLS).

The LLS (see Fig. 16) accepts two twistor pairs, both associated with the same bit of the words common to an access solenoid. The two inputs are amplified and one is selected. The selection is made as early in the store cycle as possible so that selection transients will not interfere with the readout signals. The design of the LLS is such as to avoid in-service gain adjustment.

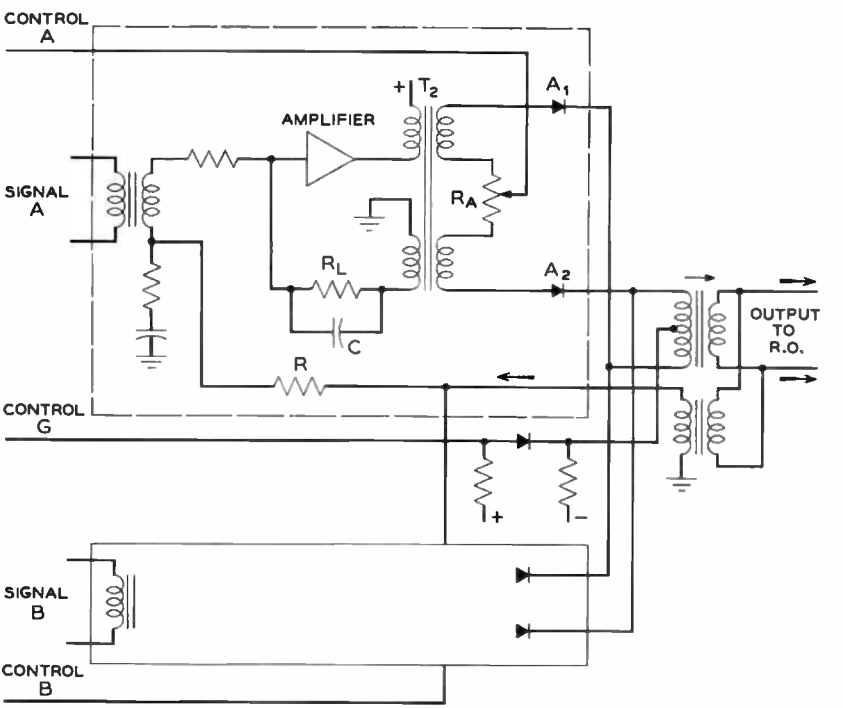


Fig. 16 — Circuit schematic — low-level selector.

The input signal is transformer-coupled to give good common mode rejection and to improve the impedance match. There are two stages of amplification with two feedback paths. There is a local feedback path, R_LC, and an over-all feedback path, R. The local path controls the high-end frequency response and gives the amplifier a low output impedance working into T_2. The over-all feedback path controls over-all gain and improves the low-frequency response.

Selection of signal A is made by making control A positive, control B ground, and G ground. With control A positive, equal currents flow through diodes A_1 and A_2. To avoid the requirement for matched diode pairs, R_A is provided to equalize currents. Since the currents are equal, only a minimum transient occurs when a path is selected. The signal at diodes B_1 and B_2 is sufficiently small that the diode threshold blocks the signal B.

Over a store cycle, the signal from the twistor memory goes both positive and negative and averages to zero. To avoid generating a dc component in the LLS, the path selection cannot be changed until the complete signal has passed. To accomplish this, a separate buffer register holds the selection information from cycle to cycle.

The LLS has a very large amount of feedback. In addition, the feedback resistors are very precise and have excellent long-term stability and tracking. The combination assures gain stability of a few per cent over the life of the circuit.

3.2 *Signal Amplification and Sampling*

Each of the 44 readout amplifiers shown in Fig. 17 accepts the outputs from four low-level selectors and provides the necessary gain to operate the sampler. The four LLS outputs are mixed by four resistors, R_1 to R_4, operating into the emitter of Q_1 and the feedback resistor, R_F. The mixing point is a very low impedance compared to the 100 ohms of R_1 to R_4. The readout amplifier has a large amount of feedback which provides frequency shaping and gain stability. Its long-term gain is constant to approximately one per cent. The signal at the output of the readout amplifier is bipolar and has a nominal amplitude of one-half volt for a one. A typical readout for several bits is shown in Fig. 18. Because of the wiring pattern, one-half of the ones have an initial positive loop, while the other half have an initial negative loop. Typical delta noise zeros shown are representative of a severe pattern interaction. The polarity of the zero is information-dependent and may be of either polarity at a given address.

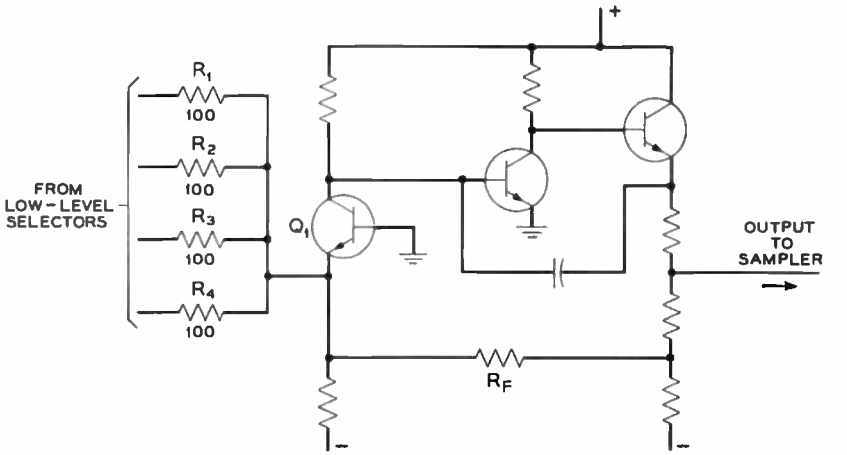


Fig. 17 — Ci rcuit schematic — readout amplifier.

A sampler (see Fig. 19) must make the decision as to whether the output is a one or a zero and quantize the output.

In designing the sampler certain characteristics of the signal were exploited. One of these characteristics is that the delta noise passes through zero near the time the one signal reaches a peak. Advantage was taken of this characteristic by placing a very narrow strobe at the delta noise zero crossing, as shown in Fig. 18. Another characteristic of the signal is that at a given address the one is of a known polarity. The sampler is designed to observe only this polarity, so that delta noise of the opposite polarity, no matter how large, cannot generate an output.

The two-stage input amplifier provides gain and a very low output impedance. The low output impedance is necessary because the load is variable, depending on the size of the signal. There is no load on the amplifier until the strobe occurs and no load then unless the signal exceeds the threshold. The low output impedance of the amplifier makes it possible to deliver enough power to trigger the output if the signal only slightly exceeds the threshold. This leads to high stability. The low impedance also prevents modulation of the signal by the sample pulse.

The strobe pulse, correct for the polarity of "one" being sampled, is supplied to each sampler at the correct time. The amplitude of this pulse is not critical. If the signal exceeds the threshold reference, diode D_1 is back-biased and the voltage change is coupled by C_1 to the output transistor, Q_1. Since Q_1 is biased in a Class A region, any minute voltage change is amplified and fed back to the base. This feedback path is

activated 0.25 μsec before the narrow strobe pulse. The purpose of this feedback path is to guarantee full quantization of "one" outputs and to lengthen the pulse to the standard half microsecond used for bus communication. Once regeneration starts it can be terminated only by opening the feedback path. The feedback path cannot cause regeneration in the absence of an input pulse.

3.3 *Strobe Timing and Generation*

In the laboratory, optimum strobe time can easily be set for a particular set of circuits. It is more difficult to design circuitry which will produce the strobe at the correct time for any combination of access or readout circuitry. To accomplish this, the strobe timing must be referenced as closely as possible to the start of the access current flow. Fig. 20

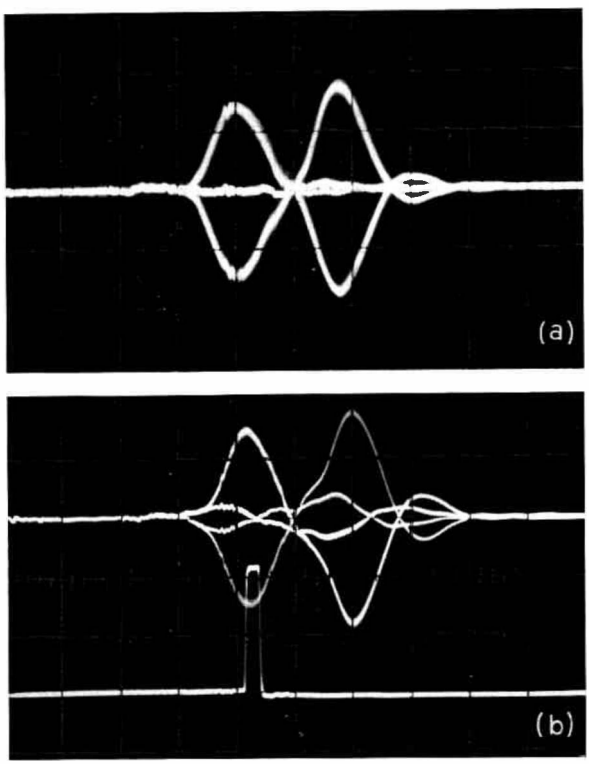


Fig. 18 — Readout waveforms: (a) typical readout; (b) readout with high "delta" noise.

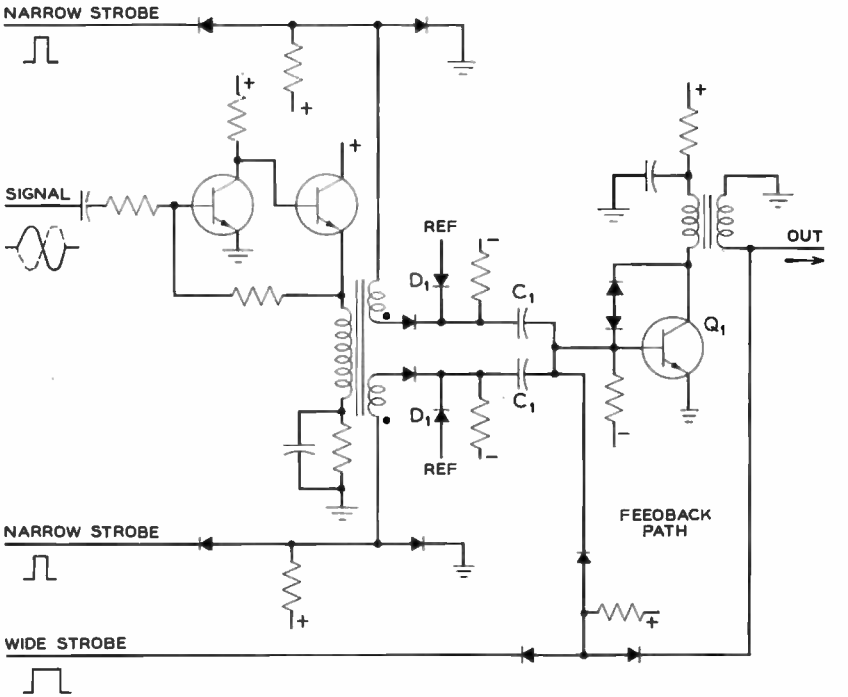


Fig. 19 — Circuit schematic — sampler.

shows the strobe generating circuitry. Three pulses must be generated: a narrow strobe of 0.25 μsec duration, a wide strobe that starts 0.25 μsec before the narrow strobe and ends 0.25 μsec after the end of the narrow strobe, and a 0.5-μsec sync pulse starting with the narrow strobe.

The readout signal occurs at a fixed time after the start of the access current. The beginning of access current for both axes is monitored by the strobe generator; only when both X and Y currents have started is the strobe circuitry initiated. This avoids the delay time variability of the access circuitry. The initiating pulse is delayed the proper amounts to set flip-flops 2 and 3, which initiate the narrow and wide strobes respectively.

The strobe amplifiers are relatively powerful, since they must drive 44 samplers. The strobe amplifiers each have an auxiliary output; these are added logically to start the timing of the wide strobe termination. This approach assures that the width of the pulse output of the sampler is 0.5 μsec.

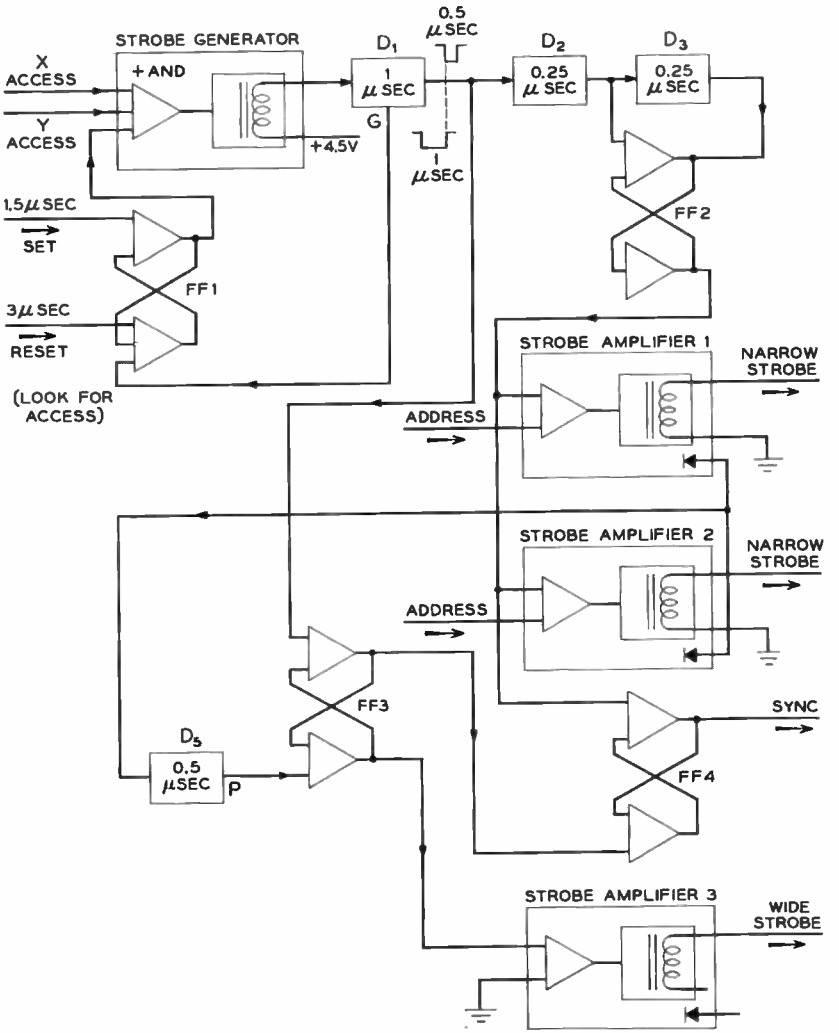


Fig. 20 — Strobe generation.

IV. CONTROL

4.1 *Communications*

The principal communication path of a program store is a duplicated two-way bus system connecting all program stores and both central controls. These buses carry the necessary addressing and control information required to read a particular word from a particular store or pair of stores, or perform a control function on a particular store. The buses also carry the reply from the program store containing either the desired stored word or, in special control modes, the states of various flip-flops. The signals are sent as 0.5-μsec pulses on a balanced line for binary ones and no pulse for zeros. The busing arrangement is described in more detail elsewhere in this issue.[8]

The program store also receives inputs from the central pulse distributor, the signal distributor, and the master control center and sends outputs to the master scanner. The inputs from the central pulse distribution are direct connections to each store, as opposed to a bus, and consist of either positive or negative 0.5-μsec pulses. These pulses must be accompanied by a pulse on a sync bus to be effective. They are used to set or reset various control flip-flops in the program store.

The inputs from the signal distributors are relay contact closures used to control special relay circuits or lights in a particular store. The inputs from the master control center are also dc contact closures. These inputs provide manual override control of the program store and can be used to force a particular store or stores off-line or to force a store to listen on a particular bus from central control.

The outputs from the program stores to the master scanners are divided into two groups. One group consists of direct wires from each store to a scanner ferrod and is used primarily to detect the status of various control relays and flip-flops in the store when they cannot be interrogated over the bus system. The second group is another common bus system to which any program store may be connected. This 48-pair monitor bus terminates in a bank of ferrods in the master scanner. When so ordered by the signal distributors, a program store will connect a group of test points to the monitor bus. This operation is used for maintenance checks of the voltage and current regulators and other dc levels throughout the store.

4.2 *Codes and Modes*

Each program store has its memory field separated into H and G halves. Each program store half will have a numerical code wired into

the store at the time of installation. This allows an office to operate with an odd number of stores and yet provides full duplication, as shown in Fig. 21. As can be seen, duplicate information (identified by the same numerical code) is always provided in the logically adjacent store. That is, information field 2 in the H side of store 2 is duplicated in the G side of store 1, and so forth. Normally, all H-half outputs will communicate on one bus to central control and the G-half outputs on the duplicate bus. Thus, when a particular information word is requested, the two stores containing the word will interrogate the required memory location and answer back on separate buses. If one of the program stores should fail, then the store carrying the duplicate information would be arranged to answer on both buses to provide duplication. The decision as to which bus or buses to answer on is made by system programs and sent to program stores as control orders.

Along with the address and code bits sent by the central controls on the program store buses are several mode bits which establish the type of operation desired of a store. Four modes of operation are decoded from these bits by the program stores. They are normal, maintenance H, maintenance G and control. If a normal mode is recognized, the stores (or store) that also recognize the code bits will send back the desired memory word. If a maintenance mode H is requested, only the store that recognizes the code bits as being the code of its H half will answer. Similarly, for a G mode, only the store that recognizes the code bits as being the code of its G half will answer. The control mode order affects only the store that recognizes the code bits as being the code of its H half. The control mode is not a memory read operation. The control mode is subdivided into two submodes, the control write and the control read. This selection is made by an additional wire pair in the bus system. In the control write mode, the address leads are used to write, dual rail, into various flip-flops within the store. Some of the address leads select which bank of flip-flops will be written; others contain the information to be written. In control read, the states of various flip-flops throughout the

Fig. 21 — Program store duplication.

store are sent back to the central controls. The address bits are used again to select which bank of flip-flops is to be interrogated. The control mode constitutes a powerful maintenance and diagnostic tool by providing a means of writing into and reading out of most of the control and maintenance flip-flops of the program stores.

4.3 *State Flip-Flops*

In order to control the store communications, several basic state flip-flops are provided. One group of such flip-flops controls the store routing. For example, one of the flip-flops selects the input bus on which the store should receive address and control information. This flip-flop is normally controlled by the central pulse distributor, but may be controlled by the master control center. Four flip-flops are used to control the normal store output to the buses. The H and G halves are each controlled by two of these flip-flops, which are normally set and reset by a control write operation.

In addition to the foregoing, two trouble flip-flops are provided that can prohibit the store from either sending or receiving on either or both of the buses. These can be set by the store itself, the central pulse distributor, or the master control center.

In addition, the office contains an emergency-action circuit which can set the state flip-flops. This circuit is activated whenever the system stops, and sets one of the stores containing program information to communicate on one of the buses and the store containing duplicate program information to communicate on the other bus. In offices with more than two stores, the remaining stores are disconnected from the buses. Following similar action with other units, the circuit attempts to restart the system.

4.4 *Timing*

The store timing is initiated by a sync pulse on the input bus from central control. The pulse is then delayed in each of two independent timing chains. One chain is used primarily for access timing, while the other is primarily used for control circuit timing. The two timing chains provide reliability in that critical communication control circuits are wired to both chains so that a failure of one timing chain does not cause a breakdown of communications and thus allows diagnostic programs to use the control mode feature to check the timing chains. Having two chains also makes it possible to check the accuracy of one timing chain against the other. The timing chains consist of a series string of delay gate

generators (DGG). The DGG's are available with delay times ranging from 0.25 μsec to 1800 μsec, all of which use the same printed wiring board but have a different timing capacitor. The DGG consists of two monostable multivibrator circuits connected as shown in Fig. 22. The first three transistors comprise one monostable circuit with a negative input signal used to trigger the circuit. This circuit generates an accurate gate pulse whose width is equal to the desired delay. The output of this circuit is applied to the final transistor, which turns on at the end of the gate pulse for approximately 0.5 μsec. Thus the output of a DGG is a negative-going pulse with a width of 0.5 μsec delayed by a fixed amount from the input negative pulse. The DGG has several diodes for temperature compensation and can be set to better the 0.1 per cent. The long term drift of the DGG is expected to be less than ± 5 per cent of the delay value. A DGG can drive up to five input loads which may either be LLL circuits or other DGG's.

The DGG's on the access timing chain are used primarily to set and reset gating flip-flops which are in turn used to control access timing gates such as required for the access switches or switch A.

The timing chains are checked against each other at several places along the chains, including the ends, to ensure agreement in the two chains. In addition, circuits are provided to check the output level of each DGG to ensure that it is normally in its high state and not loaded

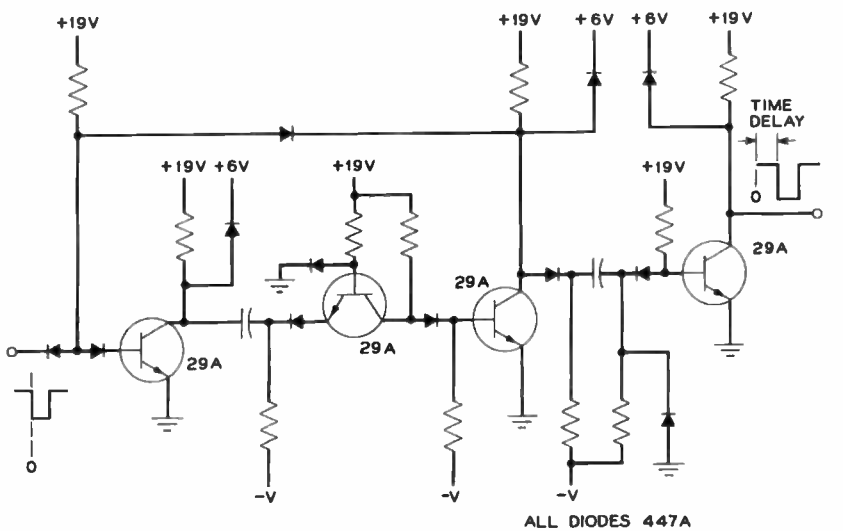


Fig. 22 — Circuit schematic — delay gate generator.

down by some other package with an input problem. Also, the operation of either timing chain can be inhibited to check the check circuits.

4.5 *Power*

The program store obtains its power from the +24-v and −48-v office batteries. This power is filtered and switched at the store. Separate power is provided at the store to power the card loader when the latter is connected to a store.

The program store draws essentially a constant load of almost 1000 watts from the battery plant. The actual battery drains are 18 amps on the +24-v bus and 12 amps on the −48-v bus. The power is filtered by a double L-section filter at the store input terminals both on the +24-v bus and the −48-v bus. The filters serve both to prevent noise spikes from entering the store and prevent power turn-on and -off transients from interfering with other operating units. The filter is so damped that the turn-on current increases smoothly to its operating value with no overshoot.

Several voltage regulators are used in the program store. A 4.5-volt regulator is used to supply the LLL stages as well as the low level selectors. Also +19- and −40-volt regulators are provided to supply several special circuit packs.

The +4.5-volt regulator is a high-efficiency "off-on" regulator which can deliver 1.5 amps at 4.5 volts while drawing about 0.4 amp from the +24-v bus. A basic diagram of the regulator is shown in Fig. 23. The output stage consists of a choke input filter which is either being charged by the 20D transistor connected to +24 volts or is being discharged by being forced to draw current from ground through the diodes. The "off-on" operation of the 20D is controlled by the output of the Schmitt

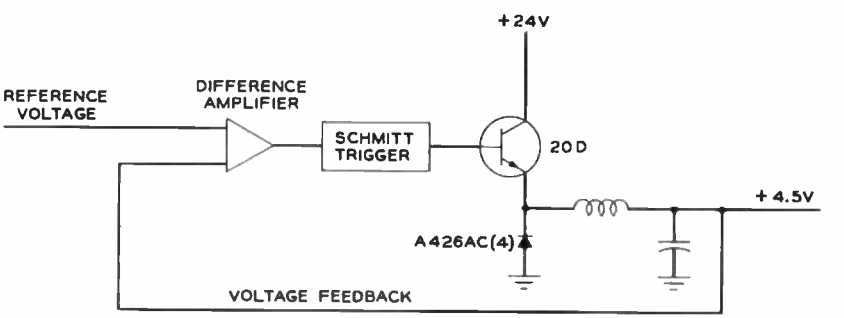


Fig. 23 — Circuit schematic — 4.5-volt regulator.

trigger, whose input is an amplification of the difference between a Zener reference stage and the regulators' output voltage. The feedback is so controlled that regulator "off-on" frequency is about 50 kc, with the 20D transistor being on about 20 per cent of the time. This type of regulator has the advantage of providing good regulation with high efficiency and low power dissipation in the semiconductors. The operation of this circuit is similar to the operation of the bias and drive current regulators described in Section 2.2.5.

The +19- and −40-volt regulators are conventional series linear control regulators. Since many of these regulators are used in each store, special reference regulators operating at +17 and −38 volts were made for the +19- and −40-volt regulators respectively. This circumvents the need of providing a Zener reference for every regulator package. Actually, two reference units for +17 and −38 volts are provided to facilitate maintenance and diagnostic checks.

The voltage regulators are checked for the correct voltages by means of the scanner monitor bus. The general method of testing the +19-volt regulators is indicated in Fig. 24. Since the regulators are cross checked against another regulator and reference unit, the failure pattern is unique for each regulator. To ensure that the check circuits are operating properly, the 19-volt check regulator outputs can be shifted by operating

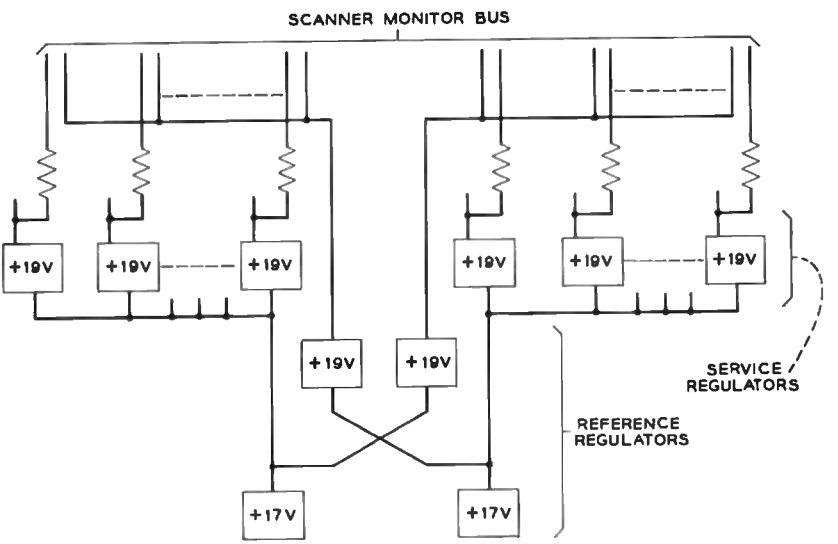


Fig. 24 — Regulator diagnostic checking.

a signal distributor input to give trouble indications of all associated regulators. The −40-volt regulators are tested in the same manner as the +19-volt units. The +4.5-volt regulators are checked two against one on the scanner bus, since only three are required per store.

V. MAINTENANCE

5.1 *Maintenance Philosophy*

The program store maintenance and diagnostic check circuits were designed along with the operational circuits. Each circuit, including the maintenance circuits, became the subject of a detailed study to ensure that a fault could be readily localized to, in most cases, a single circuit package, and in the worst case no more than three circuit packages. The effect of the failure of any semiconductor device used in the store was scrutinized to ascertain that the diagnostic checks provided could localize the fault. In general, the maintenance and diagnostic system can be divided into three sections. The first section is the all-seems-well (ASW) section. An ASW pulse is returned along with every readout from the program store to central control and indicates the success of all internal store checks. The second section makes use of the control read and write modes to diagnose and test the stores' condition. The third section is composed of the direct scan point that checks various states within the store and the monitor bus, which can be used to check many points within the store, as previously discussed.

5.2 *All-Seems-Well Circuit*

The ASW circuit provides an extremely powerful check on a store's condition. The ASW pulse indicates that: (1) the mode decoder has received a valid code, (2) the timing chains are both in step, (3) the address register and other control flip-flops were reset at the beginning of the cycle, (4) one and only one X and Y access drive winding was pulsed (for modes other than control), (5) the access pulses were of the proper amplitude, and (6) only one readout group was selected. Thus the ASW signal is derived from the status of several individual circuits. The failure of the central control to receive an ASW signal from a program store will generate a system attempt to reread the desired word. If the failure continues, the store will be checked by a diagnostic routine to isolate the faulty unit.

5.3 *Read Control and Write Control*

The control modes together with a circuit known as "freeze reset" comprise a powerful diagnostic tool for checking the program store.

When a control read (CR) or a control write (CW) is recognized by a program store, it knows the system wishes either to interrogate the state of a set of store flip-flops or to write into a set of flip-flops rather than to perform a memory operation. For example, the system can, in CW, write into a flip-flop that normally is set when an access failure occurs. This should result in an ASW signal failure if the ASW circuitry is functioning properly. The pulse detectors that check the amplitude of an access pulse can be forced to indicate either high or low as desired to check that these circuits are functional.

A CW command can also be used to place the freeze reset circuits into operation. In this type of operation, many of the flip-flops which are normally reset at the end of each system cycle (5.5 μsec) are blocked from being reset. Thus the address register, for example, can be filled and its contents checked by a CR, an operation which could not otherwise be checked. Also, the freeze reset operation can be used to localize the cause of an ASW failure. The store is placed in a freeze reset condition by a CW; then a maintenance operation is called for. This latter operation will cause the memory to be interrogated. If the ASW indicates a failure on this operation, a flip-flop for the defective circuit area will have been set and will not reset at the end of the cycle. Thus this flip-flop can be read with a CR command to localize the fault. The freeze reset condition is restored to normal by a CW command.

To localize a fault to a specific package it is often necessary to run a special sequence of instructions and determine the pattern of ASW failure indications. For example, an open diode in the group select flip-flops could cause two readout groups to be selected at the same time. The ASW circuit will indicate the program store failure, the control mode and maintenance mode operation will determine what section of the store is causing the failure (in this case the group select flip-flops), and then a simple four-step pattern will determine which of the four group selects are in trouble, since the ASW signal will indicate a failure except when the faulty package is selected.

5.4 *Direct Scan Points and the Monitor Bus*

The slowest maintenance access, but the most direct, is the use of scan points.

Each store has 17 direct scan points, most of which are used to deter-

mine the state of the more important control flip-flops such as the code, trouble and bus receive flip-flops.

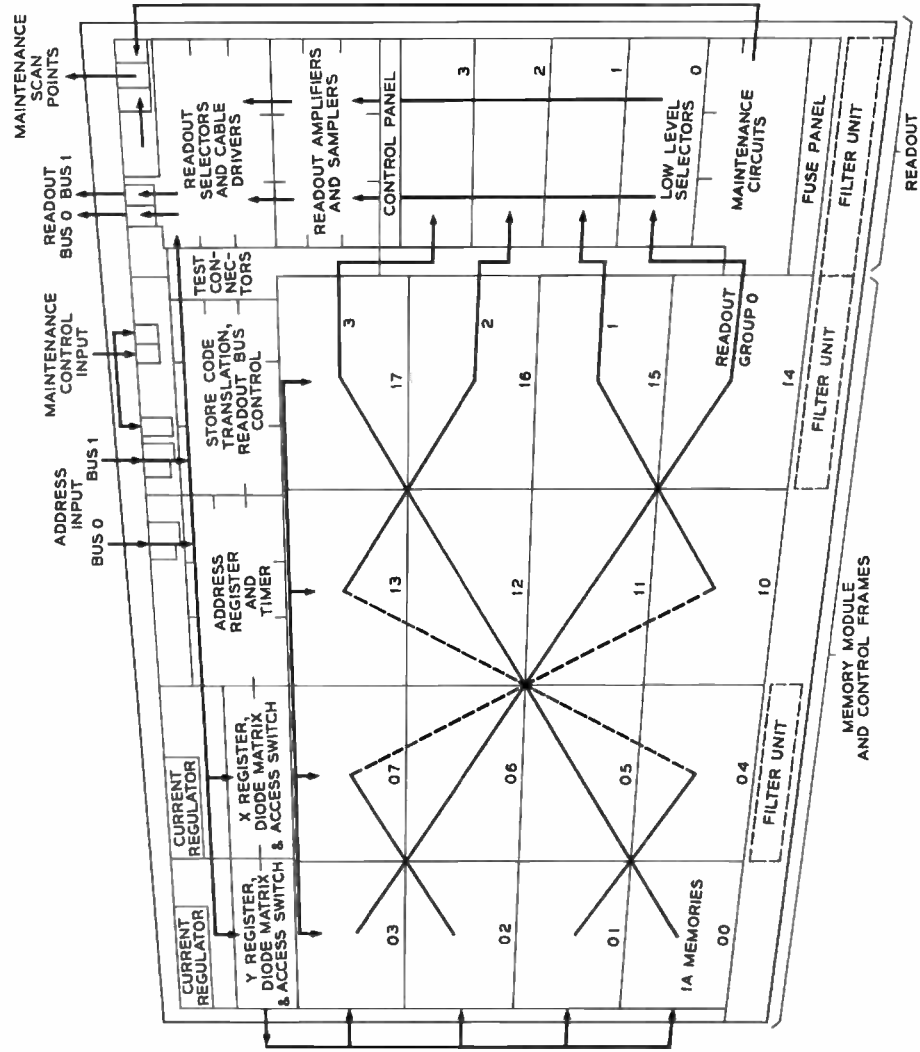
The direct scan points are also used to check the store manual control pushbuttons and power alarms as well as checking the states of relays controlled by the signal distributor, as discussed later. It is necessary to have direct scan points on functions such as the code detectors, since the store will not answer on the main communication path if it cannot recognize its codes, making the previously discussed diagnostic checks useless.

The scanner monitor bus is a 48-wire pair bus common to all the program stores within an office. The bus can be connected to points within a store by operating banks of relays under the control of the signal distributor. Each store has two banks of relays used to test the bus and connections to it, and three banks of relays to test various points within the store.

All of the store voltage and current regulators are checked by means of the monitor bus. The monitor bus is also used to check many flip-flops to aid in trouble diagnosis. In particular, the monitor bus is used to check the many packages whose failure could cause the store to become inoperative. The failure of a store to generate an access pulse in one axis could be caused by several different packages, most of which cannot be checked by a control read operation. The scanner monitor bus provides an inexpensive method of checking these packages.

VI. STORE EQUIPMENT[6]

The functional arrangement of the program store equipment is suggested by the flow chart, Fig. 25. Here the principal routes of information flow are superimposed on the equipment layout of the store. The address input signals are received by the pulse transformers in centrally located terminal strips at the top. These signals are then routed via the address register and timer unit to the X and Y drive circuits at the left and to the readout selectors at the right. The 256-point Y-select matrix reaches the select windings of the 16 memory modules with a vertical cable at the left end, while signals from the X-select matrix are distributed to the vertical select windings by means of a horizontal cable above the memory modules. These cables are shown in the rear view of the store in Fig. 26, together with the short horizontal jumpers which multiple the Y-select windings of the individual memory units and vertical jumpers which multiple the X-select windings. Thus, the desired 256×256 coordinate selection is achieved through short, direct wiring in a rectangular array.


MAINTENANCE SCAN POINTS
READOUT BUS 0 BUS 1
MAINTENANCE CONTROL INPUT
ADDRESS INPUT
BUS 1
BUS 0
READOUT SELECTORS AND CABLE DRIVERS
READOUT AMPLIFIERS AND SAMPLERS
CONTROL PANEL
LOW LEVEL SELECTORS
MAINTENANCE CIRCUITS
FUSE PANEL
FILTER UNIT
READOUT FRAME
TEST CONNECTORS
STORE CODE TRANSLATION, READOUT BUS CONTROL
ADDRESS REGISTER AND TIMER
CURRENT REGULATOR
X REGISTER, DIODE MATRIX & ACCESS SWITCH
Y REGISTER, DIODE MATRIX & ACCESS SWITCH
READOUT GROUP 0
FILTER UNIT
MEMORY MODULE AND CONTROL FRAMES
1A MEMORIES
00
01
02
03
04
05
06
07
10
11
12
13
14
15
16
17
0
1
2
3

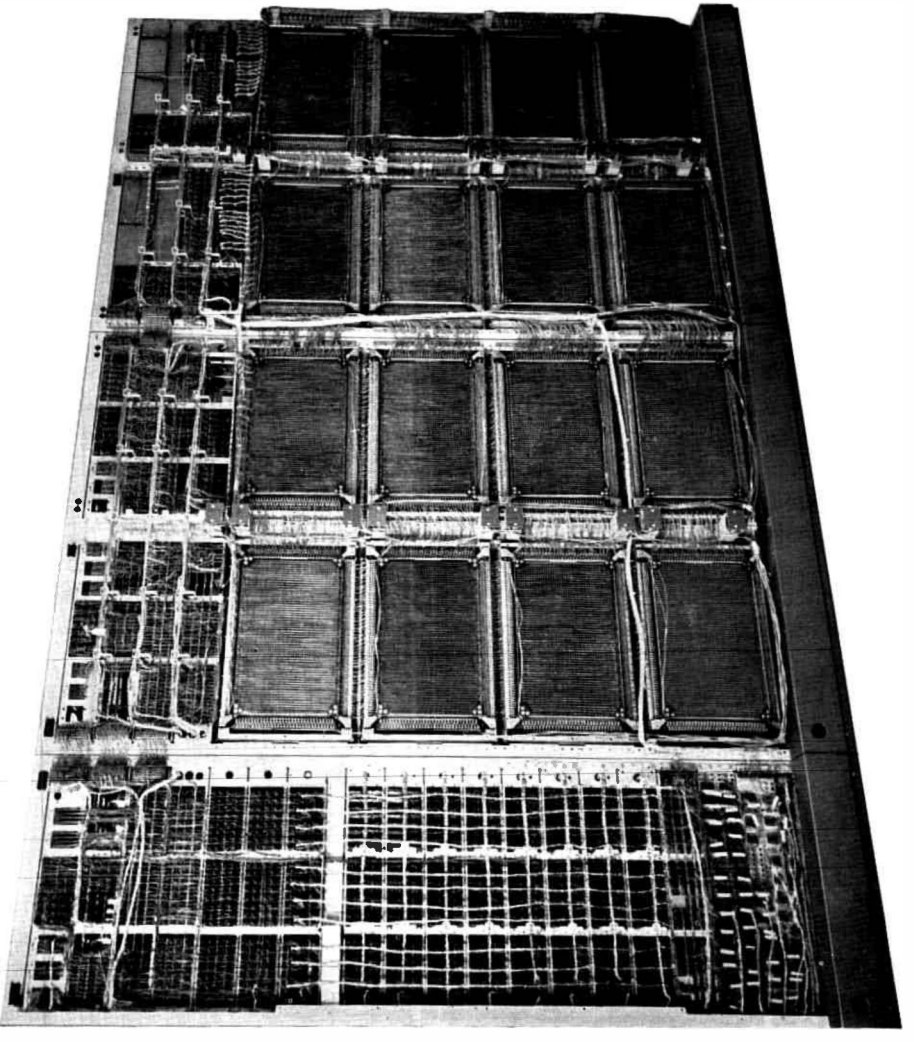


Fig. 26 — Program store — rear view with shield removed from readout bay.

For readout, the memory modules are arranged in four groups of four modules each. The twistor tapes for the four modules within each group are multipled in parallel. To minimize noise associated with half-selected addresses, no two memory units of a group are located in the same horizontal row or vertical column. The memory modules are associated as shown in Fig. 25, with readout connections to the four groups of low-level selectors at the right. These connections, as well as the interconnections between modules, are made with 26-gauge BY wire in close-twisted pairs ($\frac{1}{2}$-inch lay). These wires are precabled and routed through horizontal and vertical metal ducts between the memory units, and a vertical duct between the units and the low-level selectors. Special care has been exercised in the design of these cables and ducts to facilitate cable replacement, if necessary, in the event a memory unit should require replacement. The ends of the cables which terminate on the low-level selectors are distributed in the eight horizontal branches which appear in the left-hand bay of Fig. 26. The two branches associated with each readout group reach 44 circuit packs, which receive the 44-bit readout signals from each of the two twistor tapes associated with a readout group.

After selection, the readout signals are carried upward on the 11 vertical cables (four bits per cable) to the readout amplifiers and samplers, to the cable drivers and thence to the readout terminal strips at the top. The quality of the readout is due in considerable measure to the orderly and functional organization of this equipment.

VII. CARD WRITING

7.1 *Objectives*

The card writer is provided to rewrite, as necessary, magnet cards containing translation information for the office. It must be economical and reliable. It must not burden the real-time capabilities of the office while writing cards or require a high degree of operator skill. The procedure followed in writing cards should not destroy an existing module of cards until the new set is written and verified. Because there is no requirement of 100 per cent up time, duplication of communications, power, and equipment is not required.

The card writer must keep up with the writing requirements of a large office. Such an office, containing 6 stores, may require that $4\frac{1}{2}$ stores be rewritten each week. The card writer is designed to accomplish this in less than 18 hours of machine time.

The objective at the outset was to write the cards so well that no store margins were deteriorated due to marginal cards.

Every effort was made to design the writing process in such a way that no preprocessing such as bulk magnetization or demagnetization is needed and that the card will be correctly written regardless of the prior state. Accomplishment of this objective permits single-word writing, which is useful during program debugging.

7.2 *System Requirements*

The card writer magnetizes or demagnetizes the bits on a card by passing a head across the card. The system must provide the information to the head at the proper times. This requires suitable communication between the card writer and the system. The time between successive words is 13 milliseconds. This is slow enough that a scan point can be set by the card writer as soon as a word is written and the system can pick up the scan point on a routine scan and deliver a 44-bit word before the head reaches the next word.

Between each card writing operation approximately two seconds elapse while the system assembles the information for the next card. There is no check on the information written until the cards are inserted in the store. Every effort is made to assure a well written set and to complete the writing even when the system has difficulty delivering words on schedule. Some troubles, such as failure to receive a word when requested, cause the card writer to repeat the card. If three tries fail to write a card successfully, the process is aborted and an alarm sounded. Other troubles cause the process to be immediately aborted. In general, any indication of an incorrectly written module leads to termination during the card writing process. This follows the general philosophy of not putting suspect cards into operating stores even for checking purposes.

7.3 *Equipment*

The memory card writer, shown in Fig. 27 with a card loader attached, contains a centrally located card writing mechanism with circuitry above and below. The logic unit above is functionally arranged with terminal strips and sequence control circuits at the top and with registers and write control circuits between these and the control panel. A group of relays to control polarity of the writing heads and the writing head connectors are mounted just below the control panel on the sub-frame of the card writing unit. The lower part of this subframe supports a group

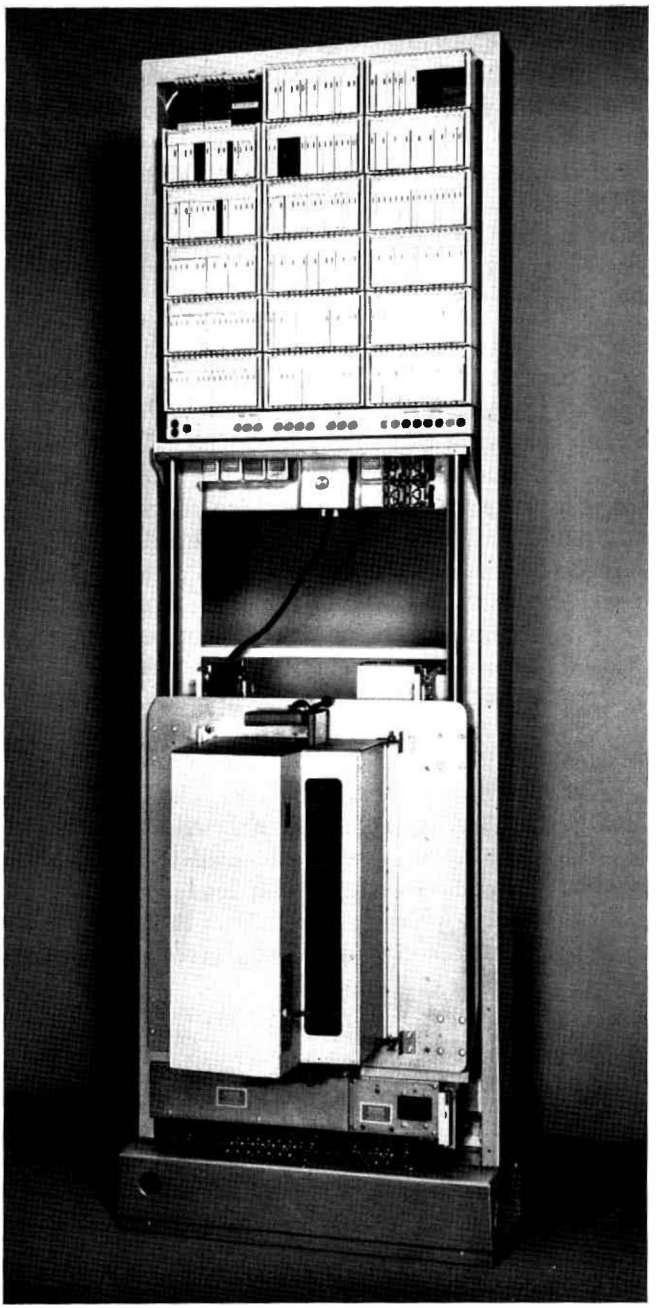


Fig. 27 — Memory card writer — front view with card loader attached.

of relays which control the mechanical sequences of the writing unit and power control circuits.

When a card loader contains a set of cards which are to be written, it is mounted on the memory card writer by clamping it in a vertical position to a carriage which forms a part of the card writing unit. This carriage is mounted on vertical rails at the sides which permit it and the attached card loader, to be indexed upward as successive cards are written. In Fig. 27 the carriage and loader are in a partially raised position.

At each index position, a pair of fingers on the card writing unit reaches forward, seizes the card in that location and draws it backward over a horizontal writing table to a fixed stop position. The card writing head then passes in a smooth, continuous motion across the card (Fig. 28). In this figure, a guard rail which prevents accidental interference between the fingers and the writing head has been removed for better visibility.

During its travel, which requires approximately one second, the head magnetizes the initializing magnets, senses the position of these along the length of the card, and magnetizes or demagnetizes each of the 2880 bit magnets. After passing the length of the card in a right-to-left direction (viewed as in Fig. 28), the head is returned to its normal position at the right, the fingers push the card back into the loader, the fingers disengage the card and return to an intermediate position, the loader indexes upward, and the cycle is repeated.

In the loader, the magnet sides of the cards face alternately downward and upward, corresponding to the alternately right and left directions which these cards face in the program store. Since only upward facing cards can be written, alternate cards are first written with the loader oriented as shown in Fig. 27. This sequence is called "pass A." When completed, the carriage and loader are automatically returned to the downward position and a buzzer sounds to alert the operator that pass A is complete. The loader is then inverted, end for end, reclamped to the carriage, and pass B is started, during which the remaining cards are written. The carriage is again automatically returned to the normal start position and the buzzer signals the operator that writing is complete.

The card writing unit and card writing head require moderately high precision in mechanical construction for satisfactory card writing. Sensing the initializing magnets to synchronize the writing function is done primarily to ease the mechanical tolerances for positioning the card in the longitudinal direction. The design incorporates fail-safe and interlock features to avoid or limit damage in the event of power failure or com-

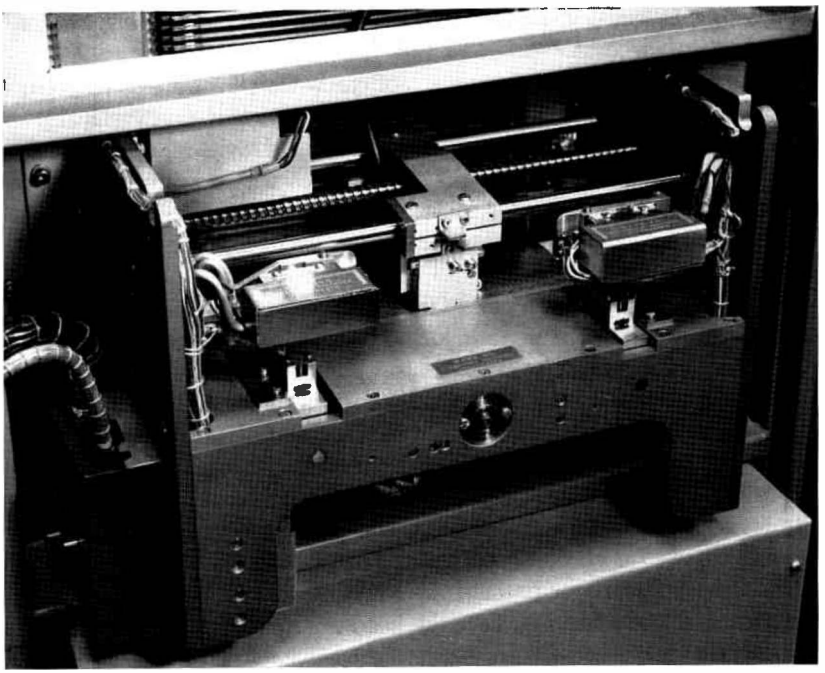


Fig. 28 — Memory card writer — rear view of writing mechanism.

ponent failure. The writing head and finger assemblies are designed for easy replacement in the field since they are more vulnerable to damage than some other parts because of repeated contact with the twistor cards. Also, the central mechanism portion of the writing unit, together with the writing head, have all electrical connections on a plug-in basis and can be removed from the frame for servicing or replacement by disconnecting two connector assemblies and removing four screws.

The card loader (see Fig. 29) is used to insert the twistor cards into and to withdraw them from the memory modules in the program store, to support the cards on the memory card writer during the writing operation, and to transport cards between the program store and the memory card writer. At the program store, the loader always inserts or withdraws all 128 cards of a set simultaneously, using a motor drive mechanism. The loader is controlled by means of three pushbuttons and a lever at the right, which is used to engage or release cards from the drive mechanism. In addition, an adjustable indicator assists the operator in keeping track of the particular store and module number with which he is working. Each memory module carries preadjusted brackets and

registration details which support the loader in accurate relationship with respect to the twistor planes in the memory. Moderately high precision is needed in control dimensions for the loader, memory module, and twistor cards for the loading and unloading operations to be carried out satisfactorily.

7.4 *Circuitry*

Successful card writing depends on accurate positioning information, which must be obtained from the initializing magnets. A head precedes the sensing head to magnetize these magnets. The initializing magnets are very long (130 mils) compared to the desired positioning accuracy (±5 mils). The position sensing head has a special shape which develops a large pulse just as the head is centered over a magnet. Fig. 30 shows the sensing head and the relationship of the two gaps to the ends of the magnet. Only when the head is centered does the flux change through the two coils at the same time in an aiding sense.

Since there is a double row of initializing magnets, two heads are used alternately. The voltage from each head is amplified and sensed in an

Fig. 29 — Card loader in use on program store.

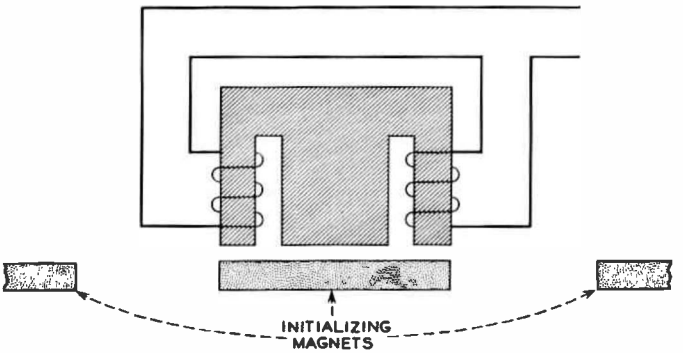


Fig. 30 — Position sensing head.

amplitude discriminator. To avoid false triggering, particularly during writing, the discriminators are enabled shortly before the proper head is at the expected position and are disabled as soon as writing is initiated.

Writing begins just as the write heads are approaching the center of a bit magnet. A relatively high frequency is used to magnetize or demagnetize the magnet. The basic waveform generated is a 20-kc sine wave which initially drives the heads into saturation and is gradually reduced in amplitude. The heads saturate at about 2 amperes of current at 26 volts peak amplitude. If it is desired to magnetize a bit, the waveform is altered during a saturated cycle just as the current crosses zero. The next current maximum is enhanced, and the current is forced to remain in the direction of that maximum. If the bit is to be demagnetized, the waveform is left on until the head is well off the magnet. The 20-kc waveform, Fig. 31, is developed by the circuit of Fig. 32. The head and two capacitors, C, form a resonant circuit which is in series with a Zener diode-capacitor network. The transistors are triggered "on" alternately for about 6 μsec, one each half cycle. The waveform reaches a peak in two to three cycles, at which time the head saturates. At this time the waveform is altered if the bit is to be magnetized, or made a zero.

To completely erase a bit magnet is difficult. Any biasing, either of unbalanced current through the head, or flux from an adjacent head, leaves a residue of flux in the magnet. Even a few milliamperes of unbalanced current can upset the erasure. It is difficult to obtain this degree of balance when the peak currents through the head are amperes. The Zener diode-capacitor network prevents slight differences in average voltages on the two sides of the head from causing unbalanced currents to flow but allows dc to flow through the head while magnetizing. During

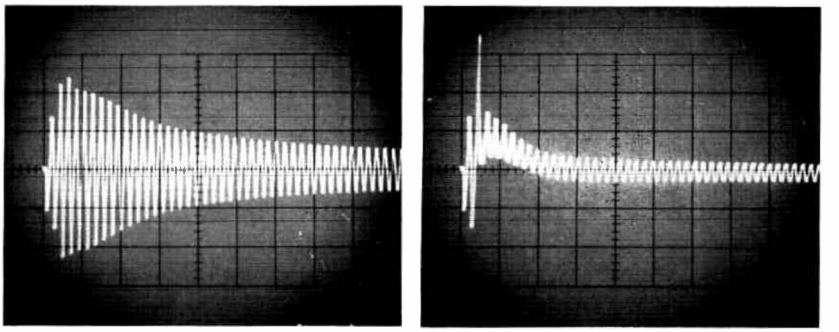


Fig. 31 — Basic demagnetize and magnetize waveforms.

the time the head is in saturation, exact balance is not maintained. Balance is recovered once the head is out of saturation, which occurs while the head is still over the bit magnet.

Biasing due to magnetizing adjacent magnets is avoided by essentially terminating the magnetizing currents during the early part of the demagnetizing period.

Additional circuitry is used for controlling the writing and mechanical sequencing of the unit. There are LLL circuits to check the count and

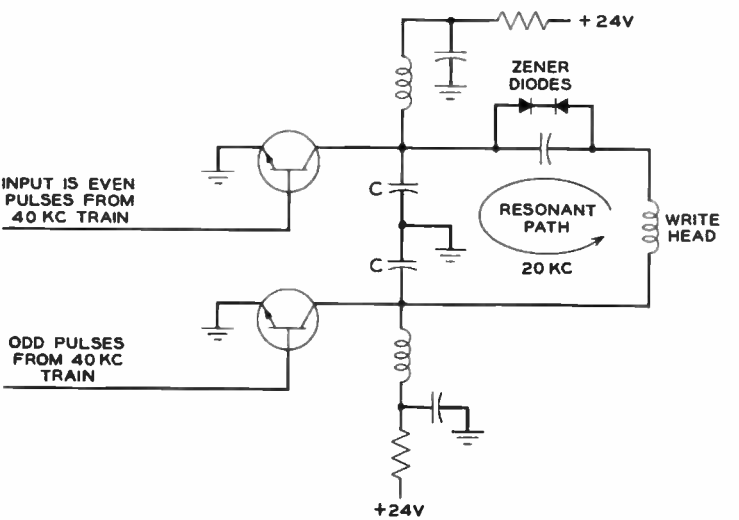


Fig. 32 — Circuit schematic — write head driver.

register the bits to be written. The control of the motors and solenoids is mainly by relay logic.

There are two drive motors, each 208-volts, 3-phase, 60-cycle. Both motors must be abruptly stopped, the head motor when at the home position and the finger motor when full forward and when full back. This braking is supplied by passing a direct current through two of the phases of the motors. The current is initially supplied by a large surge from a capacitor and is capable of stopping a 3600-rpm, fractional-horsepower motor in less than a revolution. The direct current is then allowed to decay to a value within the rating of the motor.

7.5 *Translation Changes*

Changing the translation information which is stored on the twistor memory cards is accomplished in the telephone office with the aid of two card loaders, the memory card writer frame, and a spare set of 128 twistor memory cards which is usually stored in one of the card loaders. The procedure normally followed in updating translation information is as follows:

(1) The first module to be rewritten is selected and an instruction is typed to the system identifying this module.

(2) A card loader containing the spare set of cards is mounted on the memory card writer in the pass A position and appropriate buttons are pressed to initiate the writing. On signal, the loader is inverted for pass B and the remaining cards in this set are written.

(3) The first store in which the module is to be updated is removed from service by means of the "request inhibit" button on the control panel.

(4) On signal that the store is out of service, the cards are removed from this module by means of the second card loader and are immediately replaced with the newly written cards in the first loader.

(5) The verify button is depressed, and upon a signal that the new cards are correctly written, the store is returned to normal service.

(6) Steps (2) to (5) are repeated in order to rewrite cards in the memory module containing the duplicate of this information.

A simplified diagram for changing information for a single set of cards in one program store is shown in Fig. 33. The machine time required for processing a set of 128 cards in the memory card writer is approximately 10 minutes, and time required by the card loader for inserting or withdrawing a set of cards from a memory module is approximately $\frac{1}{2}$ minute each. Including manual operations, a total time of about 12 minutes is required to update information in a single memory module. However,

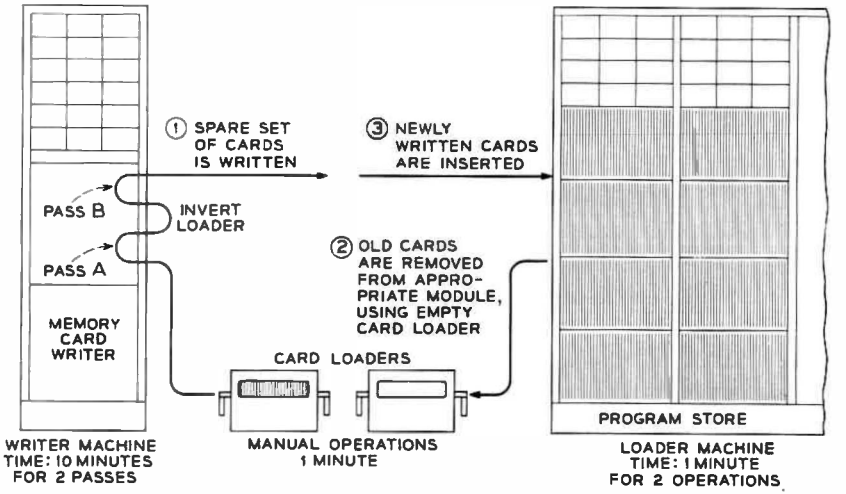


Fig. 33 — Card writing sequence.

it is seen that the down time during which the store being changed is unavailable to the system is only a little more than one minute per module.

In the event that a set of cards is incorrectly written, this is indicated after insertion into the store by the failure of the "verify" lamp on the control panel to light. In this case the new cards are immediately removed and replaced with the old set, and the store is returned to service.

7.6 *Program Changes*

In an operating office there is seldom an occasion for writing program information. However, during system debugging work this is the primary use of the card writer. To facilitate this operation a special laboratory machine called a tape reader assembler and processor (TRAP) was developed.

TRAP contains a conventional digital tape transport, a small core buffer memory, and a moderate amount of logic. Ordinarily, TRAP is supplied with a tape prepared on a commercial data processor containing the program information in the order the card writer will write it. TRAP reads the tape into the buffer and supplies a word at a time to the card writer.

An alternate mode of operation of TRAP permits the writing of a single word on a card. This is of considerable usefulness in making short program corrections.

The modifications to the card writer for TRAP operation involve the addition of several special circuit packs. These are not normally provided in an operating office.

VIII. PERFORMANCE

The program store is designed to operate over wide battery voltage limits (42 to 52 v) and ambient temperatures (32°F to 115°F) with any stored pattern of information. The evaluation of a unit of this size to assure it has met its design objectives is far from an incidental part of the development program. Ultimately the full evaluation requires operation with a complete ESS. To achieve a simulation of such operation a test set was developed to operate the store and analyze its outputs. The large number of variables involved in such an operation is clear from the control panel of the test set (Fig. 34).

Because the store has a read only memory, evaluation with different patterns of information is laborious. In lieu of actual worst-case patterns, a limited number of worst-case repetitive patterns is used. To ensure that under actual worst-case patterns the store will retain margin, the one-to-zero ratio must remain above 2.5 to 1 for the repetitive patterns. A very large number of tests of this type were carried out on a prototype store at varying conditions of temperature and worst tolerance circuits. These led to changes in both circuits and modules during development. Four stores of essentially final design have been operated for more than five store years in an actual system environment. As would be expected from the worst-case design approach, this operation has uncovered no case of errors not associated with failures of circuit or writing.

Circuit failures have been gratifyingly low despite the usual initial troubles encountered in first manufacture of a complex system. The average failure rate has been one package a week, but has shown suspicious bunching that pointed toward inexperienced debugging techniques. The mean time to failure has steadily risen during the interval.

Card writing performance has been generally excellent. The objective of erasing or writing to a point where no decrease in store margins could be assigned to card writing has been consistently realized. This required a ratio of 26-to-1 in magnet field strength between a zero and a one. Forty-to-one (2 gauss to 80 gauss) is generally obtained.

Error-free writing has not been as easy to obtain. A bit error rate exceeding 1 in 10^7 has been routinely achieved but at this rate to produce error-free modules requires rewriting one in ten. Most errors have been due to communication or mechanical problems. It is expected that an improvement of a factor of 10 will be achieved before the initial service

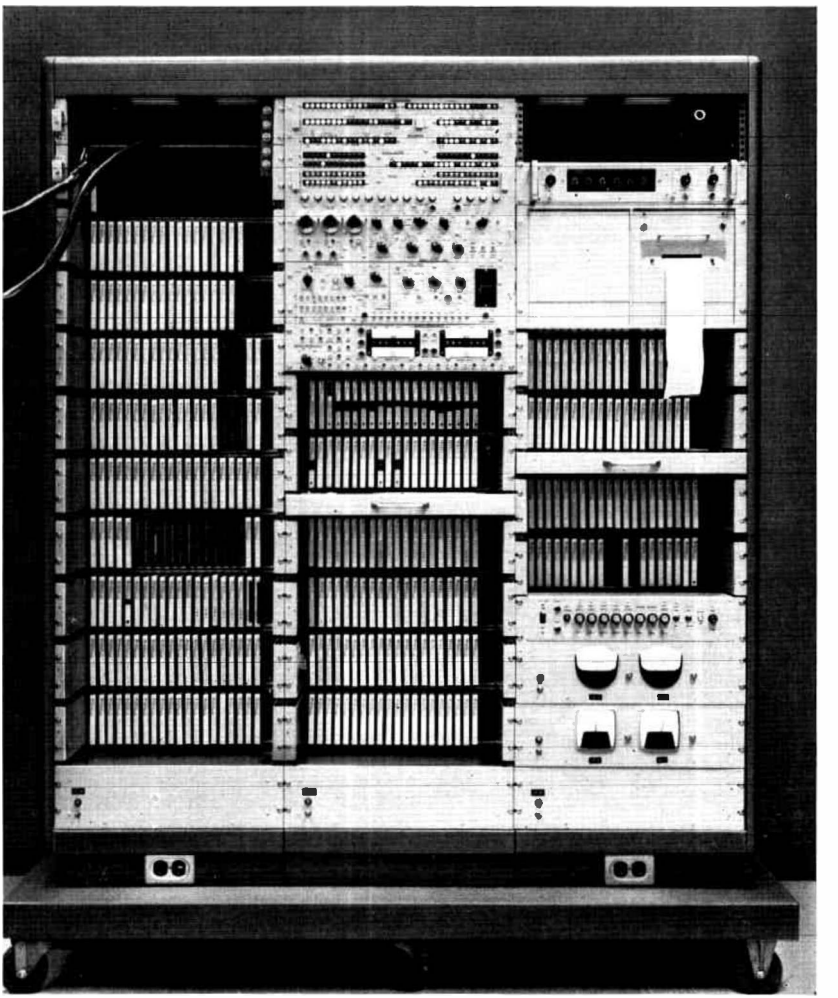


Fig. 34 — Program store test set.

date. This improvement is desired mainly to eliminate the nuisance effect of errors, since the system can tolerate a small number of errors in translation information through the recent change mechanism which allows any program store translation word to be superseded by a call store entry.

The program store has been evaluated against all of its initial objectives, and in all cases has met or exceeded them. A separate evaluation of the memory modules has been carried out and reported.[7]

IX. ACKNOWLEDGMENTS

The design of the program store has involved a large number of people, both directly and indirectly. It is not possible to acknowledge individually all of the many contributors but we are well aware that without them the development could not have been a success. The authors wish especially to acknowledge the excellence of the twistor modules developed under the direction of L. W. Stammerjohn and J. J. Suozzi.

REFERENCES

1. Downing, R. W., Nowak, J. S., and Tuomenoksa, L. S., No. 1 ESS Maintenance Plan, B.S.T.J., this issue, p. 1961.
2. Bobeck, A. H., A New Storage Element for Large Sized Memory Arrays — The Twistor, B.S.T.J., **36**, Nov., 1957, pp. 1319–1340.
3. Looney, D. H., A Twistor Matrix Memory for Semi-Permanent Information, Proc. Western Joint Computer Conf., March, 1959, pp. 36–41.
4. Gianola, U. F., Looney, D. H., Ruff, J. A., and Munn, A. J., Large-Capacity Card-Changeable Permanent Magnet Twistor Memory, *Symposium on Large-Capacity Memory Techniques for Computing Systems*, Macmillan Company, May, 1961, pp. 177–194.
5. De Buske, J. J., Janik, J., Jr., and Simons, B. H., A Card-Changeable Nondestructive Readout Twistor Store, Proc. Western Joint Computor Conf., March, 1959, pp. 41–46.
6. Ferguson, J. G., Grutzner, W. E., Koehler, D. C., Skinner, R. S., Skubiak, M. T., and Wetherell, D. H., No. 1 ESS Apparatus and Equipment, B.S.T.J., this issue, Part 2.
7. Stammerjohn, L. W., An Evaluation of the Performance of the Permanent Magnet Twistor Memory, Proc. Internatl. Conf. Nonlinear Magnetics, April, 1964.
8. Connell, J. B., Hussey, L. W., and Ketchledge, R. W., No. 1 ESS Bus System, B.S.T.J., this issue, p. 2021.

No. 1 ESS Call Store — A 0.2-Megabit Ferrite Sheet Memory

By R. M. GENKE, P. A. HARDING and R. E. STAEHLER

(Manuscript received January 10, 1964)

A call store is an 8,192-word, 24-bit per word memory system using multi-aperture ferrite sheets in a destructive-readout, coincident-current configuration and operating in a 5.5-microsecond read-write cycle. It is designed for 40-year life and operation in ambient temperatures from 32°F to 115°F.

The paper describes in detail the memory medium and memory circuits. General description is given of call store usage on the common bus in the No. 1 ESS and the maintenance and diagnostic features provided. Finally, a discussion of testing and test pattern philosophy is included, along with results measured on production stores.

I. INTRODUCTION

1.1 *System Requirements*

As the name of the call store implies, its prime function is to provide the read-write or erasable storage for all the call-related temporary information processed by the No. 1 electronic switching system (ESS).

Aside from the conventional call processing functions,[1] other common system functions involving administration, maintenance, AMA and TTY buffering and network simplification (by map and queue techniques) are also assigned to the call store. This concentration of system functions has imposed rather large capacity requirements on the store. Of course the total number of bits required is a function of both the number of lines in the office and the number of calls per busy hour. A range of 100,000 to 4,000,000 bits is required to cover the gamut of office sizes and calling rates envisioned for No. 1 ESS. To achieve a compatible solution, a specific store size is required to provide a modular building block with few stores in small offices and many in large offices.

Maintainability and dependability considerations imposed rather extensive requirements[2] on the call store. A duplication scheme to permit

economic growth by single stores requires splitting the memory in each store into two halves or two information blocks, with common access circuitry used for both. Each information block is then duplicated in another store.

Furthermore, high-speed and reliable communication between the call stores and the central controls, as well as the high-speed switching of duplicated information blocks, calls for the use of a bus system.[3] All the call stores in the office share the same set of signaling and addressing leads, with each call store on the bus capable of recognizing its own name code (sent as part of the address) and responding with the addressed word. Each store must communicate over either or both duplicated buses as directed by the central control. Furthermore, the central control must have the option of changing or rerouting these central control-call store bus assignments at system cycle speeds, thus bypassing faulty individual stores or buses to achieve an operational system.

Special maintenance control modes are required to permit use of the normal communication bus for the interrogation of various internal test points within the store and also the alteration of the operational features of various internal circuits. Separate monitor and direct scanning buses with access to additional internal points are also necessary to allow alternate channels of interrogation. The system requires verification of proper operation and transmission. This is accomplished by a parity check on each address received, as well as tests of a number of internal store circuits.

Finally, the call store must be designed against a 40-year life requirement with no program or environmental restrictions. These requirements dictate redundancy and safeguards against catastrophic failure, so that failures can be rectified by maintenance. Standardized plug-in circuit packages must be used to make failed components readily accessible and rapidly reparable.

II. DESIGN OBJECTIVES

2.1 *Memory Medium*

The basic memory medium selected for the call store is the multibit ferrite sheet with a 16 × 16 array of holes.[4,5] A copper conductor is evaporated and subsequently electroplated onto the sheet in a pattern forming the equivalent of a copper wire threading through all 256 holes in series.

The plated lead on the homogeneous sheet is a prime factor in achieving low memory module fabrication costs, since this eliminates the tedious

hand wiring operation that would be necessary to interconnect 256 discrete memory elements in an equivalent manner and lends itself to automated high-level manufacturing techniques. Also, the precise geometry of the ferrite sheet permits easy registration of holes in large stacks, and this simplifies and reduces the cost of the few hand wiring operations that remain.

The use of ferrite sheets also reduces the number of soldered connections over the use of discrete memory elements by a factor of 5 to 1. This is extremely significant in large-scale memories and enhances the over-all reliability and dependability.

To achieve these benefits in terms of cost and reliability, a considerable initial device development effort was required, which, based on present-day yields and performance, was well worth the effort.

2.2 *Store Characteristics*

The organization of a single call store is chosen to provide a capacity of 8192 words of 24 bits, or a total storage capacity of 196,608 bits. This appears to be a size that satisfies the minimum requirements of small offices and provides the growth flexibility for the larger offices. The 24-bit word length and the 5.5-μsec cycle are dictated by general system considerations and do not represent an attempt to realize the maximum capabilities of the memory. A median-size office of 10,000 lines having a calling rate of 13,000 calls per busy hour would require only two basic call stores. Of course, two additional units would be added for duplication purposes. Many smaller offices would require only one call store and one duplicate, while large offices would require up to 40 total stores.

The 768 ferrite sheets necessary to provide the 196,608 bits are wired for coincident-current access. A 13-digit binary address is necessary to control the access selection.

2.3 *Environment*

The call store must be operational over an ambient temperature range of 32°F to 115°F without recourse to environmental control.

2.4 *Simplified Block Diagram*

The simplified functional block diagram of the call store is shown in Fig. 1. There are four major blocks: the memory module, the memory circuits, the logic and sequence circuits, and the maintenance circuits.

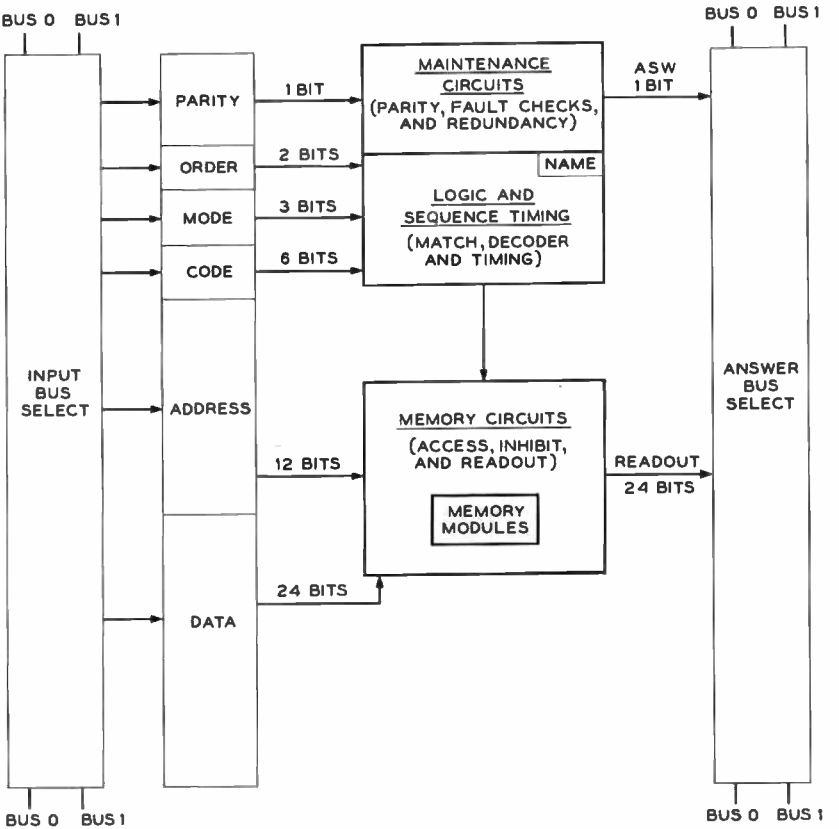


Fig. 1 — Simplified functional block diagram of call store.

2.4.1 *Memory Modules and Memory Circuits*

The ferrite sheet memory modules are surrounded by the memory circuits. A 12-bit memory location address directs the access circuits to switch the 24 "holes" desired in either of the two information blocks in the store. A single bit is derived from the name code to select one of the two information blocks. The readout signals from the memory are amplified and returned on the answer bus.

In the "write" mode the data register is set by the 24-bit data pulses received via the input bus.

In the "read" mode the data register is used for buffer storage of the readout, which is then rewritten into the memory.

2.4.2 *Logic and Sequence Circuits*

The logic and sequence control circuits decode the incoming 6-bit name code, the 2-bit read or write order, and the 3 mode bits to determine whether or not the call store should respond and in which mode of operation. Internal timing circuits are also included in this block.

2.4.3 *Maintenance Circuits*

The maintenance circuits check the correctness of the single parity bit generated over both the code name and address. These circuits also verify a number of internal store test points, and if these and the parity are all satisfactory an "all-seems-well" pulse is generated and sent on the answer bus.

Aside from the basic read and regenerate and erase and write modes, other modes are added to facilitate the maintenance of the store.

First, a "maintenance" mode is added in which the store responds as a memory but most of the flexibility in store selection and in store response is eliminated. Only one store can respond, and the store address is uniquely pinpointed by the incoming address.

A "control read" mode permits the interrogation of test points within the store. It accomplishes this by bringing the test data from the selected test points through the data register out on the answer bus.

A "control write" mode allows central control to alter the internal control states in the store. Special data and address codes are used to internally manipulate all the circuitry without actually switching the memory. Thus central control is able to electronically change the particular memory assignment or code name of a store.

The three major circuit blocks, along with the memory modules, will be discussed in detail in the following sections.

III. MEMORY

The basic storage device is a multiaperture ferrite sheet approximately 1 inch square and 30 mils thick. Each sheet contains a 16-by-16 square array of holes 25 mils in diameter, placed on 50-mil centers. The material used in this device is a mixed magnesium-manganese ferrite similar to those used in square-loop toroids but with a higher Curie point, which allows an extended temperature range. Square hysteresis loop characteristics of 1.9 oersteds coercive force, 1800 gauss saturation flux density and 0.97 squareness ratio are the governing parameters of the material. However, behavior of the material surrounding a hole is also influenced by

neighboring material and magnetic field distribution. Each hole, when excited by approximately 250-ma coincident-current pulses, acts as a square-loop memory core with the field constrained to the annular region immediately surrounding each hole. Interaction effects are reduced by generating opposing fields in neighboring holes. In this application the typical disturbed "one" output of a switched hole is 75 millivolts with 0.9 μsec switching time.

A printed copper conductor linking all the holes on a sheet is one of the four necessary conductors (see Fig. 2). The three remaining wire conductors are threaded after the ferrite sheets are mounted into stacks.

3.2 *Constraints of Large Coincident-Current Ferrite Sheet Memories*

Three of the four conductors combine to form a standard coincident-current access.[6] In this application there are 128 X coordinates and 64 Y coordinates whose product (8,192) equals the number of words in the memory. Simultaneous half-read currents, I_d, along one of the X and one of the Y coordinates, are sufficient to fully switch the selected holes of an address to the read or "zero" state but will not disturb any of the many holes half-selected by X current and Y current. Subsequent re-

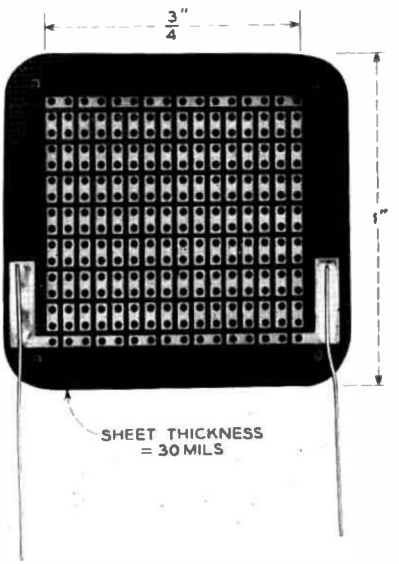


Fig. 2 — Ferrite sheet.

versal of current on the same conductors switches the previously selected address to a write or "one" state unless a separate read direction half-drive, called an "inhibit current," is applied on another conductor. For a 24-bit word there are 24 distinct bit planes, each containing an inhibit conductor and a sensing or read conductor; both are coupled through all the 8,192 address-location holes.

The major constraint in coincident-current memories is on the sense wire noise voltages due to the accumulation of signals across holes excited by the half-current drives. Attempts are made to cancel these signals by directing half the sense and drive lead hole intersections positively and the remaining half negatively. However, the positive and negative noise voltages, which are functions of information storage, past history and addressing sequences in other sections of the memory, do not cancel exactly, leaving a residue called "delta noise." A byproduct of delta noise is the large and variable load imposed upon the X, Y and inhibit drivers. The impedance to a driver, due to the summation of the inductances of each hole which is pulsed by the driver, is a function of the information stored and the remanent state of flux. These factors are, in turn, a function of the memory contents and the prior addressing sequences. The variable-load inductance, as well as the large interwinding capacitances, forms complex transmission lines that impose severe design problems on the drivers.

Methods of generating delta noise and driver loading variations are discussed in Section 9.2.

A unique problem to the ferrite sheet is the interbit reaction due to magnetic coupling of flux between the holes. To combat interhole coupling, special flux constraining wiring patterns are used which limit the fields to annular regions about each hole. (See Fig. 3.) The constraint is simply generated by guaranteeing that any hole selected with a full drive during read or write intervals is countered by an opposite half drive in the nearest neighbor holes. Therefore special wiring rules, covered in the next section, are applied.

The fixed 16-by-16 array of holes imposes another constraint, because this size must be used as the basic building block. But this larger-size sheet reduces the number of connections and improves reliability. The 24-bit per word application necessary in No. 1 ESS is achieved by coupling the sense wire through two adjacent rows for each bit; the 16 rows in a sheet can be subdivided into 8 row pairs which allow 8 bit planes. The full complement of 24 bits is formed by using 3 stacks of sheets, each stack supplying 8 rows of paired bit planes. This folding of the read or inhibit plane complicates X-plane wiring because two sep-

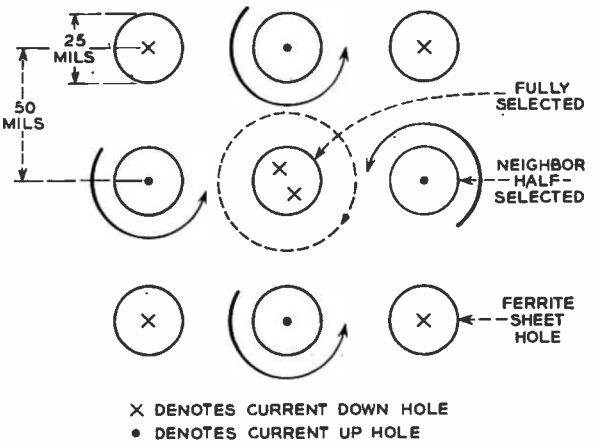


Fig. 3 — Flux locking.

arate X-plane conductors are necessary to guarantee that only 1 bit is switched in a bit plane and that all ferrite sheet holes are fully utilized.

3.3 *Basic Module — Size and Wiring Patterns*

The 8,192 words of a call store are obtained from 2,048-word modules which in turn are constructed from 512-word submodular assemblies. The modules are interconnected with all electrical connections on a plug-in basis, while the submodule assemblies are permanently mounted. Conductor length and possible increased manufacturing (yield) rejection are the major considerations which limit the maximum module size. The number of interconnecting terminals increases for smaller modules, with an accompanying reduction in reliability and an increased cost. A compromise between the yield or wiring length problems of a large module and the reliability or cost constraints of a small module has led to the 2048-word size.

The formation of the ferrite sheet module is dependent upon a number of noise-reducing techniques. An attempt is made to maintain all current carriers on a tight twisted-pair basis outside of a module and within the module, with geometric orthogonality between X, Y, and inhibit conductors. The positive and negative terminals of all conductors are arranged in close proximity. Since the read cannot be completely orthogonal with the X and inhibit, another basic requirement introduced complementary wiring with as much geometric symmetry as possible to realize balance and assist noise cancellation. Noise cancellation tech-

niques take on particular importance for the readout wire because of the large capacitive and magnetic coupling between drive and readout leads. Consequently, special noise cancellation wiring patterns are used.

A single ferrite sheet module is shown in Fig. 4. This unit with the four subassemblies, each containing three columns of sheets, is mounted with all holes in registration. The submodules are first fabricated with separate read conductors, each woven with special wiring patterns. These patterns (shown in Fig. 5) minimize coupling with the Y or inhibit drive lead by reversing the sense direction in a binary fashion. For example, the first two columns of read holes intersect the Y in the positive sense and the next two in a negative sense. For the next set of four holes the pattern is reversed. For the next set of eight holes the first eight-hole pattern is reversed, and so on. This scheme compensates for the variation

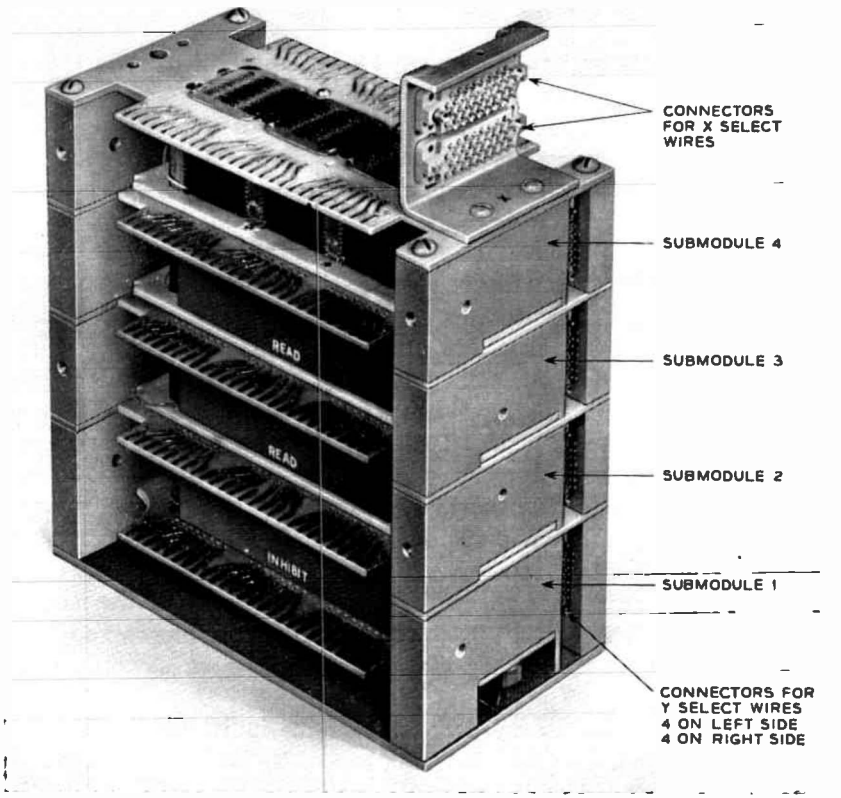


Fig. 4 — Module.

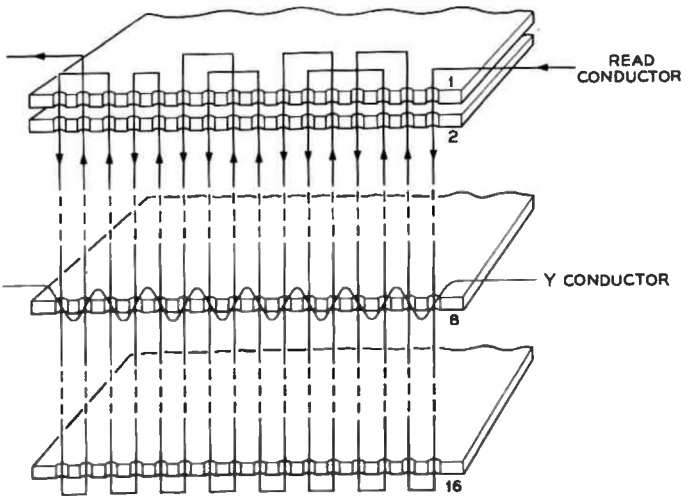


Fig. 5 — Readout and Y-axis wiring.

of current along the drive wire due to distributed capacitance losses (see Fig. 6). The shunt losses due to stray capacitance degrade the current entering on the left to that shown on the right. This difference causes different coupling from the drive to the readout wire and prevents proper noise cancellation. Simple inductive cancellation schemes which do not take this factor into account give less effective noise cancellation.

The separate 24 readout wires for each submodular assembly allow further noise reduction, as well as introducing flexibility of interconnections (for reasons to be described presently). Once the four submodules are wired they are mounted one above the other. The 24 inhibit wires are then inserted like the read wires, except that they pass up and down through all the holes of the four submodules in a simple up-and-down weave. The 32 X-select wires are threaded orthogonally to the inhibit, with 2 X-select wires per X plane, so that each X wire intersects a folded read wire only once on a sheet. Fig. 7 illustrates the X weave wired with a close-twisted pair arrangement both within and outside the module. The pattern also reflects the added requirement of coupling two adjacent holes which are not coupled by the same inhibit, to satisfy flux locking constraints.

The 64 Y coordinates are derived from the individual ferrite sheet plated conductors. The plated conductors of the same level sheets on each of the three stacks are soldered together to form a Y plane, with 16 Y planes per submodule and 64 Y planes per module. Although a

careful screening process on individual sheets is implemented prior to assembly, occasional defective sheets might occasionally appear in a finished module. One additional Y plane is included per submodule to allow a defective sheet to be replaced by a spare. This substitution is employed only in the manufacturing process.

To summarize the makeup of the completed module, there are four Y connectors on one side for entrance and four Y connectors on the other side for exit, each connected to a submodular assembly of 16 coordinates, each coordinate containing 3 sheets. The twenty-four twisted-pair readouts are brought out to four printed wiring boards, one at each submodule. One printed wiring board located at the bottom of a module provides access for 24 twisted-pair inhibits. Two connectors are located at the top of the module where the twisted-pair X-selected wires originate and terminate. Each module contains 64 Y coordinates, 32 X coordinates and 24 bit planes.

3.4 *Memory Organization*

Overcoming the prohibitive delta noise and the large, variable driver loads in an undivided or unduplicated memory, is a difficult goal for a

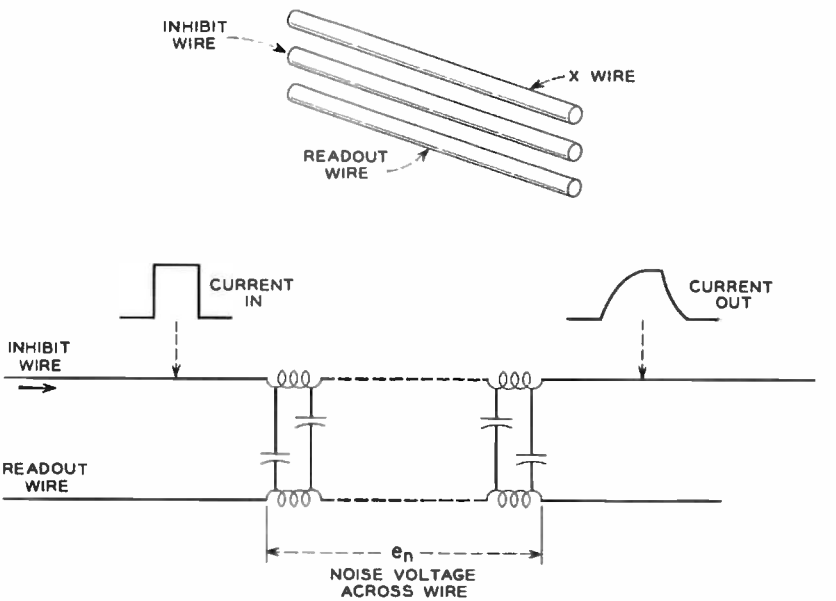


Fig. 6 — Inhibit transmission line.

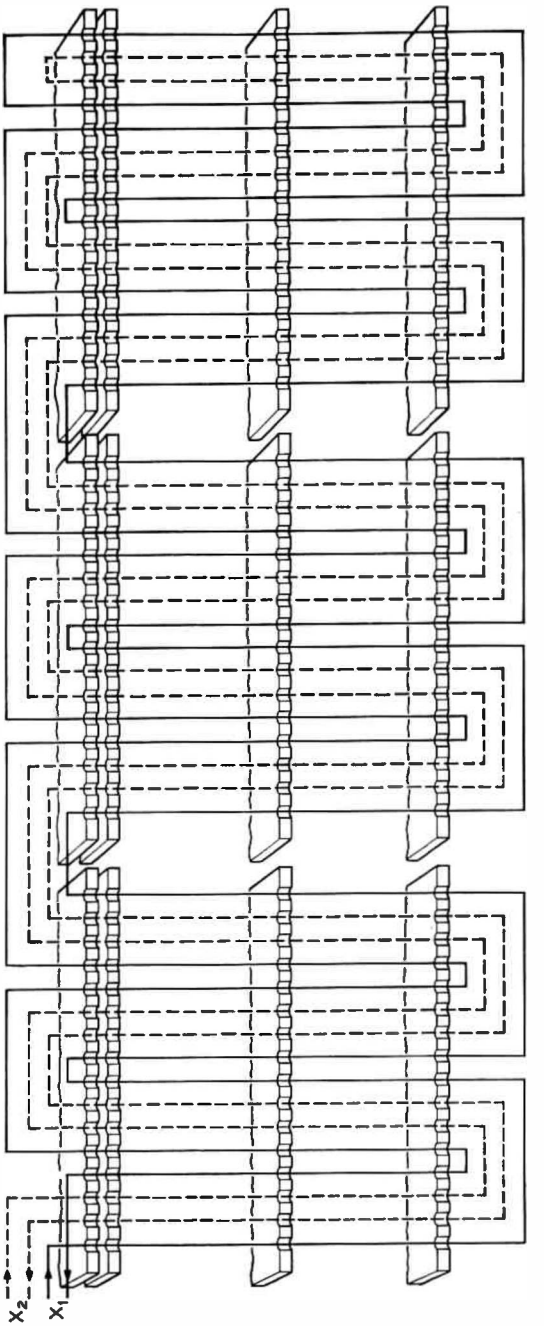


Fig. 7 — X-wiring module cutaway.

large memory designed for a 40-year life without potentiometer adjustments. Circuit schemes are introduced which drive large loads, stabilize currents, reduce delta noise via a post-write disturb, and shape current rise time. A novel scheme of measuring the noise with sample currents and subtracting the predicted noise during the read interval is also included. However, these appliques by themselves are insufficient to overcome the large delta noise of a full 8192-word coincident array with the extended life and reliability requirement of a telephone switching office.

Consequently, in this system the inhibit, read, and Y coordinates are subdivided and duplicated. The duplication is incorporated in such a manner as to minimize delta noise and reduce interactions between the coordinates, and cost reduction techniques are developed by utilizing common controllers as much as possible and placing the minimal hardware in the duplicate sections.

The interaction and delta noise minimization is illustrated by means of a Venn diagram (see Fig. 8). The matrix shown represents one of the 8192-word bit planes containing 64 horizontal Y lines and 128 vertical X lines. The matrix is split into 16 sections, each a 16-by-32 coordinate array with the upper 8 sections coupled to the A-read wire while the lower 8 are coupled to the duplicate B. The sections labeled α are driven by the α inhibit, while the remainder are driven by the duplicate β. The one-half length horizontal line, representing a Y current into the memory, shows the Y duplication, while the full-length vertical line represents an X current into the memory. The horizontal and vertical line segments indicate how the memory should be organized to minimize the coupling

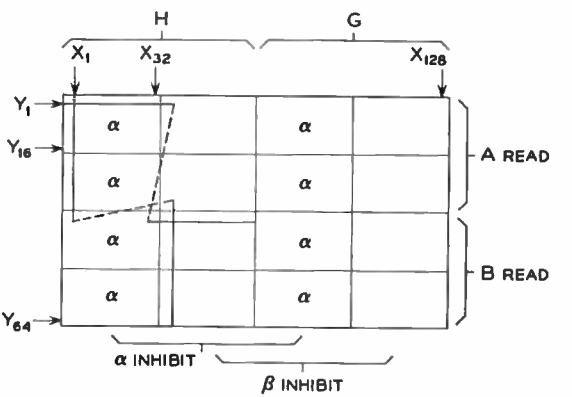


Fig. 8 — Bit plane, Venn diagram; minimum interaction and delta noise.

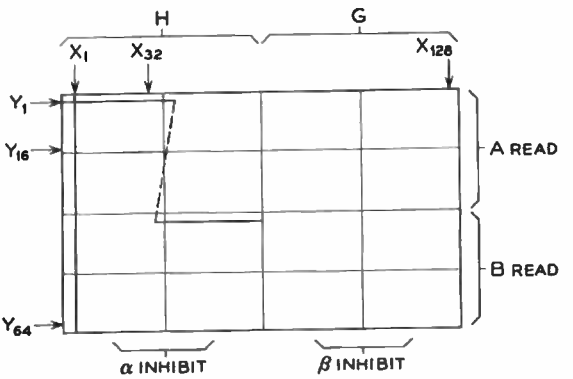


Fig. 9 — Bit plane, Venn diagram; compromise scheme.

of half-drive holes to the read wires and the coupling of inhibit currents to the X and Y wires. Since this organization is highly related to the module construction and fabrication, it is difficult to completely realize the objective; a compromise scheme results (see Fig. 9) which reduces terminal and solder connections.

Fig. 10 is a block diagram of the full 8192 words formed from 4 modules and shows the Y, inhibit, and read duplication. The first submodule read conductors are connected in tandem to the first, second, third and fourth modules. A jumper at the end cross-connects two adjacent read conductors to form a folded read plane. Each bit plane is split into two segments. One segment is derived by interconnecting the first and second submodules of all four modules, while the second segment is derived from the third and fourth submodules. Each segment is terminated by a separate preamplifier with 24 preamplifiers in the upper A half and 24 preamplifiers in the lower B half. The 2 preamplifiers per bit plane are recombined by an exclusive OR gate into a decision-level circuit.

The duplicated 64 Y coordinates are interconnected between modules so that Y currents through submodule 1 of module 0 will travel to submodule 3 of module 1. This type of cross connection reduces the Y-to-read noise by one-half. The Y-access source is an 8-by-8-by-2 three-dimensional matrix with current steering logic to decide if the G or H half is chosen. Through this technique the more expensive selecting switches and current generators are shared by both Y halves. Only the selecting diodes, some transformers, and two switches must be doubled.

The 128 X selections, with 32 X coordinates per module, are not duplicated. However, the noise coupling between the X and read has been cut in half because of the A and B read segments. For this access an

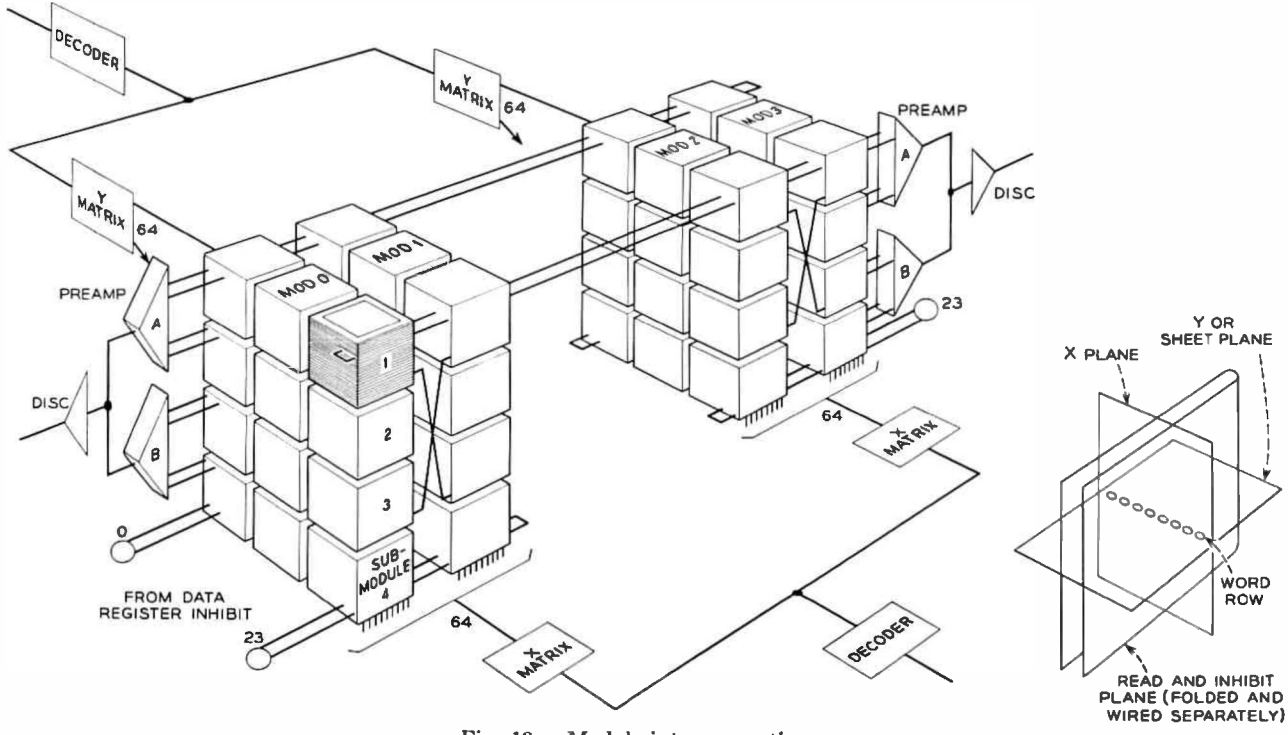


Fig. 10 — Module interconnections.

8-by-8-by-2 matrix, identical to the Y matrix, introduces significant cost reductions.

The 24 separate write back or inhibits also are duplicated because of the large inhibit wire inductance. One set of 24 inhibit drivers is coupled to the first two modules and the second set is coupled to the last two modules. This duplication is orthogonal to the read split to minimize the intersections between read and inhibit.

IV. MEMORY CIRCUITS

4.1 *Readout*

The readout complex is a circuit block which equalizes near- and far-end read signals to the preamplifier output. It contains two 3-stage high-gain transistor feedback preamplifiers with an EXCLUSIVE-OR gate for recombination of the read segments. DC restoration is used to reduce bias drifting caused by long bursts of unidirectional inputs, along with a scheme that measures the noise before sampling and subtracts the undesired noise from the output. The complex also accepts either positive or negative signals and samples or strobes the signal into a bidirectional gated-oscillator decision-level circuit (see Fig. 11).

Transmission line characteristics causing unequal response of read signals from near- and far-end points can create severe degradation of performance. To equalize the read response for both gain and phase, two transformers are used to couple both ends of the read wire, matching the characteristic impedance and balancing the transmission. The secondaries are connected in series to reconstruct the full source amplitude without attenuation.

The EXCLUSIVE-OR gate, dc restoration, and special noise measurement and subtraction functions are combined by a simple resistance-capacitance and bidirectional switch scheme (see Fig. 12). In this scheme the preamplifier capacitor output is shorted to ground by a clamp unless the preamplifier is chosen; if chosen, it is enabled only during the read interval. The small time constant formed by the low preamplifier output impedance and the low resistance in series with the capacitor allows quick discharge for dc restoration. After the clamp is open, the time constant is increased and the network transmits the preamplifier output with little attenuation. By opening the clamp during the early noise peak, the noise is measured and stored across the capacitor. During the subsequent sampling interval, the capacitor voltage subtracts from the input noise voltage, reducing the noise or undesired signal amplitude. The clamp is

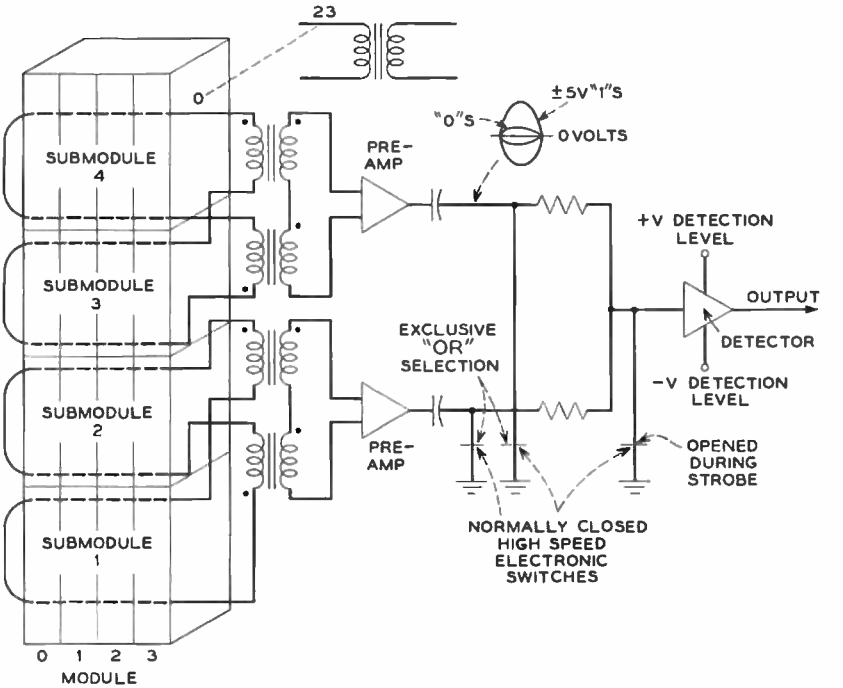


Fig. 11 — Read circuits.

closed shortly after the read sample is taken, to continue the dc restoration.

The bidirectional sampled signal is level-detected by means of a diode bridge arrangement (see Fig. 13). Input currents of either polarity in excess of the bridge currents back bias the diodes and raise the impedance, which changes the detecting amplifier from conditionally stable to unstable. The unstable state causes a severe oscillation, detectable as a "one" output. Use of a gated-oscillator technique with an amplifier normally biased in the class A region along with a current threshold detector significantly improves the sensitivity.

4.2 *Current Drivers*

A single transistor of the type available in No. 1 ESS was not capable of driving the large X or Y load with a 0.5-microsecond rise time. It was necessary to combine transistor circuits so that larger induced voltages could be accommodated. The solution chosen (see Fig. 14) combines 4

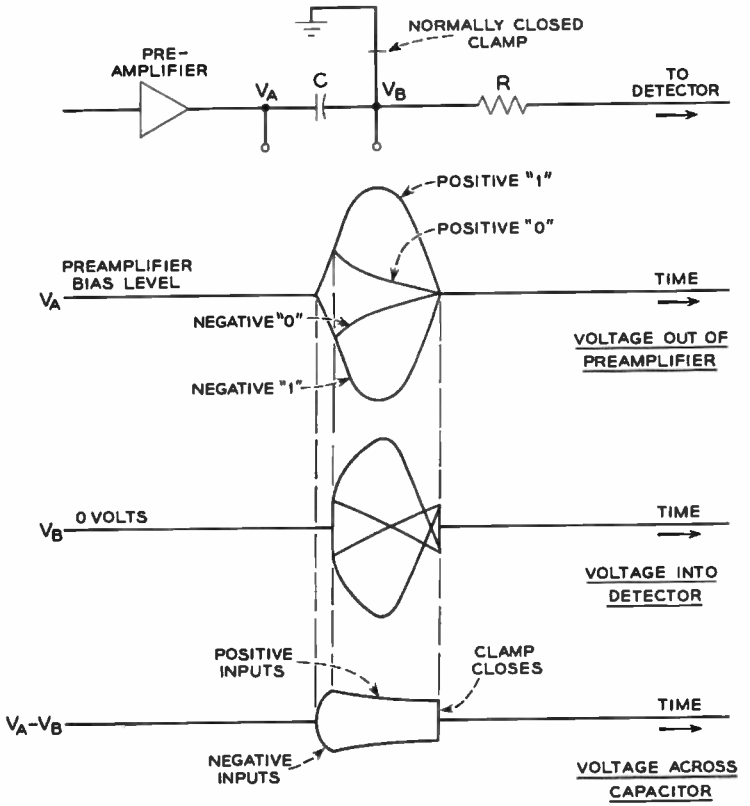


Fig. 12 — DC restoration noise reduction scheme.

separate read current drivers by means of 4 transformers with primaries in parallel and secondaries in series. A similar set of 4 write current drivers uses different windings of the same transformers; the write current drivers are fired after the read currents are turned off. The circuits, each a pulsed-current source of approximately 250 ma, combine to supply a 250-ma, 0.5-microsecond rise-time pulse in the read secondary during read time and in the write secondary during the subsequent write interval. This scheme not only distributes the load voltages but statistically improves stability, since the secondary current is equal to the average source current minus the average shunt transformer losses. The current standard deviation is one-half of a source current driver deviation, because the average of n independent but equal variables has a standard deviation reduced by $\sqrt{n}$.

Repetition rate effects are eliminated because the unbalanced transformer flux of the read interval is recovered by the reverse polarity drive of the subsequent write interval.

One added feature in this scheme is the two-step current rising shape. Rather than stagger the start of the X and Y currents so that the early X noise expires first, both currents are turned on in coincidence with a 125-ma first step and a 125-ma final step. This staircase is useful in three ways. First, the load voltage transients are smaller. Second, the early quarter-drive currents sample the coordinates without fully switching the selected address and the resultant readout noise is stored in the read preamplifier capacitor; during the subsequent full drive when the actual address is gated, the capacitor voltage is subtracted from the second step incoming signal, causing a significant reduction in delta noise. Third, the first step of the current staircase acts as a partial pre-read disturb, removing part of the undesired delta noise and precharging stray capacitance.

The variation of coercive force with temperature necessitates a reduction of drive current as the ambient increases. Currents should track within ± 2 per cent of an optimum value at any given temperature. To

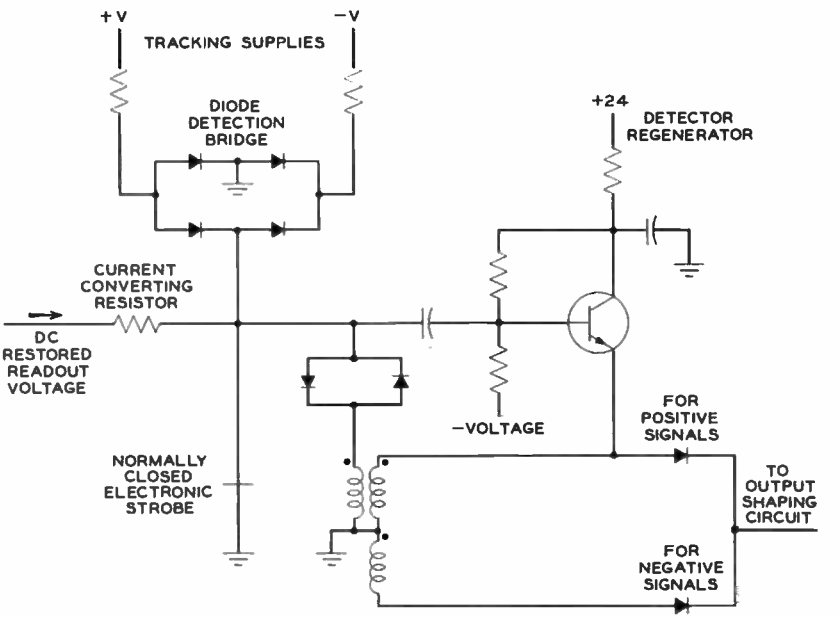


Fig. 13 — Detector.

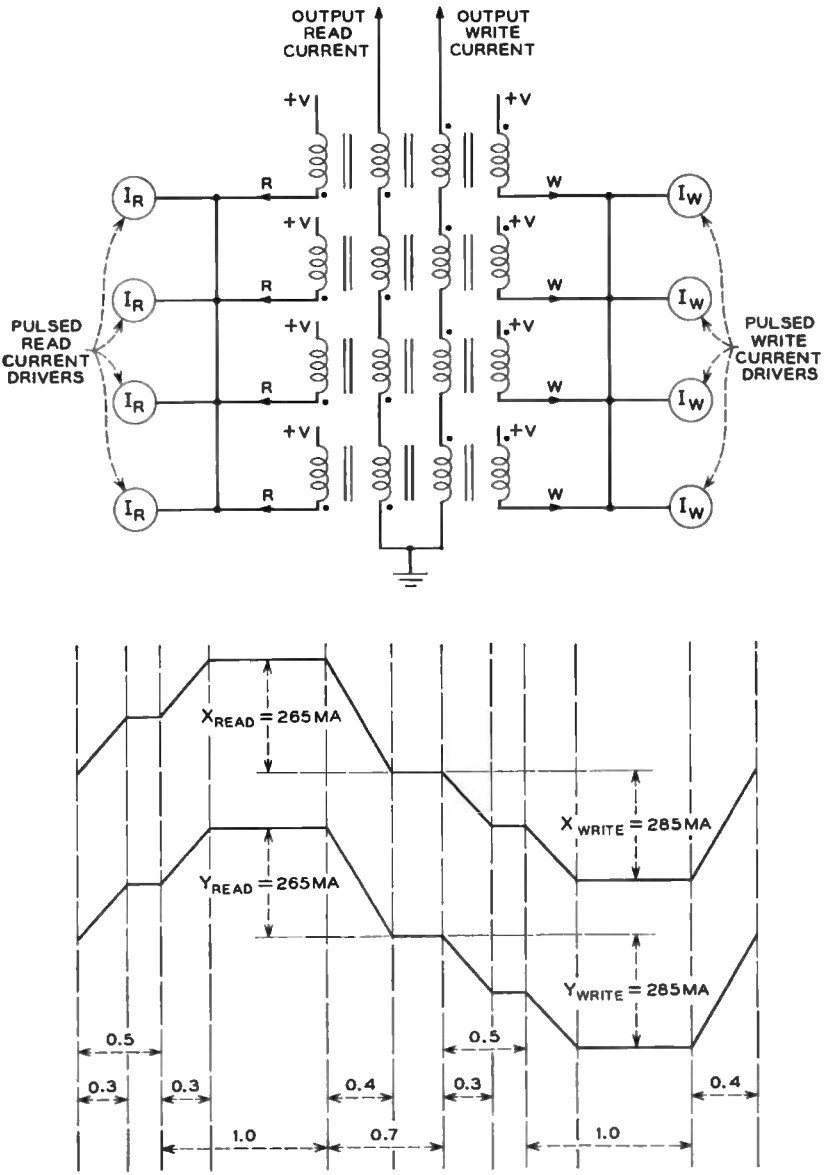


Fig. 14 — Current drivers. All values are μsec except those indicated otherwise.

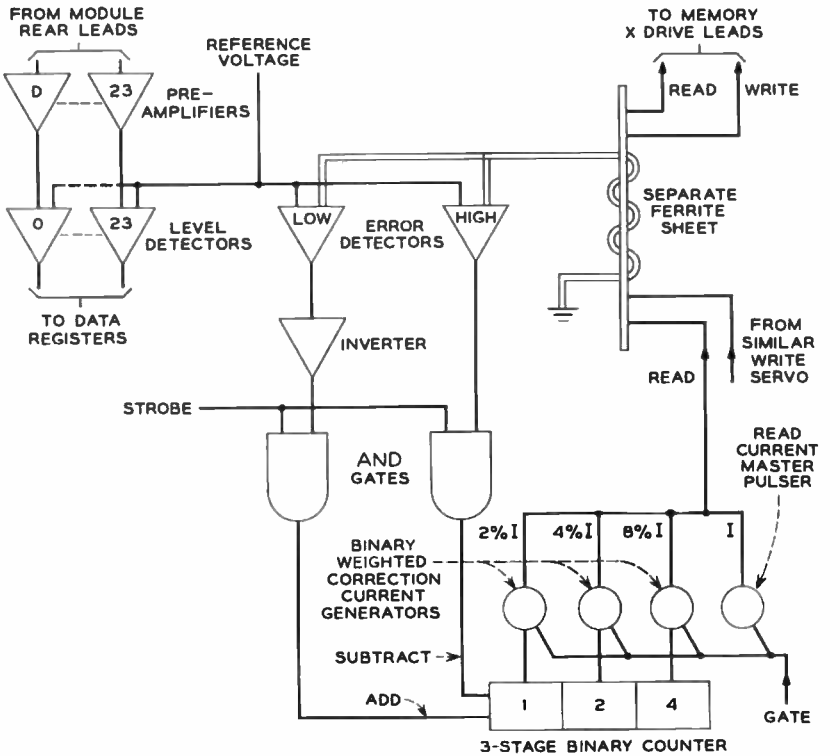


Fig. 15 — Current stabilizing servo loop.

achieve a constant readout over the ambient range of 32°F to 115°F, this optimum current must be varied by 20 per cent. The precise tracking over this large current range is accomplished by means of an electronic analog-to-digital servo loop (see Fig. 15). Two turns of the read and write drive currents pass through 16 holes of a separate sensing sheet. An equivalent 8 turns couple the sum of the 16 switching signals out of this sheet into a high- and low-level detecting circuit. If the signal is lower than a bias voltage, a one-step add command into a 3-stage binary counter is initiated. If the signal is high, a one-step subtract command is initiated. The three counter stages directly control three auxiliary current circuits whose amplitudes are binary weighted at 2, 4, and 8 per cent of the nominal value. This system automatically adjusts itself so that the sensing ferrite sheet peak output signal is constant at the high-low detector bias voltage level. However, since the ferrite sheet coercive force

changes with temperature, the currents are automatically adjusted to maintain constant peak switching voltages. This adjustment corresponds exactly to the required variation of drive current with temperature changes. Another significant feature is the stabilizing nature of the ferrite sheet, which, in addition to the extremely low aging drift, is due to the nonlinearity of the device. The product of the 16-hole switching voltage (V) and the switching time is approximately constant for full drives, but as the current (I) is raised, the voltage increases in the following manner:

$$dV/V = 4dI/I.$$

A 2 per cent variation in input current results in an 8 per cent variation in output signal. In this application the ferrite device amplifies the hard-to-detect small current drift and develops a large voltage variation which is easier to detect.

The drift effect of the high-low detector bias voltage is also reduced by tracking this supply with the readout detector level voltage. A downward voltage drift reduces the servo loop current, which in turn lowers the memory output signal. However, compensation for the decreased output signal is achieved by a simultaneous lowering of the readout detector threshold.

These current driver schemes are used both for X and for Y. Each contains an individual servo loop with independent controls. The X and Y read and write currents are also threaded through another sensing ferrite sheet, with the resulting 500-ma sum of the X plus Y read currents resetting 16 holes and the 500-ma sum of the X plus Y write currents setting the same 16 holes. The amplified voltage output of the 16 holes is used as a timing source for the readout clamp and sampling signals. Because of this arrangement the critical timing signals track the current rise-time variations.

4.3 *Access*

In this system the X and Y access circuits are identical. The 1-out-of-128 coordinate selections for the X (constructed from two sets of 1-out-of-64 selections circuit), is realized by a circuit identical to that which derives the two sets of 1-out-of-64 for the Y. Costs are minimized by employing a 3-dimensional $8 \times 8 \times 2$ diode array as the selecting source; the 1-of-128 selections can be decoded by 18 transistor switches. The positive read and negative write current bidirectional requirements, which normally necessitate a duplicate set of switches, have been satisfied with unidirectional circuits by employing a four-winding transformer-

coupled switch scheme in which a common selection of all the secondary windings determines the direction of current.

Fig. 16 illustrates a 1-out-of-64 access grouping with current drivers of the type described in Section 4.2. One of the eight horizontal and one of the eight vertical dc switches are simultaneously closed by decoding the input binary address. During read time, a closed read ac switch directs the current through the primary read winding of the selected horizontal transformer to the closed horizontal switch. This generates a pulse in the read secondary winding through the selected load, down a column line, through the selected vertical transformer, and back to the read secondary winding of the horizontal transformer. The vertical transformer is chosen by first shorting the primary to ground through the selected vertical switch and then establishing a +24-volt bias on the unselected column lines, so that all diodes in the unselected seven columns are back biased. During the subsequent write interval a closed write ac switch directs the current through the write primary winding of the chosen horizontal transformer. This reverses the current in the secondary loop and recovers the unbalanced flux generated during the read interval. The necessary biases to uniquely select one of the column lines are derived by precharge switches which insert a −24-volt bias on all unselected columns.

The combined group of the 8 horizontally arranged transformers is called an "input marker" and the combined group of the 8 vertically arranged transformers is called an "output marker." The set of 8 horizontal and vertical dc switches is shared by two groups of input markers, output markers and diode matrices. (See Fig. 17.) Each group is uniquely selected by a primary ac switch. By this technique only the passive markers and diode matrices must be duplicated when extending from a 1-out-of-64 to a 1-out-of-128 scheme.

A highly significant byproduct of the access is the precharge feature. The stray and interwinding capacitances of unselected memory coordinates can cause severe current leakages into unselected locations until these capacitances are charged. Leakages, which destroy the fidelity of the current rise-time shape, occur even if an ideal system with ideal switches is used. This system precharges all the capacitances until the unselected diodes are back biased.

4.4 *Inhibit*

The 48 inhibit drivers are simple pulsed-current sources whose amplitudes are determined by a 5-volt reference bias. The voltage is designed

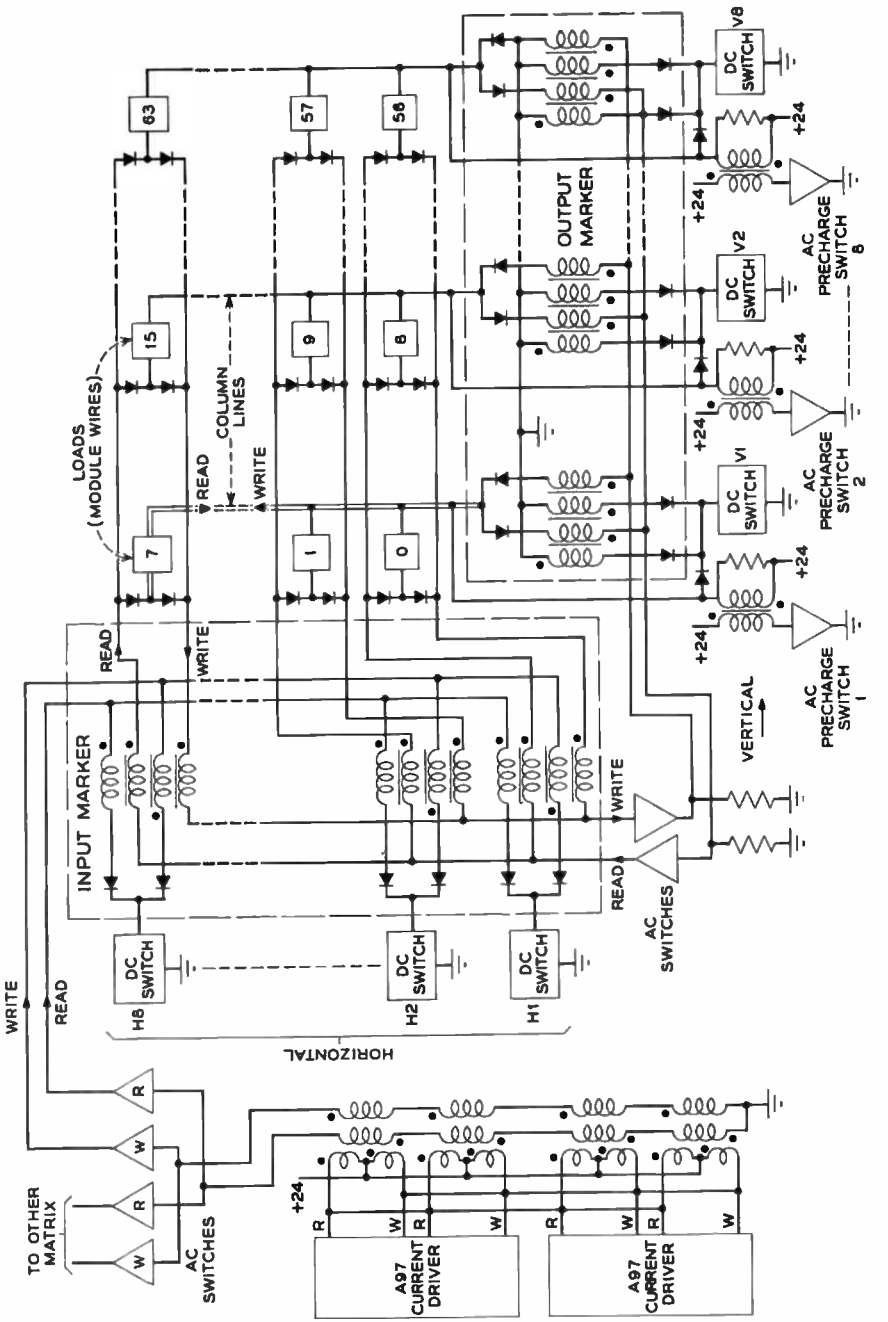


Fig. 16 — 1-out-of-64 selection matrix.

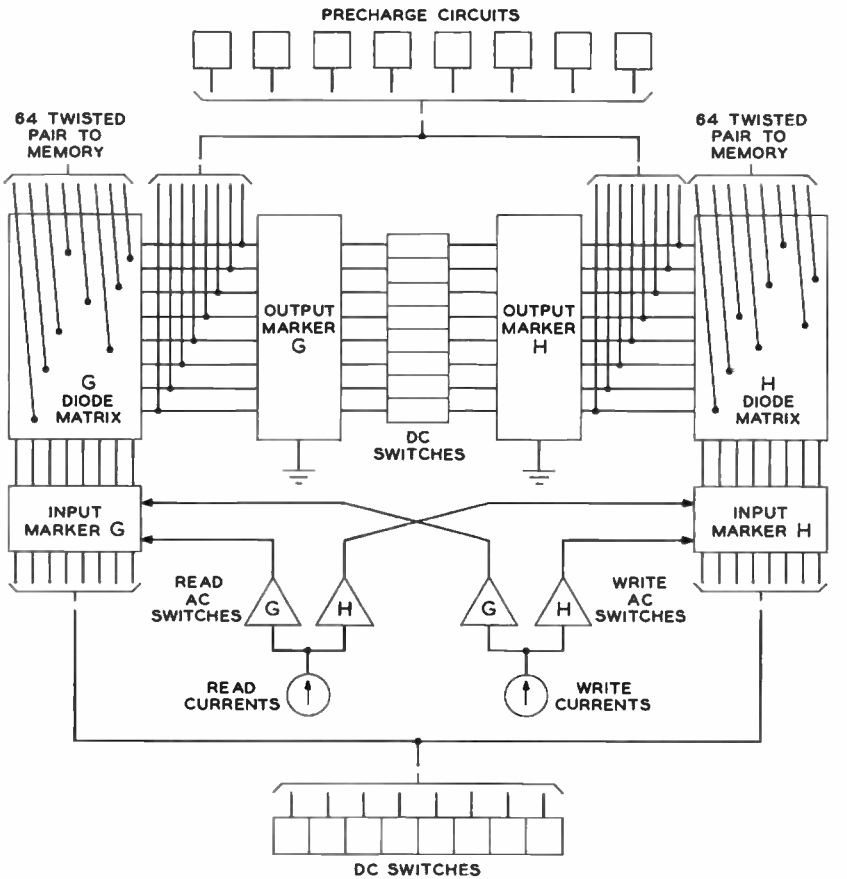


Fig. 17 — Input markers, output markers, and diode matrices.

to vary with temperature to properly compensate the inhibit current with ambient temperature. The major feature of the inhibits is the application of a post-write disturb (see Fig. 18), which reduces delta noise by referencing all holes with a half-read drive.

V. LOGIC AND SEQUENCE CIRCUITS

5.1 *General*

As shown in Fig. 1, the No. 1 ESS call store is a self-contained memory system that can read or write information, as ordered, in any externally

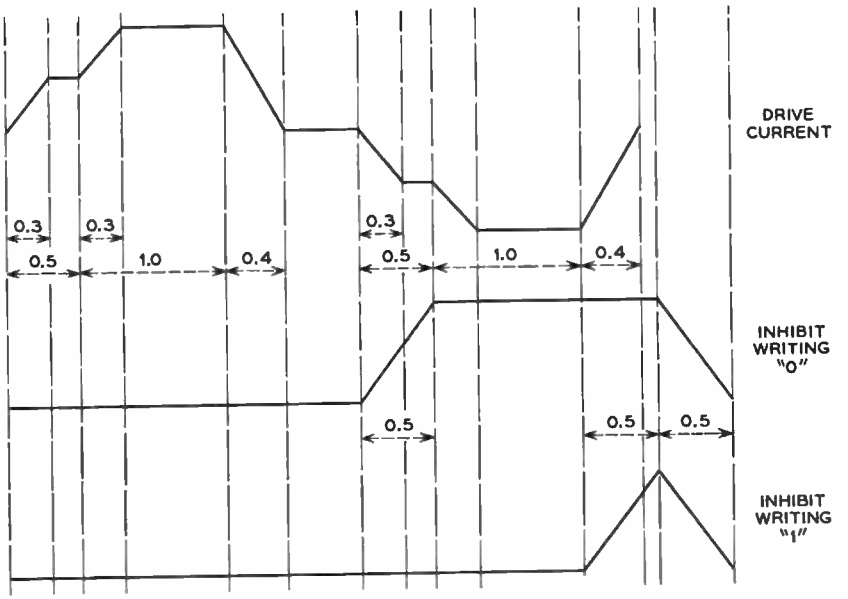


Fig. 18 — Inhibit current with post-write disturb. All values are μsec.

specified address location. It contains logic circuits such as address decoding and data registration, and timing circuits typical of memory systems. However, it also contains a wealth of maintenance and checking circuits not typical of memory systems. Its operation can be considered as an independent subsystem with self-diagnostic facilities. Much of the logic design is concerned with the operation of the call store in the larger system. Of particular interest is the common bus communication which provides duplication of memory and eliminates the need for expensive private buses in central offices with widely varying memory requirements. Other important sections are the automatic maintenance circuits, including fault-recognition and trouble diagnosis, which integrate the call store operation with the system maintenance plan.

5.2 *Common-Bus Communications*

Inputs are transmitted over a duplicated bus shared by all call stores. Each store is preset to receive from one of the buses by an internal flip-flop called the "R_0 flip-flop," which is controlled in each unit by a separate input channel. A call store will consistently receive from the same bus unless a trouble condition arises.

All call stores on the common bus receive and register every order.

An order is processed and the answer returned if the 6-bit input code matches the store name. In effect, the code is part of the information address and from the system point of view the 6-bit code and 12-bit address make up an 18-bit address for the entire office temporary memory file. For convenient bookkeeping, however, the office temporary memory is broken up into 4096 word blocks. The 12-bit address selects one word location within a block and the 6-bit code selects the memory block within the office. Recall that for growth economics the 8192-word capacity call store is divided into two 4096-word blocks called the "G half" and "H half" ("Gee" and "Haw" being a mule driver's terminology for "right" and "left" respectively). The code name of the H half is wired in a store at installation and is called the "fixed name." The G-half code name, called the "variable name," is contained in 6 call store flip-flops which are under system control and can be changed in a 5.5-μsec cycle.

Fig. 19 shows three call stores on a duplicated call store bus. As shown, all three units are set to receive from bus 0. Answers from the H half of each store are sent back on bus 0. G-half answers are returned on bus 1. A read order with a code of 1 will activate CS1 and CS3. Each store sends the answer on a separate bus, where central control match circuits can check for a fault in communication or memory action.

Routing of answers from the G and H halves is controlled by four answer flip-flops in each store. The answer flip-flops can be set in any combination, so that either half may answer on either, neither, or both buses. Answer flip-flops, like the variable-name flip-flops, can be set in a 5.5-μsec cycle by a control mode order from the central control.

5.3 *Bus Receivers and Registers*

All inputs to the call store are routed to register flip-flops, and the essential memory circuits use the semi-dc outputs of these registers.

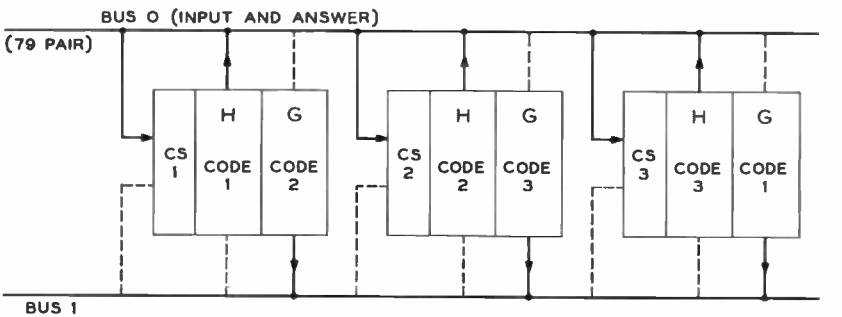


Fig. 19 — Call stores on bus.

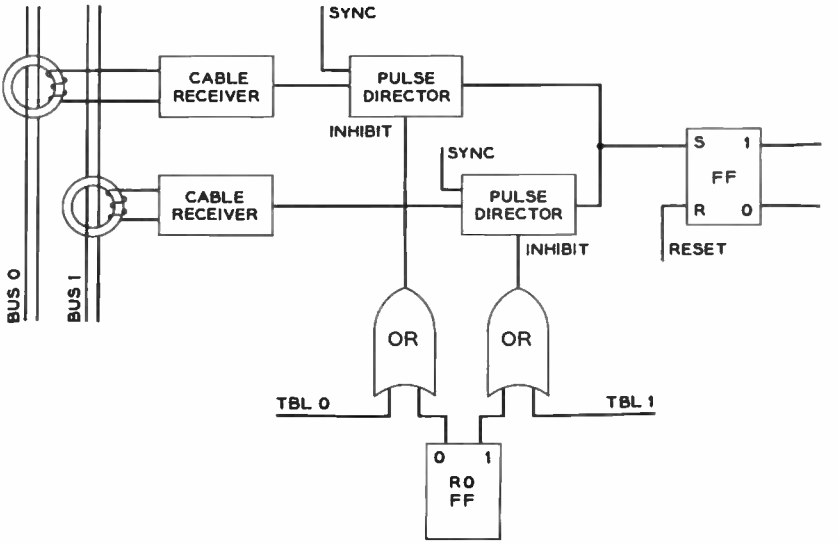


Fig. 20 — Bus input path.

Fig. 20 shows a typical input path. The inputs from the separate buses are transformer coupled to cable receivers and transmitted to "pulse director" circuits. The pulse directors gate the inputs with their associated sync pulse and may inhibit the input. Bus selection is made by inhibiting the unselected bus input at the pulse director. Also shown in the figure are the TBL0 (trouble on bus 0) and TBL1 inhibiting functions. These leads are controlled by a separate communication channel and can be used to shut off inputs from either or both buses.

The outputs of the pulse directors are logically added to set the register flip-flops. All inputs, code, mode, address, order, and data are single-rail, and the register flip-flops are reset by internally generated pulses at the end of each cycle.

The diagram in Fig. 21 gives the complete contents of the input and answer buses along with the nominal timing.

Each wave of information on the input or answer bus is accompanied by a time-coincident sync pulse. The receiving circuits require the coincidence of the sync pulse and the information to protect against false operation on bus noise.

5.4 *Modes*

In the "normal" mode two call stores respond as a regular memory to every order. Under trouble conditions it is necessary to communicate

with a single store in order to determine the location and nature of the trouble. For this purpose, two maintenance modes are used in which the call store responds to a code match with the fixed name only. Thus two maintenance modes are necessary to define the H or G half of a given store. These are designated "H maintenance mode" (HMM) and "G maintenance mode" (GMM). The call store operates as a normal memory in the HMM and GMM; however, only the fixed name is used for selection and the store answers and receives on the same bus.

The fourth and final mode of operation is the control mode. Here the call store does not operate as a memory — no ferrite is switched. In the control mode, the write operation sets flip-flops or initiates conditions in the call store and the read operation sends the state of flip-flops or other test points to the system via the answer bus. For example, setting the 6 variable-name and 4 answer flip-flops is a control mode, write operation. The first 20 bits of the 24 data-input leads carry the set and reset information for the 10 flip-flops. A corresponding control mode read operation returns the state of the variable name and answer flip-flops to the central control on the normal answer bus.

5.5 *Orders*

The read and write orders are received in a redundant code over two bus leads. In the normal and maintenance modes, R,W = 10 is a legitimate read and R,W = 01 a legitimate write. The inputs R,W = 00 or 11 are trouble conditions; however, drive currents have started before the

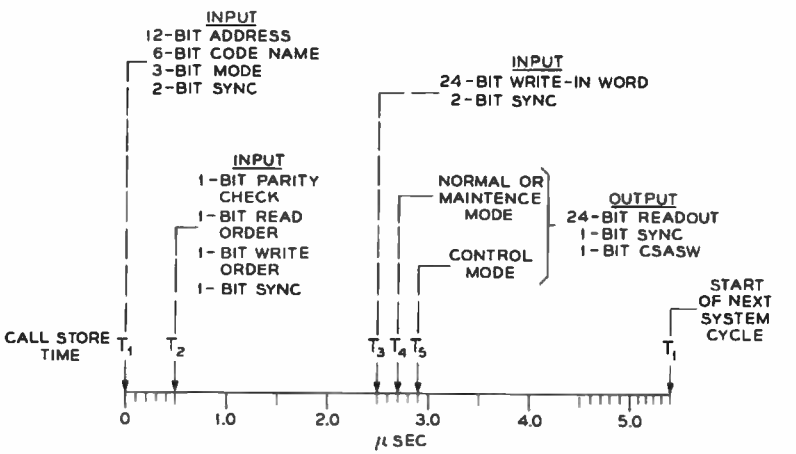


Fig. 21 — Timing of input-output signals.

order is decoded and cannot be interrupted. Illegitimate orders are therefore converted to read orders and no ASW is returned with the answer. A good write order, R,W = 01, is also converted to a read order if the ASW circuits detect any trouble. Thus information is safeguarded by forcing a regenerate sequence if any ASW failure is detected.

In the control mode all four combinations of the read-write order leads are used. Control read and control write operations are used, and the corresponding order bits R,W = 10 and 01 are required. The control functions are further defined by a control address which uses the four least significant address bits. To test the control address circuits themselves, however, it is desirable to have a control order that is independent of address so that all combinations of address inputs can be tested. To this end, the normally invalid order combinations R,W = 00 and 11 are used to read test points associated with the addressing circuits. These special orders check the Y and X portions of the address registers as well as the dc switches in the memory access circuits.

5.6 *Addresses*

In the memory modes, the 12-bit call store address defines one word in the half-store, 4096-word memory block. This block is generally visualized as a square 64 × 64 array in an XY plane. The six least significant bits constitute the Y address. The memory access system described earlier further divides the six Y bits into two groups of three bits each. Each three-bit group is decoded into a one-out-of-eight selection by the dc switches. Selection of the X-drive switches is similarly derived from the six most significant address bits.

In the control mode the address bits select one of the possible control operations in a manner analogous to the selection of a memory location. Since only a few control locations exist, only the four least significant address bits are needed for control addresses. For example, the name and answer flip-flop control address is octal 3. Therefore (see Fig. 19) if we send on bus 0: control mode, code = 3, address = 3, and order = read, the right-hand call store will reply that its variable name is octal 1 and that its answer flip-flops are set to return H-half answers on bus 0 and G-half answers on bus 1.

5.7 *Data Channels*

Call store data are handled by 24 identical channels, one of which is shown in Fig. 22.

During a read order in the normal or maintenance mode the output of

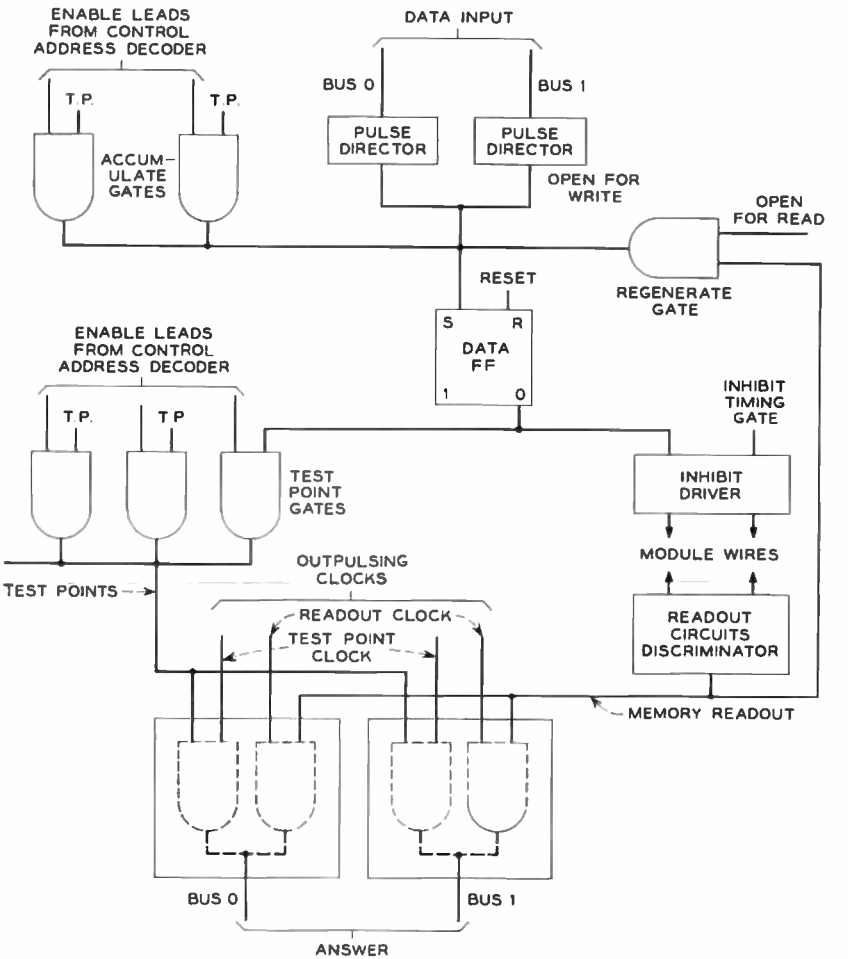


Fig. 22 — Data channel.

the discriminator is gated onto the selected bus by a readout clock pulse. If the mode is normal the four answer flip-flops select the answer bus and either, neither, or both buses could be pulsed. In addition, on a read order the regenerate gate is open and the discriminator output sets the data register. If a 1 is read from memory the data register flip-flop will be set, the associated inhibit generator will not fire, and a 1 will be rewritten in the memory during the write portion of the call store cycle. When a 0

is read out from memory the data flip-flop remains reset, the inhibit generator fires, and a 0 is written in that bit.

For a write order in the GMM, HMM or normal mode the sequence is the same except the regenerate gate is closed, the input data pulse director is enabled, and the data register is set or not set from the bus. On a write order the read part of the call store cycle is used to erase existing information.

Both input data and readout are single rail; the data registers are reset by an internally generated pulse. One of the data register reset pulses is generated at the end of the write-back drive current, 0.5 μsec before the end of the inhibit gate. Therefore all data flip-flops are reset while the inhibit gates are still open, and as a result, all inhibit drivers fire, producing a post-write disturb signal.

In the control mode the data channels are used to transmit test point information to the central control. The flip-flops themselves serve as pulse accumulating registers in two control read cases.

For semi-dc conditions that exist throughout a cycle, or at least exist when the output is gated on the bus, the test point gates are used. Pulses that occur at odd times during the cycle are checked by first giving a control write order to accumulate the pulses in the data register. This enables one of the accumulate gates shown on Fig. 22. The timing pulses, if and when they occur, set the data register flip-flops, whose contents can be read out if the next order is a data register control read.

VI. MAINTENANCE CIRCUITS

6.1 *General*

The maintenance circuits are designed to accommodate a system philosophy of fault recognition and trouble location quite different than those used in most information processing machines. The prime objective is to maintain the continuity of system operation. Thus the first action is to locate the faulty store and remove it from service. Detailed diagnosis which will isolate the fault to a specific plug-in package can then be accomplished on a deferred or low-priority basis. The possibility of 40 or more call stores in an office accents the demand that diagnosis be automatically programmed and as complete as possible.

Flexible switching provides rapid rearrangement of the bus configuration, and the variable-name facility allows the location of memory blocks to be reassigned to retain a duplicate copy of priority information. Because of the complete information duplication and the flexible communication, the system can survive even if half of the stores fail.

The existence of a trouble is generally noted by mismatch between duplicate stores, parity failure, or failure of "all-seems-well" (ASW). Mismatch and parity are clearly over-all checks that include the entire communication link between central control and the stores. Furthermore, they indicate trouble on the bus or in a complete call store rather than detecting failure of some circuit within a store. The ASW circuits do detect specific conditions within a unit, but the system's use of the ASW pulse does not elaborate on the nature of the failure; it merely specifies which store is in trouble.

Table I lists the maintenance facilities in a call store along with their primary functions.

The duplicate bus answer matching as well as the input and answer bus selection were described previously. These insure alternate routing to carry on communication in spite of failures and allow cross checks between channels to weed out faulty links. The maintenance mode orders, which enable the system to choose specific addresses in specific stores, have also been described. The functions of the maintenance mode orders as well as the remaining maintenance facilities are discussed in the following sections.

6.2 *All-Seems-Well (ASW)*

There are six conditions tested by the ASW circuit:

(1) The order combinations R, W = 00 and 11 cause ASW failure. In the normal mode they indicate trouble. In the control mode, these combinations are legitimate but they still cause an ASW failure. However, this failure is intentional and is used to test the all-seems-well circuit.

(2) Parity failure causes ASW failure.

(3) An invalid mode input causes ASW failure.

(4) Simultaneous operation in the control mode and a memory mode causes ASW failure.

TABLE I — CALL STORE MAINTENANCE FACILITIES AND FUNCTIONS

All-seems-well (ASW)	fault recognition in communications
Address parity (over code and address)	
Duplicate bus answer matching	over-all fault recognition
Stored parity (code, address and data)	
Input and answer bus selection	communication diagnosis
Fault flip-flops	logic diagnosis
Control mode orders	
Maintenance mode trouble location	memory circuit diagnosis
Monitor bus	
Scan points and central pulse distributor	store status test

(5) Simultaneous operation in both the G and H halves of the store causes ASW failure.

(6) A permanent code name match causes ASW failure.

The all-seems-well circuit makes all six tests during every call store operation, normal or otherwise. If there are no irregularities this circuit returns an ASW pulse along with the readout onto the answer bus. However, if a trouble is detected the store answer is transmitted without the ASW pulse and the call store is forced into read and regenerate operation regardless of the input order. This restriction on memory writing protects information from being destroyed when the incoming orders are incorrect and is particularly useful with address parity error detection.

6.3 *Fault Flip-Flops*

Five fault flip-flops are included in a call store to detect and register the occurrence of intermittent or transient faults. Either maintenance programs or a maintenance man using teletypewriter control can use the fault flip-flops to locate the general area of intermittent or marginal troubles.

The fault flip-flop indications provided are:

(1) all-seems-well failed,

(2) both the G and H halves were selected simultaneously,

(3) the store tried to operate in the control mode and a memory mode simultaneously,

(4) test point data were sent along with memory readout on the answer bus, and

(5) the code match circuit has a permanent decoder output (PDO).

Item (5) is particularly troublesome. A PDO means that a store will answer an order with any input code and mask the bus with unwanted signals. If PDO is detected, a call store will set both of its own trouble flip-flops, thus taking itself out of the system. This is the only case where the call store tampers with system status or bus selection.

6.4 *Monitor Bus*

The monitor bus is an 8-pair bus common to all call stores, on which test points may be connected through relay contacts to a ferrod scanner. In the call store the monitor bus is used exclusively to test regulated voltages. The outputs of two voltage regulator boards of the same type are connected through current limiting resistors to opposite sides of a bus pair, and the ferrods detect current flow if the output voltages are different and out of tolerance.

6.5 *Address Parity and Stored Parity*

Every memory mode order (normal or maintenance mode) includes a parity bit over the code and address. This parity is checked in the call store. Failure of parity causes failure of ASW and will convert a write order to a read order. Central control is informed of the trouble when it fails to receive the ASW answer pulse and can take corrective action. Since the write order is not executed, information in the incorrectly selected address is not destroyed.

Another parity bit is transmitted to the call store during memory write orders. This parity is over code, address and data. It is stored in memory as the 24th bit and returned to central control when the word is read. This parity is not processed in the call store and is treated as any other data bit. However, central control interprets the parity and checks for errors.

6.6 *Maintenance Mode Trouble Location*

Diagnosis of the memory module and its associated circuits is primarily done by storing and reading information. The faulty package is located by noting at what addresses storage fails and/or in which data channel. For example, a bad diode matrix will cause errors along one X or one Y address. A bad input marker will cause 8 consecutive addresses to fail. Bad output markers cause every 8th address to fail.

To increase the selectivity of this type of diagnosis, associated circuits are packed on the same printed wiring boards, and common operations are grouped to cause recognizable failure patterns. As an example of the latter statement, the 24-bit data register is reset in groups of 12, the inhibit generators are pulsed in groups of 8, and the regenerate gate and input pulse directors are gated in two groups of 8 and 16.

6.7 *Scanner Points and Central Pulse Distribution*

The bus routing R_0 flip-flop, the two error status trouble flip-flops, and the code match indication flip-flop have a very high degree of influence upon call store operation. Therefore the indication of state of these flip-flops is not trusted to the regular bus communication system. Instead the outputs are directly connected to a direct reading ferrod scanner.

The R_0 flip-flop and the two trouble flip-flops are so critical that separate system control of the setting or resetting commands are required.

In these cases the set and reset inputs are directly connected to a central pulse distributor which can alter the flip-flop state.

6.8 *Control Mode Orders*

The control mode operations complete the survey of maintenance features. These are essential for automatic programming diagnosis through the central office facilities. They allow the system to interrogate the inner core of a call store; many circuit failure conditions do not have to be interpreted by evaluating memory readout, but can be directly tested. A test such as the "read Y dc switches" sends back onto the answer bus the binary condition of key circuits in the Y access, which indicates the contents of the Y section of the address register and the Y-access decoding circuits.

6.9 *Control Mode Operations*

(1) Write data register,
(2) read data register,
(3) write name and answer flip-flops,
(4) read name and answer flip-flops,
(5) write — set current correcting servo counter to all 1's,
(6) write — reset servo counter to all 0's,
(7) read servo counter test points,
(8) read fault test point group,
(9) write — accumulate timing pulse group 1 in data register,
(10) write — accumulate timing pulse group 2 in data register,
(11) read Y dc switches,
(12) read X dc switches.

VII. EQUIPMENT

Equipment design of the call store consists of a highly functional layout in which the ferrite sheet memory modules, located behind a shield below the center of the frame (see Fig. 23), are surrounded by the access circuits at the sides and readout and inhibit circuits above and below. This permits the necessary short paths to be maintained for the readout connections while minimizing interference from the access and inhibit circuits. The logic control circuits and cable drivers are located near the top of the frame, below the terminal strips with which they connect. As indicated in the figure, the call store follows the general pattern of equipment design for No. 1 ESS.[7] With only a few exceptions, components are mounted on plug-in circuit packs. External connections

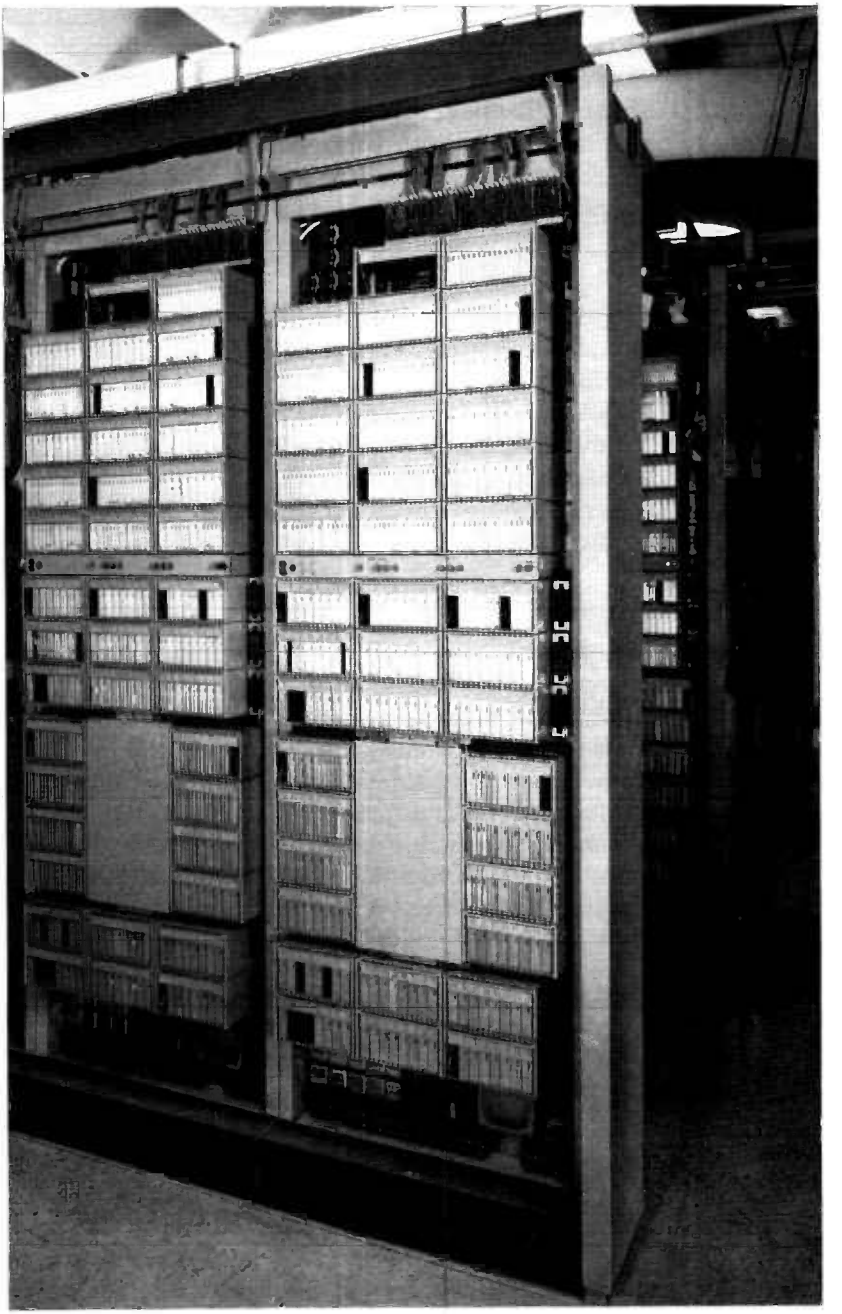


Fig. 23 — Two call stores in a No. 1 ESS installation.

are made with terminal strips and transformers at the top of the frame, while fuses, filters and other power control equipment are at the bottom. The control panel is of standard construction at a convenient height 52 inches above the floor, which is standard for No. 1 ESS. Connectors below the control panel, at the right, are for test purposes.

VIII. POWER

The entire external power supply for the call store comes from +24- and −48-volt central office battery. Emergency limits during commercial ac charging power failure can be as much as 20 per cent below the float voltages. This variation is accommodated by all circuits in the store.

Average power dissipation in a continuously addressed store is 470 watts. Unselected stores on the bus use 360 watts.

IX. OPERATING EXPERIENCE

9.1 *General*

Experience with ten call stores over a combined operating time of 80,000 hours has been very favorable.

The large number of call stores to be installed in the many No. 1 ESS installations dictates complete circuit pack interchangeability without any final adjustments in working stores. To achieve this goal most circuits have been designed without potentiometers and yet maintain tight tolerances. Some packs, notably regulators, delays, current drivers and servo reference circuits, require potentiometers to compensate for semiconductor junction differences and initial parameter differences, but they are set at the factory and will not be adjusted in individual stores.

Temperature effects on the module over the range of ambients from 32°F to 115°F, although quite pronounced, are corrected through the joint action of temperature-sensitive regulators and the servo system to keep the memory output signals well within the margins of the decision circuits. The concurrent change in switching time for the ferrite material with change in temperature is adequately compensated by deriving strobe timing from the switching signal of a standard ferrite sheet.

Although the combined effect of component drift, temperature variations and reasonable circuit pack misalignment does reduce 1 to 0 output signal separation somewhat over the ideal case, the most severe effect is the delta noise generated by the particular pattern of information stored in the memory.

9.2 *Delta Noise Considerations*

9.2.1 *General*

No single information or addressing test pattern suffices to simultaneously tax both the memory and the circuits. Furthermore, it is desirable to facilitate circuit design by separating effects and measuring their relative magnitudes. To achieve this goal, random-access, random-content memories must be exercised by a series of different programs, each generating a different type of worst case. The accumulated results of all these tests are combined to calculate the design margins of the memory. The call store must have extremely good margins because of the intended 40-year life. The huge number of call stores and the vast complex of different program combinations in the many telephone offices throughout the country almost guarantee that any remote combination that is possible will occur. For this reason the call store performance must not be address-sequence dependent or information-content dependent; it must be impervious to any possible normal sequence of events.

9.2.2 *Delta Noise Definition*

The paramount problem in design of coincident-current memories is the delta noise due to accumulation of half-current selected memory locations. Consider first two wires through a single hole in a ferrite sheet (or through a single core). If a half-drive current were passed through one of these wires, it would develop a relatively large signal across the other winding by direct coupling. Now if we consider two windings through a pair of holes, the situation is quite different. The readout winding can be threaded through the two holes so that it couples the drive winding in the opposite sense at each hole. A half-drive current through the pair of holes would induce signals of opposite polarity in the readout wire, and the net signal would be zero. From this it follows that along any memory drive wire there must be an even number of holes (or cores); half of these holes must couple the readout wire with a positive polarity, while the other half couple the readout wire with a negative polarity. In a two-dimensional coincident-current memory both axes must satisfy the above condition. Therefore, it is convenient to think of the holes along a drive winding as forming "canceling pairs." For example, a square array of 1024 holes in a 32 × 32 coincident-current memory plane would have 16 canceling pairs in each dimension.

At this point delta noise will be defined as the summation of the mag-

netic unbalance in canceling pairs due to the half-read drive currents. The large capacitively coupled signals are not as important, because very good cancellation can be achieved by proper wiring techniques and common mode noise rejection techniques. Magnetic unbalance can exist, however, and it is important to consider it from two separate sources.

The output signal from a canceling pair due to a half-drive current is zero if (1) both holes are in the same state (i.e., both 1's or both 0's) and (2) both holes have had the same past history of disturbances. The past history that is important is the polarity of the last half drive, either half read or half write. Condition 1 in the above statement depends only on information contents stored in the memory plane. Delta noise, arising when this condition is violated, will be called pattern noise or ΔP. The past history of disturbances is a function of the sequence of operations and addresses used in the memory and will be called sequence noise or ΔS.

9.2.3 *Hysteresis Model*

The 4-state hysteresis diagram shown in Fig. 24 illustrates sequence and pattern delta noise. This model assumes that a single half read will walk down a 1 hole from state 1W to 1R or a 0 hole from 0W to 0R. Further half-read pulses will not change the conditions, but a single half write will restore a 1 hole to 1W or a 0 hole to 0W. Experimental studies on small groups of cores, single ferrite sheets and the complete ferrite sheet module indicate that this model and its assumptions are very good. The different conditions of canceling pair noise and their relative amplitudes are given in Table II.

Pattern noise which is due to the unbalance of 1R and 0R pair depends on the information content of the memory and very little can be done about it. Asymmetric drive currents with write current larger than read current reduce pattern noise over the symmetrical-drive case, but it cannot be eliminated. The asymmetric drive apparently distorts the *B-H* curve and tends to equalize the slope of walked down 1's and 0's. This effect is limited by partial switching due to the oversize write half drive and inhibit, which can deteriorate storage.

Sequence noise, on the other hand, can be eliminated by providing a uniform "last" operation for all holes. The most popular method of doing this is to use a "post-write disturb" pulse. Generally, post-write disturb is accomplished by firing all inhibit drivers after writing is completed. The inhibit drive is equivalent to a half read; therefore, all holes would be in the 0R or 1R state of Fig. 24. The cure has its cost, how-

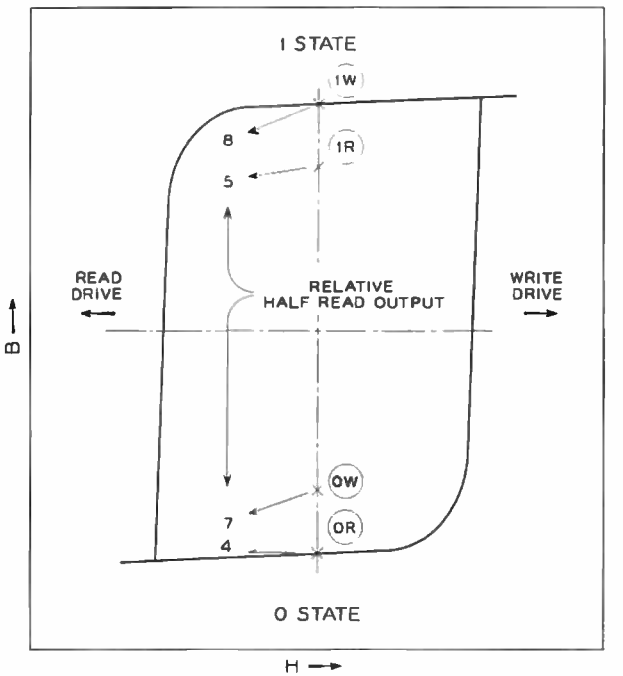


Fig. 24 — Four-state model B-H curve of ferrite sheet.

ever, in cycle time. Inhibit post-write disturb is particularly inefficient, because generally the inhibit winding is very long and has a high inductance. This makes it difficult to produce fast rise and fall on the post-write disturbs. The walk-down phenomenon, on the other hand, is very fast. Unlike hole switching, which for ferrite sheets requires 1 μsec, walk-down from 0W to 0R or 1W to 1R is accomplished during drive current rise time alone.

Sequence noise requires consideration, even though a remedy is at

TABLE II — CANCELING PAIR STATES AND RELATIVE DELTA NOISE AMPLITUDES

State of Holes (Canceling Pairs)	Delta Noise	Relative Amplitude (16 Canceling Pairs)
0R vs 0R	none	1.6
0R vs 1R	ΔP	16
0R vs 0W	ΔS	83
0R vs 1W	$\Delta P + \Delta S$	100

hand, because sequence noise represents energy in the ferrite material, and if this energy is transferred by a post-write disturb or any other disturb, it must be provided for in the readout circuits. In other words, a post-write disturb will generate an output signal equal to the noise condition it corrects, and this signal must be dissipated before the next readout appears.

9.2.4 *Delta Noise Test Patterns*

The basic test pattern for studying delta noise in a two-dimensional coincident-current memory is the pattern which writes all positive readout holes to the one state and the equal number of negative readout holes to the zero state. Fig. 25 shows a 32 × 32 bit plane pattern that satisfies this requirement. It also represents the worst-case delta noise pattern for call stores, because the duplication of readout and Y-axis system reduces the plane of intersection of the X- and Y-access wires with a given readout to a 32-by-32 array.

The figure does not represent the relative location of holes on any physical plane. The X-drive wire, for example, may thread alternately through positive and negative holes, or through any grouping of positive and negative holes. Translation of the address assignments is necessary to produce the four-quadrant picture; any coincident-current memory can be represented in this way.

Ignore for the moment any sequence effect and suppose the pattern of Fig. 25 is continuously repeated. Each read does indeed include the

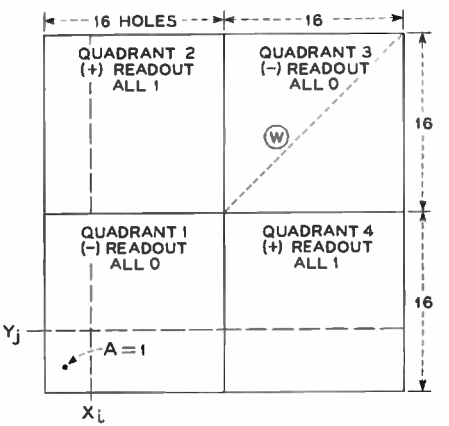


Fig. 25 — Address plane test patterns.

worst case of pattern-dependent delta noise (ΔP), but the noise component is always in the direction to enhance the signal separation by increasing 1's and decreasing 0's. For example, in quadrant 1, the 0's are negative and ΔP is positive; in quadrant 2, both the 1 readout and ΔP are positive.

To measure pattern-dependent delta noise that degrades the 1 to 0 ratio with consistency over every hole in the memory, a double-operation checkerboard pattern can be used as follows. Scan the worst-case delta noise pattern (which is a checkerboard) sequentially with two operations on each hole. First read a hole and write it to the opposite state. Second, read the hole and write it to the original state. On the second read, delta noise will be of a polarity to degrade the signal.

9.2.5 *Sequence Patterns and the "Worst" Worst Case*

Measurement of sequence-dependent delta noise not only adds complications to the testing procedure but also creates some pitfalls that must be avoided. The great opportunity to err comes from the fact that the polarity of the sequence-dependent delta noise can be independent of pattern; therefore it is possible to cancel sequence noise against pattern-dependent noise and obtain measurements that do not reflect the worst case of anything. For the call store the following statements apply:

Pattern-dependent delta noise from a 1,0 canceling pair will have the same polarity as the 1 hole.

Sequence-dependent delta noise from a half-read, half-write canceling pair will have the same polarity as the half-written hole.

To generate the "worst" worst possible delta noise on a test hole A the following operations are necessary:

(1) Write the basic worst-case pattern as shown in Fig. 25.

(2) Write the single test hole A to the opposite polarity. In the figure, A is shown in a quadrant of 0's, so it must be written to 1.

(3) Write a 0 somewhere in the memory. This causes an inhibit current which half reads all holes in the bit plane and neutralizes the effect of writing the 1 in test hole A.

(4) Write 1's along the diagonal W. This makes the "last" operation on every hole in quadrants 2, 3 and 4 a half write. The "last" operation on all holes in quadrant 1 is still the half read due to the previous inhibit pulse of step (3).

(5) Readout test hole A. The readout will be a negative 1. ΔP and ΔS will both be maximum and positive.

The zero "worst" worst case can be generated in a similar fashion.*

9.2.6 *Results*

Testing for "worst" worst case delta noise is extremely difficult, because practical test equipment becomes very complex. However, a partial implementation of this condition has been realized with the "worst" worst case along one axis and the worst case along the other axis. Measurements made on 10 stores over a temperature range of 32–115°F, with currents and circuit parameters perturbed within their expected life variations, have indicated signal-to-noise ratios in the order of 3 to 1.

Using post-write disturb to neutralize sequence noise, the 3-to-1 signal-to-noise ratio or, more accurately, 1-to-0 ratio, is measured with the double-operation worst-case checkerboard described in Section 9.2.4 and the drive currents at the extremes of the servo control (± 2 per cent). The measurements are made by changing the discriminator level reference voltage until a 0 is detected as a 1 and until the first 1 is detected as a 0. The nominal value for the discriminator reference is 19 volts. Typical measured signal separations range between 6 volts to 26 volts and 10 volts to 29 volts. The major factors in separation variations are the memory modules themselves and the unclamp and strobe timing differences due to the variability in logic circuit delays.

X. SUMMARY AND CONCLUSIONS

Using ferrite sheets with relatively high Curie temperature, an economical, fast, high-capacity temporary memory unit capable of operating over a 32–115°F temperature range has been achieved.

Many novel circuit features are incorporated to fully exploit the inherent device capabilities.

To achieve the high degree of dependability essential in an electronic switching system, it contains numerous maintenance facilities as well as provisions for rapid reassignment of memory function by means of a flexible intercommunication system.

The experiences with ten stores, in operation for more than 80,000 memory hours during the past year, have proven that the stringent design objectives necessary for reliable and dependable performance have been achieved.

---

* $+0$ with $+\Delta P$ and $+\Delta S$ is impossible to generate. -0 with $+\Delta P$ and $+\Delta S$ is the worst practical case.

XI. ACKNOWLEDGMENTS

The authors wish to acknowledge the contributions of many others to this development, particularly those of E. Ley for the store mechanical design, C. W. Thulin for his efforts in the ferrite sheet module development, J. E. Mack for his part in the early phase system design, E. H. Siegel for his novel circuit designs, C. G. Corbella and R. W. Rolund for their accurate analysis and evaluation, A. A. Stockert for his work on maintenance aspects, and R. W. Ketchledge for his counseling and guidance during the development.

REFERENCES

1. Harr, J. A., Taylor, F. F., and Ulrich, W., Organization of No. 1 ESS Central Processor, B.S.T.J., this issue, p. 1845.
2. Downing, R. W., Nowak, J. S., and Tuomenoksa, L. S., No. 1 ESS Maintenance Plan, B.S.T.J., this issue, p. 1961.
3. Connell, J. B., Hussey, L. W., and Ketchledge, R. W., No. 1 ESS Bus System, B.S.T.J., this issue, p. 2021.
4. Meinken, R. H., Ferrite Sheet Memory, Proc. 1960 Electronic Components Conference, Washington, D.C., May 10–12, 1960.
5. Meinken, R. H., A Memory Array in a Sheet of Ferrite, Proc. Magnetism and Magnetic Materials Conference, October 16–18, 1956, p. 674.
6. *Digital Applications of Magnetic Devices*, ed. A. J. Meyerhoff, et al., John Wiley and Sons, New York, 1960, pp. 372-390.
7. Ferguson, J. G., Grutzner, W. E., Koehler, D. C., Skinner, R. S., Skubiak, M. T., and Wetherell, D. H., No. 1 ESS Apparatus and Equipment, B.S.T.J., this issue, Part 2.

No. 1 ESS Switching Network Plan

By A. FEINER and W. S. HAYWARD

(Manuscript received January 13, 1964)

An eight-stage space division switching network with ferreed crosspoints was adopted for No. 1 ESS. It has a low crosspoint count and is adaptable to a wide range of office sizes and traffic parameters. This article discusses the network topology, control philosophy, and traffic aspects.

I. INTRODUCTION

This article presents the topology, traffic properties and control of the switching network for No. 1 ESS. Companion articles describe the physical implementation of the network[1] and its control.[2] In the near future, an article dealing with the program control will be published.

To assist the reader with unfamiliar terms, a brief list of definitions is given in the Appendix.

In connection with the plans to develop an electronic central office, the problem of switching network design has received much attention. Due to the many, often conflicting, requirements and possible choices of technology and geometry, the synthesis process requires a fair measure of that ill-defined catalyst commonly referred to as "intuition." The invention and the successful development of the ferreed[3,4] provided the technological solution that resolved the early difficulties of all-electronic networks. The ferreeds provide a metallic transmission path while retaining high switching speed.

Among the early recognized requirements was the desire to use the switching network not only for the obvious function of interconnecting lines and trunks, but also for all link functions — connections between signal transmitters and trunks, ringing circuits and lines, etc. The reasons behind this requirement were to simplify trunk circuits, to eliminate the problem of engineering and administering several different networks, to simplify control, to provide the connecting function at high efficiency and to provide full freedom to associate trunks with all types of signaling circuits.

The field of application envisaged for No. 1 ESS encompasses offices

from just a few thousands to many tens of thousands of lines, with line occupancies and ratios of intraoffice to interoffice traffic highly variable from one office to the next. From the standpoint of manufacture and engineering, it is important that this wide range of requirements be met with a single, standard, but adequately flexible, network plan rather than with numerous custom-tailored solutions.

This approach, however, leads to a compromise and results in some loss of efficiency at the extremes of the parameter range.

The total cost of the network can be viewed as consisting of three main parts:

(a) the cost of network crosspoints

(b) the cost of equipment directly associated with network controls (this usually grows with the number of links), and

(c) the proportionate share of central processor cost applicable to the handling of the network.

Freedom from standard matrix sizes embedded in the electromechanical technology (such as crossbar switches) and promise of more subtle network control methods opened the way for considering networks with a larger number of switching stages containing, perhaps, switches of smaller dimension. It was felt that this would permit attaining large network sizes, while retaining a low crosspoint count per line.

The network plan developed for the Morris, Illinois, trial exchange,[5] largely upon ideas of Mr. C. E. Brooks, underwent much scrutiny and served as a point of departure. It was felt that a further crosspoint saving could be made by changing the topology; more importantly, an improvement in growth characteristics was sought.

Independently from studies of central office networks, an exploration of suitable remote concentrator arrangements was being carried out at that time. It yielded the two-stage concentrating configuration shown in Fig. 1.

The first stage of the concentrator is formed by four 16×8 switches in which each of the sixteen inputs has access to only four output links. The placement of crosspoints is identical to the position of diodes in a binary to one-out-of-sixteen diode translator. As a result, every input has access to each of the four pairs of output links. Since these are distributed in pairs over the four-output 8×4 switches, the resulting configuration provides access from every input to every output.

The numerical elegance of this pattern, the economy of crosspoints, and certain other properties (such as the fact that the first-stage switch maps into a full 8×8 switch) made a full-scale investigation appear worthwhile.

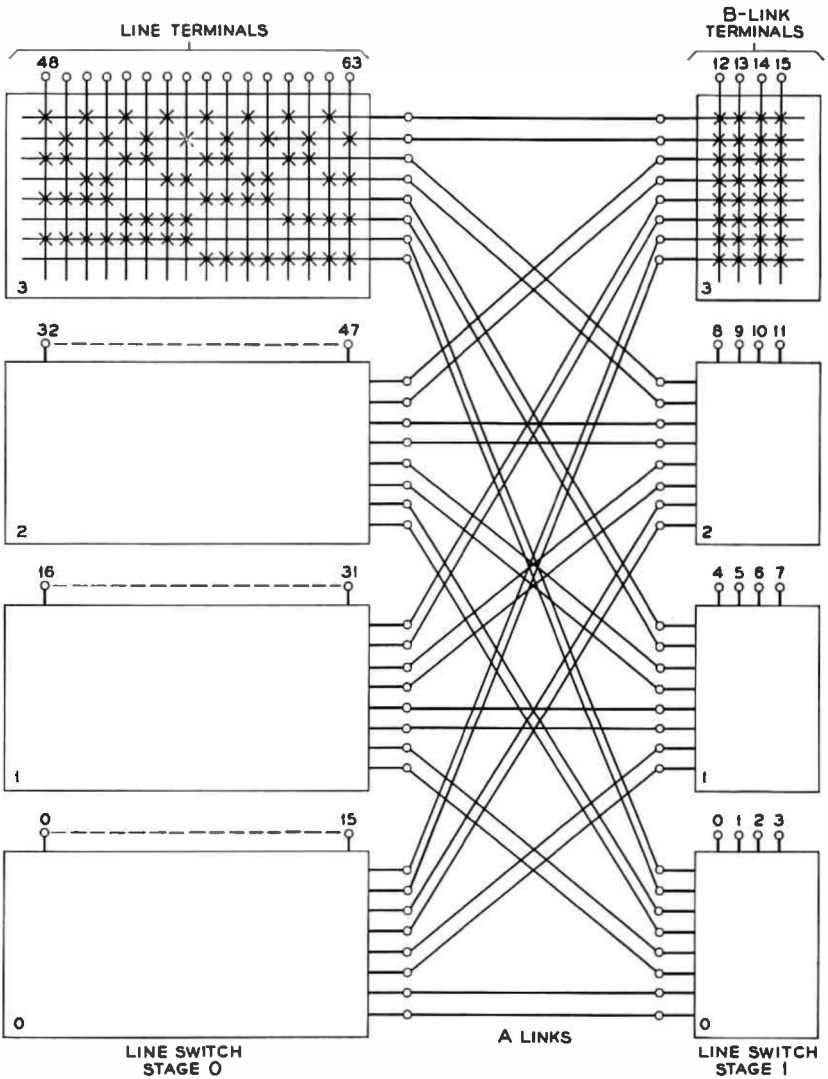


Fig. 1 — A 64-to-16 two-stage concentrator arrangement.

The performance capabilities of this concentrator were studied with the help of a computer simulation; the results will be discussed later in this article.

The attempt to solve the problem for a concentration ratio of four originated in the knowledge that the average occupancy of a line in the

Bell System is about 0.1 erlang and that most economical switching networks investigated at that time were capable of internal link occupancies of 0.4 erlang.

II. THE OCTAL NETWORK

2.1 *Over-All Plan*

The binary nature of the ESS control language led to adoption of switch and grid sizes characterized by numbers of terminals that are powers of two to realize translation and control economies. The choice of switch size was made on the basis of studies of physical design, control cost and number of switches needed to meet objective size and traffic capacity. Of the binary sizes, 4×4, 8×8 and 16×16 were obvious contenders; 8×8 was chosen. Considerations of access and blocking in the largest network size led to the adoption of a network with eight stages of switching.

Topologically, the network consists of four-stage groupings of which there exist two types — the line link networks and the trunk link networks. Connecting these subnetworks among one another are junctor groups provided in a pattern consistent with the specific size and traffic character of a given office.

As the name implies, the subscriber lines connect to the line link network. Two basic sizes exist for the line link network. One, with a concentration ratio of 4:1 in the first two stages of switching, provides terminations for 4096 subscribers; the other, developed with higher traffic loads in mind, has 2:1 concentration and provides terminations for 2048 inputs. A constant number of 1024 junctor terminals characterizes all (fully equipped) link networks.

Trunks and service circuits connect to the trunk link networks; these have the basic size of 1024 trunk terminals.

Fig. 2 shows an example of the network in an office with approximately 8000 subscribers and 1000 trunk and service circuits. A novel feature of the octal network is the method of handling intraoffice calls; these are routed on direct intraoffice junctors that link the line link networks among one another and to themselves and bypass the trunk link networks. The intraoffice junctors contain the circuitry to apply battery and to supervise both subscribers.

2.2 *The Line Link Networks*

The first two stages of the line link network contain a concentrator arrangement, shown in Fig. 3(a) for the 4:1 ratio and Fig. 3(b) for the 2:1

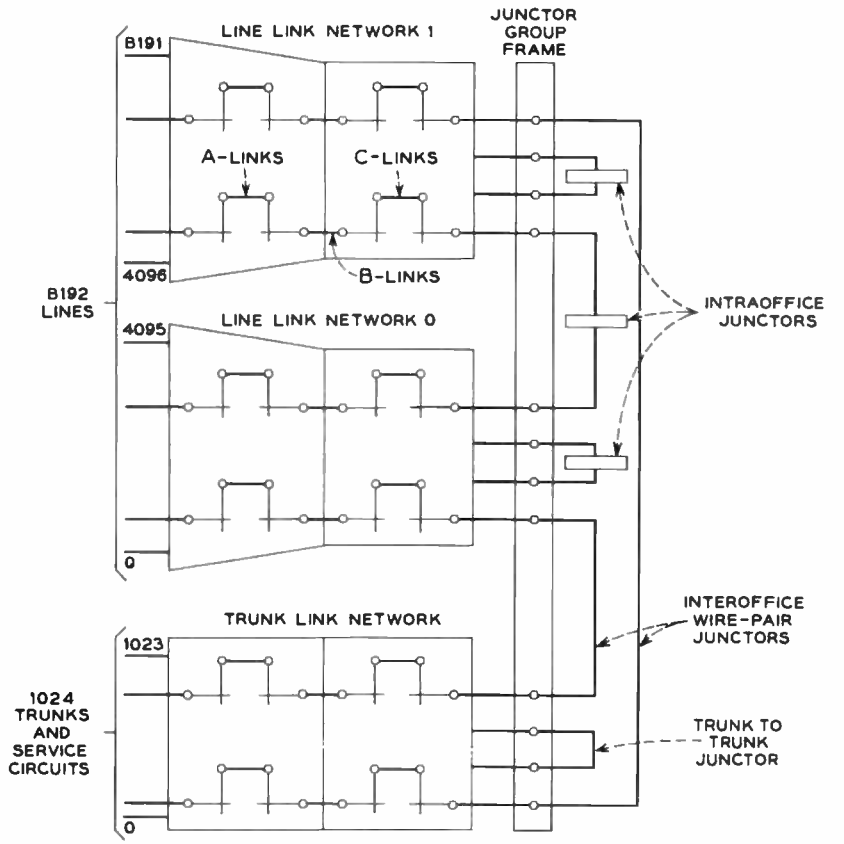


Fig. 2 — An example of the network for an office with 8000 lines and 1000 trunk and service circuits.

ratio. The first is a slight modification of the previously described concentrator; the change in the first-stage switch pattern was found to sacrifice little traffic carrying capability and simplified the internal structure of the ferreed switch. The third and fourth switching stages are formed from 8 × 8 switches in both types of line link networks, organized into 16 grids (see Fig. 4). This configuration provides every line with full access to the 1024 junctors. It is convenient when dealing with grid networks of this type to express the access within the network as the product of the individual switch access numbers

$$A_{LLN} = a_1 \cdot a_2 \cdot a_3 \cdot a_4$$

and with $a_1 = a_2 = 4$ and $a_3 = a_4 = 8$

$$A_{LLN} = 1024.$$

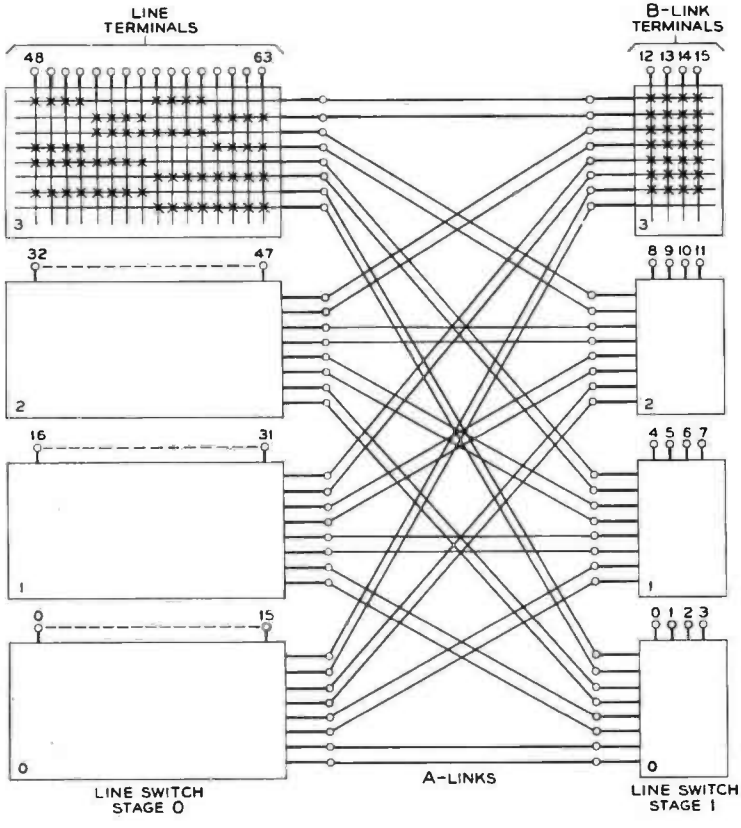


Fig. 3 — (a) A modification of the 4:1 concentrator of Fig. 1 for No. 1 ESS network. (b) The 2:1 concentrator connection pattern.

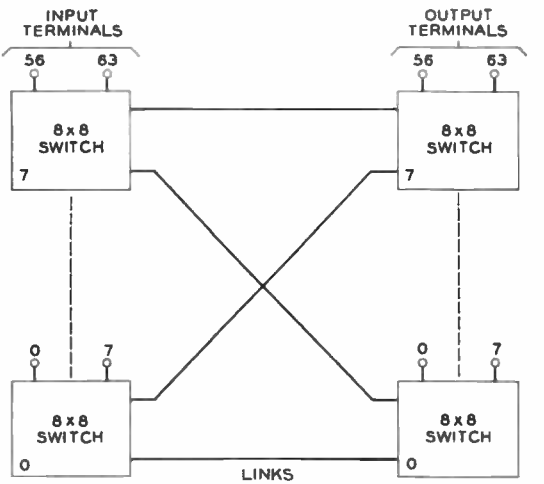


Fig. 4— The 8 × 8 grid.

This provides a quick verification of the full-access nature of the line link network; needless to say, proper link distribution between the concentrating stages and the 8 × 8 grids must be observed. In this case there must be one B link between every concentrator and every grid.

The equipment design of the network is discussed fully in other articles.[1,2] It will suffice here to say that the line link network contains two types of frames — the line switch frames, into which are packaged 16 concentrators, and the junctor switch frames containing eight octal grids each. In addition to the crosspoint arrays, the line switch frames house the equipment that constitutes the line circuits, namely the line scanner and the ferreed devices for removing the current-sensing ferrod element from the line after a service request has been detected and registered. The junctor switch frames also contain additional ferreeds, one per junctor, that provide test access into established network connections. Fig. 5(a) shows the composition of a line link network in terms of these equipment units.

All of these equipment frames contain their own duplicated control circuits. These circuits receive and translate orders from the central processor and perform all the other functions that lead to their execution. Typically, an order calls for the closure of a specified path through the two switching stages contained in the frame.

The two duplicated control units work independently of each other; when both are functioning properly, each restricts its activity to its assigned half of the network contained in the frame.

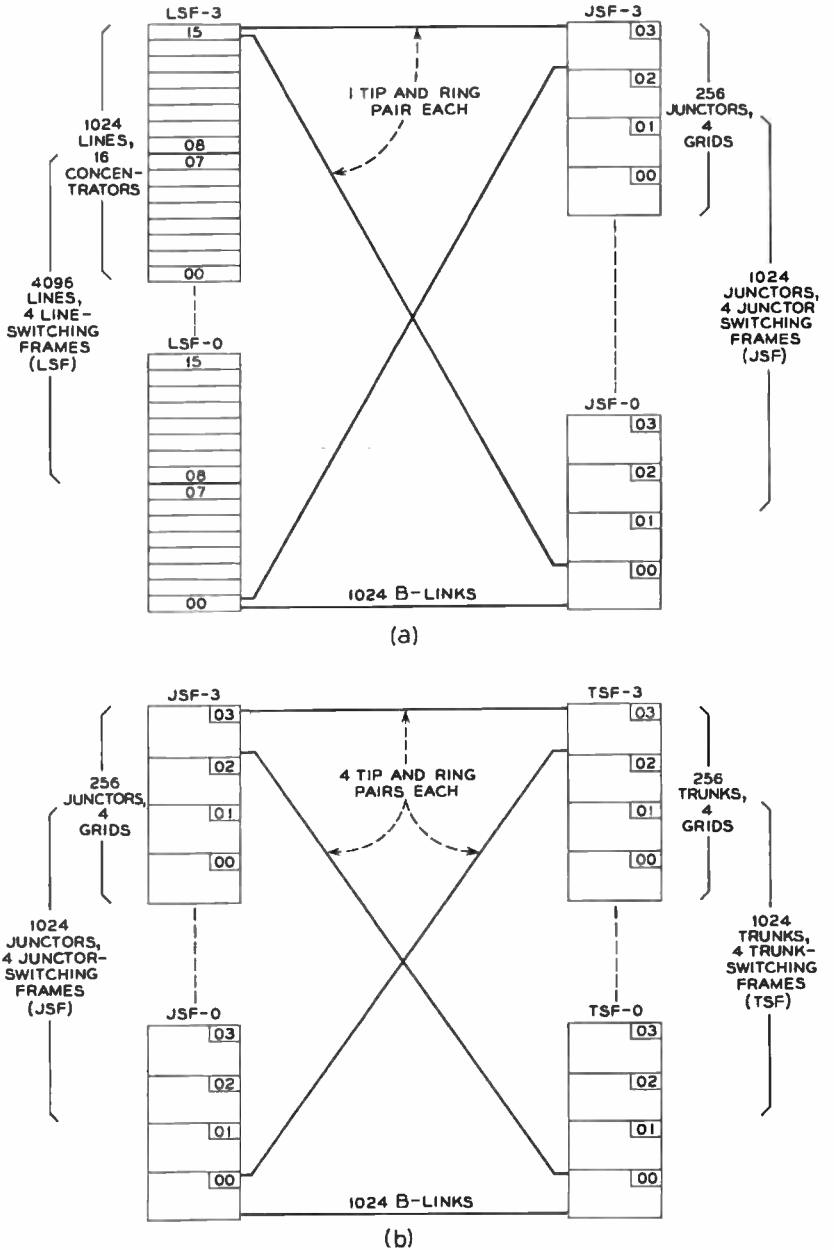


Fig. 5 — (a) Line link network (with 4:1 concentration). (b) Trunk link network.

2.3 *The Trunk Link Network*

When the trunk link network was first studied, it was proposed to construct it as a three-stage network containing a 16 × 16 switch at the trunk side followed by two stages of 8 × 8 switches. This naturally resulted in the convenient full-access subnetwork of 1024 inputs and 1024 junctor outputs (16 × 8 × 8 = 1024). Further investigation has shown, however, that at the slight control complication of introducing another stage, a substantial gain in traffic carrying capacity could be realized with the same number of crosspoints per terminal by going to four stages of 8 × 8 switches.

Since the trunk link network access is thus increased to $A_{TLN} = 8^4 = 4096$, if the same size of the trunk link network of 1024 × 1024 is retained, a multiple access from trunks to junctors is obtained with every trunk capable of reaching every junctor by four different paths. This reduces considerably the trunk-to-junctor blocking despite the addition of a stage of switching.

Two types of equipment frames are contained in the trunk link network. One of them is the junctor switch frame that also serves as a building block for the line link network. The other, the trunk switch frame, contains the same number of octal grids but has none of the test access provisions of the junctor switch frame. Fig. 5(b) shows the composition of a trunk link network.

2.4 *Interconnection within Line Networks*

Fig. 6 gives a three-dimensional view of a full line link network, showing in skeleton form the way in which frames are connected together. Concentrators are shown as horizontal planes; octal grids are shown as vertical planes. The connections between line switch frames and junctor switch frames, B links, are shown only for the edge of the link network. This diagram gives a picture of how all lines have access to the 16 B links of their concentrators and of how each B link connects to a different octal grid in the junctor switch frame. The grids in turn give each B link access to 64 junctors. Fig. 7 gives a similar representation of a full trunk link network. Pictorial representation is more difficult here. Each trunk has access to 64 B links which are treated as four groups of 16 links. Each group of 16 covers the 16 octal grids of the junctor switch frames.

2.5 *Partial Network*

The size of a link network is large. Equipment frame sizes, on the other hand, have been so chosen that the incremental cost of buying equipment

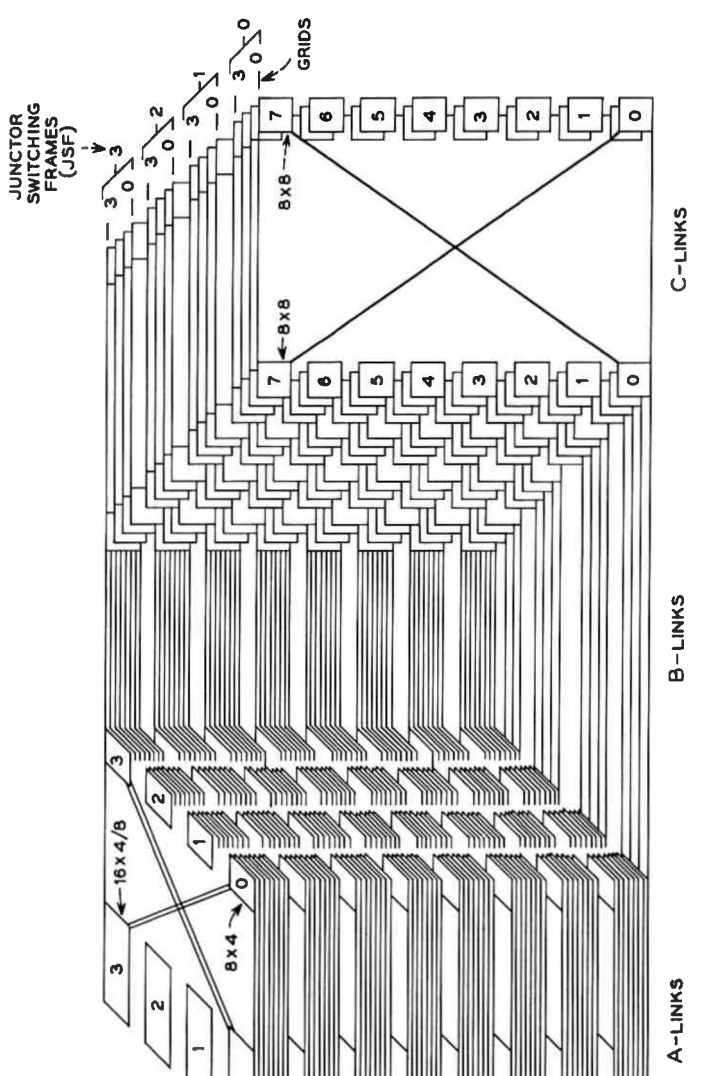


Fig. 6 — Schematic diagram of line link network.

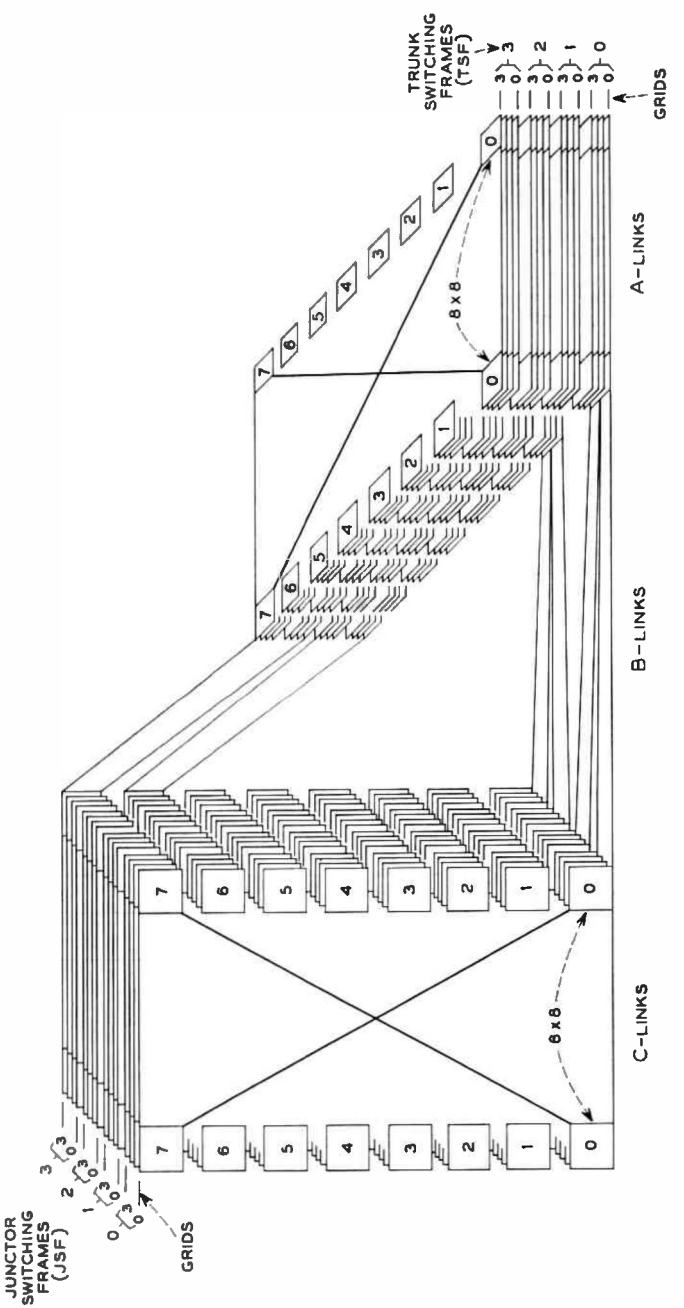


Fig. 7 — Schematic diagram of trunk link network.

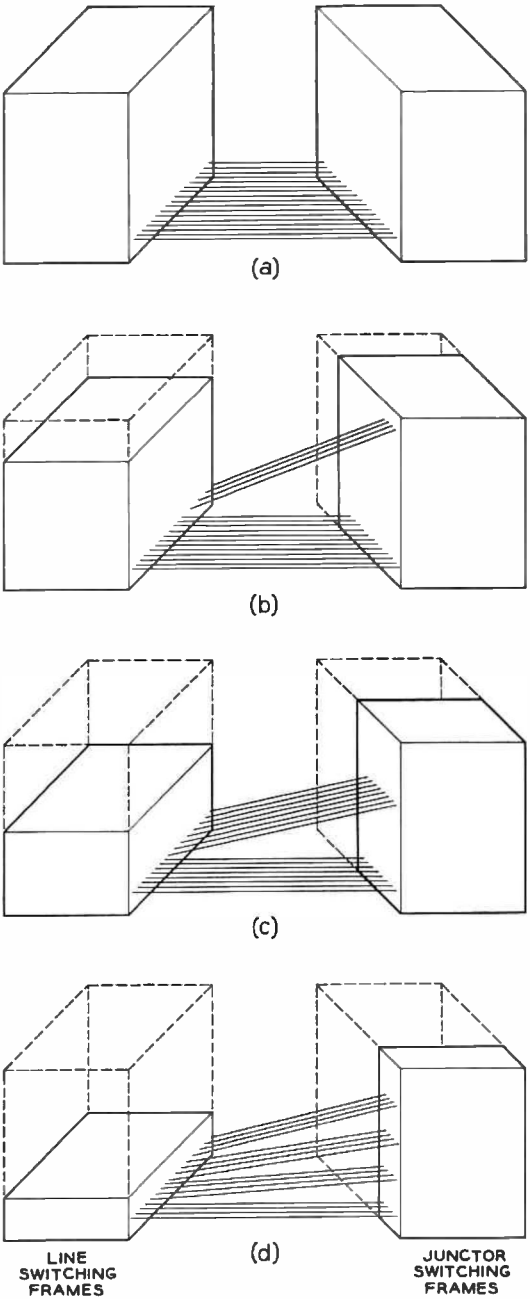


Fig. 8 — Schematic diagram illustrating the partial equipping of line link networks.

can be kept within reason. In the frame design, a compromise has been made between the savings resulting from manufacturing equipment in large units and the excess amount of equipment that may be purchased in each installation because of the necessity of buying integral numbers of frames. As mentioned previously, line switch frames provide terminals for 1024 lines (512 for the 2:1 line concentrator), junctor switch frames for 256 junctors and trunk switch frames for 256 trunks. In order to equip a network with fewer than its full complement of frames, a special wiring plan must be used. Line or trunk switch frames are easily omitted, since they take with them their B link traffic. To omit a junctor switch frame, however, it is necessary to reassign some of the B links which carry traffic from the line or trunk switch frames. Fig. 8 shows in a simplified form based on Fig. 6 how a group of 16 B links from a line concentrator is reassigned for quarter, half, three-quarter and full line link networks. Similar patterns are used for the trunk link networks, although they are somewhat more complicated by the large number of parallel paths. Partial equipping in either link network increases the number of parallel paths while reducing the number of junctors available for connections to other link networks. The two offset each other to a certain extent, so that partial link networks can be traffic loaded almost as efficiently as full link networks.

2.6 *Network Sizes*

Up to now a line link network of 2048 or 4096 lines and a trunk link network of 1024 trunks have been described. If the line and trunk average usages happen to be just right, these link networks carry traffic with high efficiency. If, however, the load per line or trunk is too high, not all terminals can be assigned. On the other hand, if the terminal load is too low, some of the switching equipment is wasted. To reduce the waste in the latter case, wiring patterns have been evolved which provide for up to twice as many line or trunk switch frames as make up the basic network. Thus up to 8192 lines may appear on a line link network and up to 2048 trunks on a trunk link network. This higher concentration ratio from terminal to junctor is achieved by multipling B links at the third stage of the network. Fig. 9 shows the effect on crosspoint requirements in a line link network with an upper traffic bound of 0.4 occupancy on its links. (Cutoff and test access crosspoints are included.) Unfortunately, except in the maximum size, one cannot merely add new links to the basic links without rearrangement. To do so would result in the added equipment being served only by shared links, while the basic equipment would be served by a mixture of shared and private links. The resulting

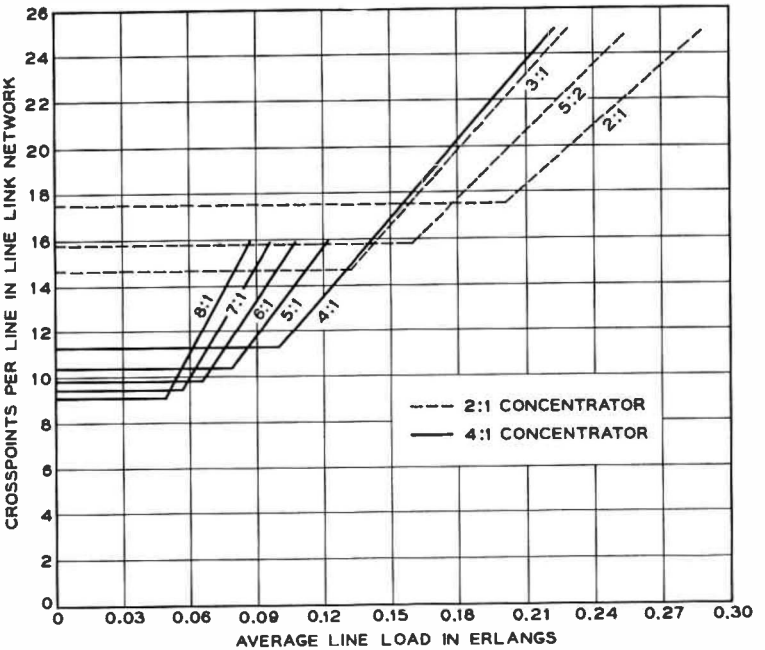


Fig. 9 — Crosspoints per line as a function of line usage when an upper bound of 0.4 occupancy is selected.

uneven service is unacceptable; instead, the patterns are arranged to provide as nearly equal sharing as possible. In choosing the patterns, attention was given to reducing the number of link reassignments that would be necessary if the average load per line or trunk should change significantly. Patterns for partial equipping are unaffected by the choice of concentration ratio.

Central offices with heavy PBX development may have line usage as much as twice the usual average. A blank terminal on a line switch represents not only wasted crosspoints, but a wasted scan point, cutoff contact and main distribution frame appearance. As the demand appeared high enough, thc second design of line switch frame was made in which 512 lines reach 256 B-links. By similar multipling of B-links this frame can fill the gap between average usage and double usage on lines.

2.7 *Junctor Patterns for Growth*

The link network junctor terminals are cabled to plugs and jacks to make possible orderly transitions from one size office to another, both

at the time new equipment is installed and afterwards to take care of changing traffic patterns. Fig. 9 shows how the 64 subgroups of a network are cabled to the junctor grouping frame. A subgroup contains 16 junctors, each junctor from a different octal grid. Half of the subgroups are jack ended and half plug ended — one jack or plug per subgroup. Interconnections are made only within the shelves shown on Fig. 10. Plugs and jacks always interconnect different numbered switches to insure two different sets of C links on connections originating and terminating within a link network.

The junctor grouping frame is also used to insert junctor circuits in

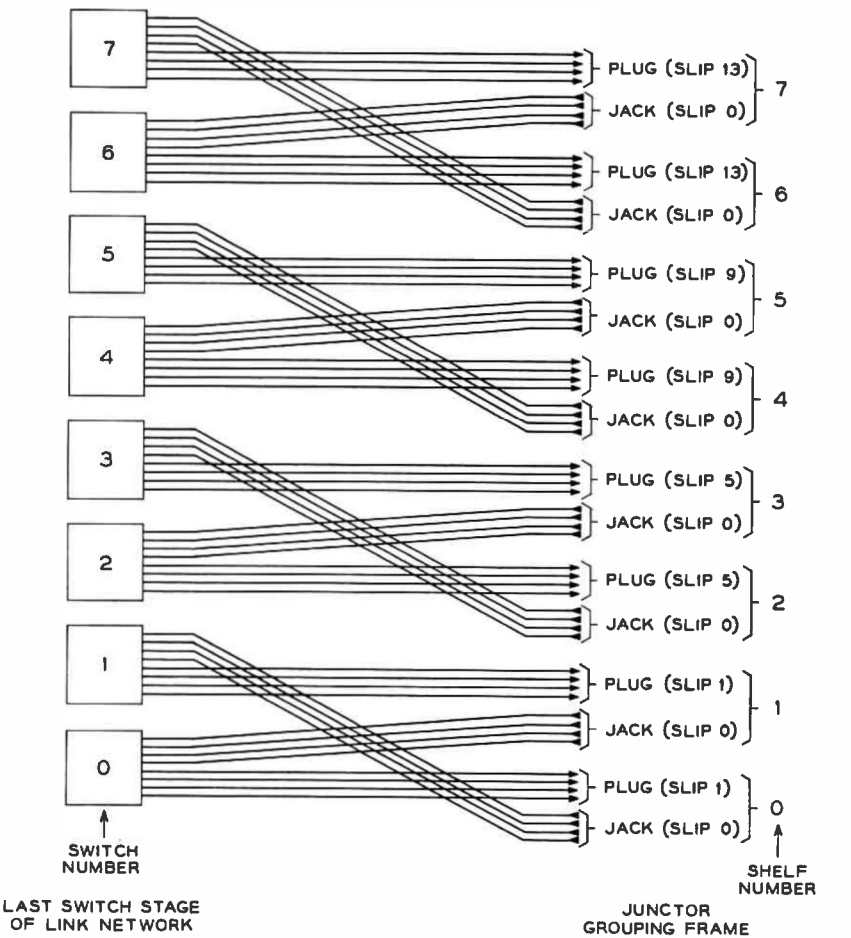


Fig. 10 — Junctor group assignments.

line-to-line junctor subgroups. One side of each circuit terminates on a plug, the other on a jack. Connecting a line-to-line subgroup consists of inserting a line link network plug into a junctor circuit jack, then inserting the corresponding junctor circuit plug into a line link network jack.

It is convenient to use the linear-graph representation of the network in discussing the next aspect of junctor connections. Fig. 11 shows the links available for paths between two lines with a subgroup connected plug-to-jack on shelf one (solid) and shelf three (dashed). The junctors are seen to be "slipped" by one and five terminals, respectively. A slip of at least one is necessary for making a connection between lines assigned to the same concentrator — a slip of zero would provide no path for intraconcentrator calls because the same B link would be required twice on any path. A slip of at least four is necessary for completing calls between lines on the same line switch. The additional choice of paths provided by giving a second set of junctors a slip which differs from the slip of the first by at least 4 gives many more opportunities of finding paths than a simple parallel choice. The second set aligns both the A links and B links of the two ends of the connection in different combinations. As noted in Fig. 10, the plugs are wired for slips of 1, 5, 9 and 13. The definition of an actual slip depends on the point of view. The solid wiring of Fig. 11 gives a slip of 1 when viewed from left to right, but a slip of 15 when viewed from right to left. Thus 8 slips (1, 3, 5, 7, 9, 11, 13, 15) are available in the assignment of junctors — governed by the choice of shelf on which junctors are connected and whether the connection is plug-to-jack or jack-to-plug.

When networks are partially equipped with either one, two or three junctor switch frames, the junctor wiring is changed so that full junctor subgroups still appear on a plug or on a jack. As with the B links, the particular choice of wiring is made with the objective of reducing the number of wires that must be moved as the network grows.

III. NETWORK CONTROL

3.1 *Path Searching*

A basic decision in the design of the switching network was that of isolating completely the path searching function from the switch itself. External control circuits cannot determine directly the states of the crosspoints within a switch. In accordance with the general No. 1 ESS approach, the central processor makes all path searches and keeps a continuous record of all pertinent switching information in its temporary

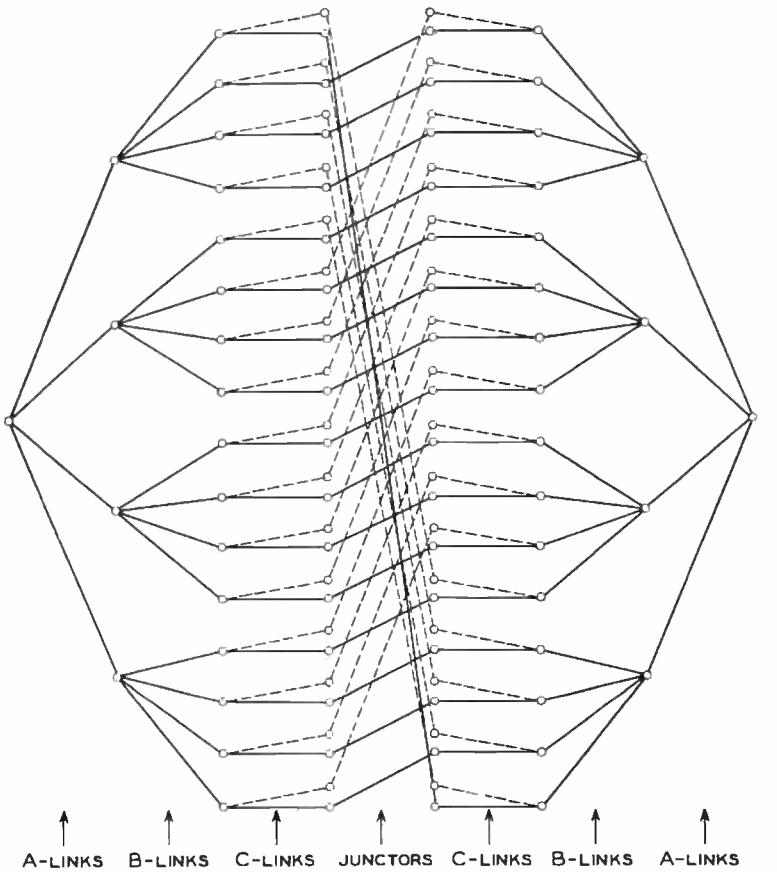


Fig. 11 — Linear graph of line-to-line paths with two junctor subgroups.

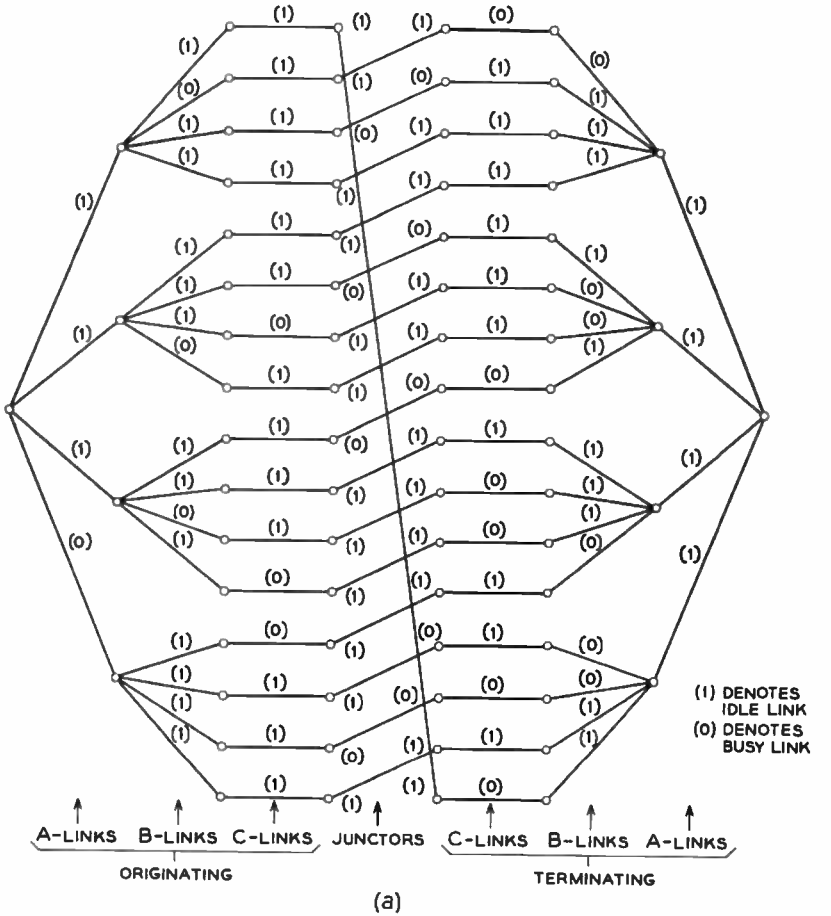
memory, the call store. Programs which use this information, either in setting up paths or in releasing them, must keep the network records up to date. Network records are among the most vital of those kept in call store. Their loss would be equivalent to the loss of power in an electromechanical switching network. The memory reliability has been made high. Beyond this, the network control programs have been designed to keep the chance of error low.

The format of the switching network record in call store was chosen with a view toward low processing time in establishing or releasing network connections. The records are somewhat redundant because of this objective but the redundancy also provides additional insurance against

memory failure. There are two basic records: "link memory" is provided on a basis of one bit for each link and is used in the path searches. A "0" indicates a busy link; a "1" indicates an idle link. "Path memory" is provided on a basis of one word for each junctor terminal in a line link network and one word for each trunk terminal in a trunk link network. It is used to store data necessary for releasing connections.

Let us consider first the link memory and, for example, the problem of finding a path between two lines. This can be divided into searches for a path from a line to a junctor on the terminating and on the originating link networks and then a search for a commonly accessible junctor. Fig. 12(a) gives a graph of a typical situation showing link status. Fig. 12(b) gives the corresponding link words. The 16 lines terminated on a line switch have access to 8 A links. The bits for these links (and for the 8 links in an adjacent switch) are contained in a single link bit word. Through the A links, each line has access to 16 B links. Again the 16 bits corresponding to these links are contained in a single word. Now a path through the network must be set up through an idle A link and an idle B link. By suitable masking and shifting, the bits corresponding to the 4 eligible A links can be extracted, and from them a 16-bit word can be generated with 4 bits for each A-link bit occupying positions corresponding to each of the eligible B links. The resultant 16-bit word can now be combined using the logical AND central control function, with the B-link bit word to produce a matching word in which 1's represent those idle links with free paths to the particular line. Continuing through the network, each of the B links can reach 8 C links. The link bits corresponding to these C links are so arranged that a bit in a C-link word represents one of the C links accessible to a B link (represented by a bit occupying a corresponding position in a B-link word). A C-link word can therefore be combined with the matching word to test 16 paths one stage further into the network. A similar action on a junctor link bit word results in a matching word indicating all available paths from line to a selected subgroup of 16 junctors. Junctor connections between link networks are always made in integral numbers of these junctor subgroups. Different combinations of link and junctor words can be used to match paths from a line to any of the 64 subgroups of a full link network.

After a matching word is determined between originating line and junctor subgroup, a similar word can be derived for the terminating line and the same subgroup. Assuming that idle paths exist in both words, it is now necessary only to take into account the junctor slip. Because of this slip the bit positions in the matching word for one network will not correspond to those in the matching word of another network. One of



(a)

	ORIGINATING	TERMINATING
A-LINK	1110 1101 1110 0110	1011 1111 1110 1111
EXPANDED A-LINK	1111 1111 1111 0000	1111 1111 1111 1111
B-LINK	1011 1110 1101 1111	0111 1001 1110 0011
A-B	1011 1110 1101 0000	0111 1001 1110 0011
C-LINK	1111 1101 1110 0111	0111 1110 1001 1010
A-B-C	1011 1100 1100 0000	0111 1000 1000 0010
JUNCTOR	1101 1011 0111 1101	1011 0110 1111 1011
A-B-C-J	1001 1000 0100 0000	0011 0000 1000 0010
ROTATED ORIGINATING WORD		0011 0000 1000 0001
MATCHING WORD		0011 0000 1000 0000

(b)

Fig. 12 — (a) An example of the states of the links of a line-to-line connection. (b) Searching a path.

the words is "rotated" to line up the path bits and then the two words can be combined to see if any free path exists. If so, the central control order for finding the leftmost 1 can be used for fast identification of the path. If not, a second junctor subgroup must be tested. Because the above method tests 16 paths at a time and because there is a reasonably high chance of success on the first junctor subgroup chosen, searching time is relatively low. Other concentration ratios and partially equipped link networks require variations on the steps given in this example.

A further complication, not shown above, is the necessity of reusing links which have been in use on a previous section of a call or which are being reserved for an anticipated connection. Thus, in the example, at least the A and B links which were used in the dialing connection to the originating line should be available for the line-to-line connection used. Similarly, the A and B links reserved for the line-to-line connection should be available for ringing connections. Failure to make these links available would drastically reduce network capacity.

Link bits contain sufficient information to hunt an idle path between two network terminals. They do not, however, contain enough information to identify a path for releasing a connection, because they do not indicate which link is connected to which. This information is stored separately in blocks of call store named "path memory words." Line path memory words are assigned in blocks of 1024, with each word corresponding to a junctor terminal of a line link network. Trunk path memory words are assigned in blocks of 256, with each word corresponding to a trunk terminal of a trunk switch frame. A line path memory word contains the identity of the line connected to the associated junctor, while a trunk path memory word contains the identity of the junctor connected to the associated trunk. Additional bits in these words identify the path used when more than one path is possible between the junctor and line or trunk.

The correspondence between path memory word and network terminal was chosen to simplify most of the programs which change link bits when a path is released. Thus when a trunk indicates that a network path is no longer needed, translation from trunk identity to network termination serves also to locate the trunk path memory word. This in turn identifies the line-to-trunk junctor number which serves to locate the line path memory word. Line-to-line connections are traced in a similar fashion, but additional information is needed on trunk-to-trunk connections because only half of the path can be traced from the trunk path memory word, and there is no path memory associated with trunk

junctor terminations. On trunk-to-trunk connections, therefore, additional path memory is provided in a register associated with the call.

3.2 *Network Actions*

Searching for a path is only part of the network control job. Orders must be issued to the switch frame controllers to close the specified crosspoints, to signal distributors to open or close relays in trunks, and to scanners to verify that all orders have been properly executed. Because the time restrictions on the network are given in tens or hundreds of milliseconds, relays have been used in both switch frame and signal distributor controllers. To match the high-speed central control with the slower-speed relay circuitry, a cyclic method of controller operation has been adopted. Orders are issued in batches at about 25-millisecond intervals. The program which sends out orders keeps its own record of those controllers it has called and will not send two orders to the same controller in one batch. This record is kept in call store, where it can be interrogated at central control speed without waiting for the slow response time of the controllers themselves.

A new batch of orders is sent when four full five-millisecond intervals have passed after the last order of the previous batch; since the interval in which the batch was sent cannot be counted, the expected time between batches is 25 milliseconds. To meet this method of timing, the controllers must finish their work in under 20 milliseconds. At extreme traffic loads, the time taken to issue the orders, when added to other essential work, may stretch many cycles out to 30 milliseconds.

Network actions are controlled in two parts. The first program prepares a list which is placed in a call store area named a "peripheral order buffer." This list contains instructions for the proper issuing of controller orders and is held until the connection is set up. The second program scans the controllers at the start of a network cycle to check that all are ready to receive new orders. It then works through the list, sending out as many orders as it can. When it reaches a point at which it cannot proceed further, either because of a planned delay or because it finds an order to a controller that has already received an order in that cycle, it proceeds to the other buffers which have waiting work. If it finishes a list, it reports back to the call processing program which had requested the connection.

Consider, for example, the task of setting up a path between an incoming trunk and a line. At the time this connection is to be made, the calling trunk is connected to a ringing tone trunk and the called line is

connected to a ringing trunk. The functions to be performed are, in order:

(1) open cut-through relay in ringing trunk*
(2) open cut-through relay in ringing tone trunk*
(3) open cut-through relay in incoming trunk
(4) delay for one network cycle to allow relays to open
(5) close second-stage crosspoint in line concentrator
(6) close two crosspoints in trunk switch frame
(7) close two crosspoints in trunk junctor switch frame
(8) close two crosspoints and test access crosspoint in line junctor switch frame; make false cross and ground test
(9) wait one cycle for test to be completed
(10) open test access point in line junctor switch frame
(11) close last crosspoint in line concentrator
(12) wait one cycle for crosspoints to act
(13) close cut-through relay in incoming trunk
(14) wait two cycles for relay action and for transient decay
(15) scan for line current at the incoming trunk to verify continuity of connection
(16) report back to originating program.

The network control program places the list of orders for this sequence in a peripheral order buffer. At 25-millisecond intervals the list processing program will work as far through this list as it can get. Assuming no delays from busy controllers, the first delay will occur at step (4). A pointer will be left at step (5) so that on the next cycle the list processing can be taken up again through step (9), and so forth, until the path is verified.

IV. TRAFFIC PERFORMANCE

As indicated earlier, the large number of meshed paths in the network makes possible a high efficiency of the network links for objective levels of blocking. In actual use this high efficiency must be weighed carefully, since, as with most traffic carrying facilities, an increase in efficiency at a given service point results in more rapid decline in service at loads beyond this point. The evaluation of a network design includes an examination of a range of loads to make sure that operating points can be found with sufficient range of good service. The final choice of the operating point is an administrative responsibility of the operating telephone company.

* These are the only relay actions required to release the network path. The other parts of the path will be used as needed without interference because of the differential excitation of the ferreed switch.[3,4]

The performance of this network is conveniently broken into two parts, that of the line concentrator by itself and that of the full network in its various uses.

4.1 *Line Concentrator Characteristics*

Fig. 13 shows a load-service curve for the concentrator based on simulation results. This simulation was applied to the concentrator alone; it consisted of repeated offerings of simulated calls to a concentrator modeled in a general-purpose digital computer. The data for Fig. 13 were collected for a concentrator with all lines assumed to have equal calling rates and usage, and are compared with data for a similar theoretical group of 40 lines with full access to 10 links. While this kind of curve is of interest for general evaluation, more concern is felt about the situations which will be found in the field, where equal line usage is the exception rather than the rule. In particular, concern is felt for situations where, through chance, several high-usage lines are assigned to the same switch. With purely random assignments this chance can be shown to be small — it can be made much smaller by adopting assignment practices which tend to spread lines known to be high-usage (such

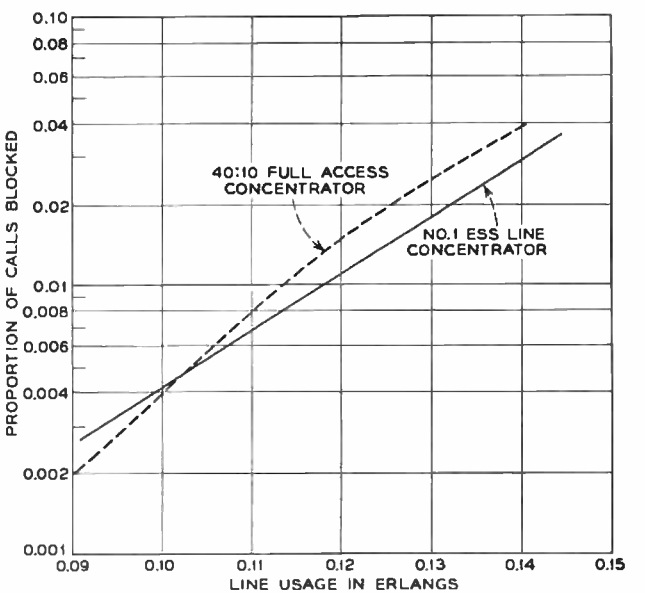


Fig. 13 — Load-service curve of 4:1 concentrator, determined by simulation, compared with theoretical load-service curve for full-access group of 40 lines on 10 links.

as PBX lines) over the concentrator switches. For further simulation, line usages were assigned at random from an exponential distribution. Fig. 14 shows the results of a typical run under these conditions. The average blocking (measured as the number of blocked attempts divided by the total number of attempts) is materially less than the blocking of Fig. 13. On the other hand, the 16 lines on the switch with the highest load experienced a slightly higher blocking.

The drop in average blocking with a wide spread of line usage can be explained as the double effect of the narrowing of the load variance accompanying a variation in source loads and a reserved path effect whereby a high calling rate line may place a new call over the path it has just abandoned before a call from another lower calling rate line can seize it. The average load within the groups of 16 lines is not as closely correlated to the service encountered by the group as might be expected. Closer examination of the groups indicates that the spread of line loads within the group itself is a strong factor. Allowing for the difficulty of predicting the exact performance of each concentrator, it appears safe to use the equal-usage curve of Fig. 13 as a basis for deriving engineering procedures.

In addition to the blocking of the concentrator, the delay given to those originating calls which are blocked is of prime importance. Because new trials are made through the rest of the network to find customer dial pulse receivers, the main network source of this delay is in the con-

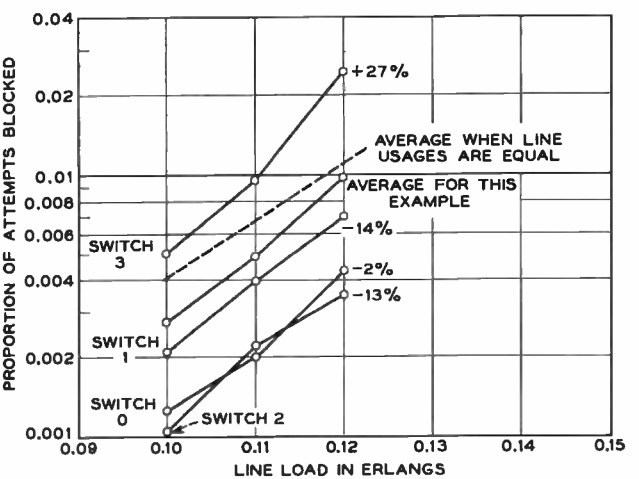


Fig. 14 — Example of effect of switch unbalance on 4:1 line concentrator.

centrator itself. At the completion of dialing, the path through the concentrator which was used for dialing is available for reuse on the ringing and talking connections. In order to evaluate the delay generated by the concentrator, the simulation program was modified to treat all calls as originating calls and to hold delayed calls until they were served. This made possible a direct comparison with the theoretical performance of a full-access group.[6] Fig. 15 shows the results of a simulation run with 0.15 erlang per line, which resulted in slightly over 0.04 of the calls being blocked. The distribution is shown only for the calls which were blocked and is compared to similar theoretical data with a full-access group of 40 lines reaching 10 links. In both cases it is apparent that these groups must be run normally at low losses because dial tone delays, when they occur from this cause, are long. The No. 1 ESS line concentrator with 6 crosspoints per line compares well with the full-access concentrator with 10 crosspoints per line.

The 2:1 concentrator has not been studied in as much detail as the 4:1 concentrator. Not only will it have much lower blocking at comparable link loads because of only second-stage concentration, but the smaller

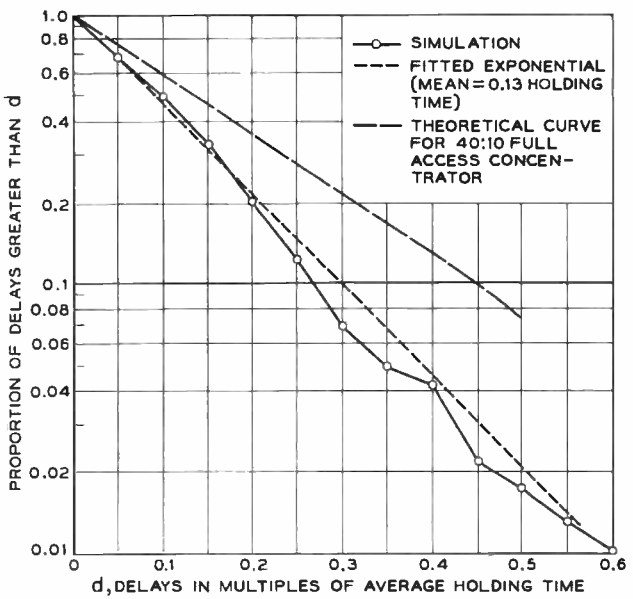


Fig. 15 — Delay distribution of blocked calls, determined by simulation at 0.15 erlang per line, compared with theoretical delay distribution for full-access group of 40 lines on 10 links.

number of lines will introduce a stronger limited-source effect. Under these conditions the network as a whole will be the main source of congestion, and loading of the 2:1 concentrator is not a serious factor.

Because of the delay characteristics of the concentrator, it is expected that the concentrator will be used at low blocking probabilities. If a point such as 0.10 occupancy of the lines is chosen, increases in load of 40 per cent are seen to produce a significant increase in blocking, but this is minor compared to what will happen in the rest of the switching network and interoffice trunks if such high traffic overload occurs.

Fig. 16 shows an estimate of the performance of the switching network

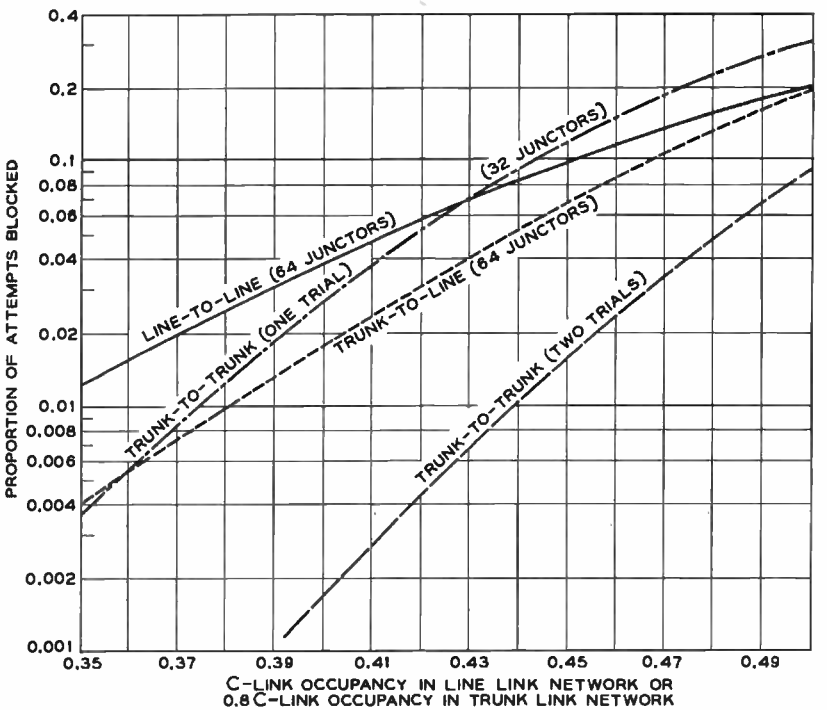


Fig. 16 — Load-service curves of switching network.

(including blocking contributed by the line concentrators). Trunk networks and junctors terminating on trunk networks were assumed to be loaded 20 per cent above the line link networks. All of these data were based on NEASIM* simulations which in turn were verified by a small

* NEASIM[7] is the name given to a computer program designed in Bell Telephone Laboratories. It uses a technique of simulating link states of a linear graph.

number of large simulations by digital computer, in which the network was fully represented. For a large network these full-scale simulations are expensive and generate little data not available by NEASIM techniques.

In actual practice, of course, varying numbers of junctors will be available between link networks; higher efficiencies are available with larger groups. In such cases, the line concentrator may well be the limiting item in loading line link networks of a small office and the intraoffice junctor group the limiting item in a large office. The latter limitation can be overcome by providing intraoffice trunks on the trunk link networks and letting them carry the traffic which overflows small groups of high-efficiency line-to-line junctors.

As with other networks in the past, it is to be expected that initial installations of the network will be over-provided with switches to insure good service until operating experience gives additional data for more precise traffic engineering. Should either the estimates of office traffic or the estimates of service characteristics be significantly in error, the flexibility of junctor assignments and the ability to change the concentration ratio of the link networks offer insurance that efficient operating points can be found.

V. ACKNOWLEDGMENTS

Mr. R. F. Grantges proposed the NEASIM program, which has been of great importance in evaluating network patterns. He has also made frequent contributions to the network plan. Many others have also contributed, in particular, T. N. Lowry, J. G. Kappel, W. C. Jones, C. Klingman and R. A. Rosenthal.

APPENDIX

Definition of Terms

Switching network — that part of a switching system that establishes transmission paths between pairs of terminals.

Space division (separation) switching network — a switching network in which the transmission paths are physically distinct.

Crosspoint — a two-state switching device, possessing a low transmission impedance in one state and a very high one in the other.

Switch — a rectangular array of crosspoints in which one side of the crosspoints is multiplied in rows and the other in columns.

Stage — those switches in a switching network which have identical, parallel functions.

Grid — a two-stage switching network in which a single path exists between every first-stage switch to every second-stage switch. The number of outputs in each first stage must equal the number of second-stage switches; the number of inputs in each second-stage switch must equal the number of first-stage switches.

Link — the connection between terminals on one switch and terminals on a switch in the next stage corresponding to a single transmission path.

Junctor — any link between central stages of the network.

Concentration — the function usually associated with the first stages of a switching network and characterized by configurations possessing fewer output than input terminals; provided to improve network efficiency when the input terminals carry a light traffic load.

Expansion — the inverse of concentration.

Access — a term indicative of existence of paths within a network configuration from an input terminal to a set of output terminals in absence of traffic; partial access refers to the ability of reaching only a fraction of the output terminals; full access permits reaching all terminals by unique paths; multiple access allows reaching all output terminals in more than one way.

Graph — a graphical representation of all possible paths between two network terminals.

Blocking — inability to interconnect two idle network terminals because some of the applicable links are used for other connections.

Erlang — the traffic unit corresponding to an average of one call present on a traffic carrying facility.

Occupancy — average proportion of time that a traffic carrying facility is busy.

REFERENCES

1. Ferguson, J. G., Grutzner, W. E., Koehler, D. C., Skinner, R. S., Skubiak, M. T., and Wetherell, D. H., No. 1 ESS Apparatus and Equipment, B.S.T.J., this issue, p. 2355.
2. Danielson, D., Dunlap, K. S., and Hofmann, H. R., No. 1 ESS Switching Frames and Circuits, B.S.T.J., this issue, p. 2221.
3. Feiner, A., Lovell, C. A., Lowry, T. N., and Ridinger, P. G., The Ferreed — A New Switching Device, B.S.T.J., **39**, Jan., 1960, p. 1.
4. Feiner, A., The Ferreed, B.S.T.J., **43**, Jan., 1964, p. 1.
5. Joel, A. E., Jr., An Experimental Switching System Using New Electronic Techniques, B.S.T.J., **37**, Sept., 1958, p. 1091.
6. Descloux, A., *Delay Tables for Finite- and Infinite-Source Systems*, McGraw-Hill, New York, 1962.
7. Grantges, R. F., and Sinowitz, N. R., NEASIM: A General-Purpose Computer Simulation Program for Load-Loss Analysis of Multistage Central Office Switching Networks, B.S.T.J., **43**, May, 1964, p. 965.

No. 1 ESS Switching Network Frames and Circuits

By D. DANIELSEN, K. S. DUNLAP and H. R. HOFMANN

(Manuscript received January 10, 1964)

The switching network of No. 1 ESS is a space-separated network employing ferreed switches. The switches are organized into a few basic building blocks or frames from which all network configurations are realized. The frames are autonomous in operation and demonstrate the peripheral bus concept employed in No. 1 ESS.

The presence of the high-speed digital processing center in the central office has had a marked effect upon the circuits of the switching network. Path hunting has become a central control task, which has freed the network of its traditional path memory functions.

A combination of electronic and electromechanical devices is used to control the ferreed switches. Control duplication assures reliability.

1. INTRODUCTION

The switching network is a major functional unit in a telephone central office, representing in many cases 50 per cent or more of the equipment in the office. The choice of apparatus and general circuit design is highly influenced by the search for the optimum balance between cost, reliability, and maintainability.

The switching network serves the function of interconnecting the many lines and trunks served by the telephone office. In addition, the network interconnects the lines and trunks with the many service circuits and tone sources, such as signal transmitters, signal receivers, party test circuits, coin supervisory circuit, ringing circuit, etc.

A second function of the network is to match the traffic between a large number of lightly used lines and a much smaller number of heavily used trunks, maintaining a satisfactorily low probability of blocking.

These basic functions are performed in an operating environment that slowly changes over the life of the office. Not only can the number

of lines and trunks served by the office change, but the occupancy or offered traffic per line can change, the call-routing patterns within the office can change, and the types of facilities to be interconnected can change. The network, in order to meet the requirements of an office at any time and to match the requirements of initial offices that cover a broad range of sizes and traffic characteristics, must be capable of very flexible interconnecting patterns. This can be facilitated by the use of a minimum number of basic network building blocks whose sizes are tailored to acceptable growth or change patterns and which are then interconnected in series-parallel arrangements to form the complete office network.

A further requirement of a general-purpose network is that it be compatible with present outside plant and not be too restrictive on the future development of that plant. The ideal in this regard is a network that does not reflect its requirements on the outside plant. One method of achieving this is to have a network that is transparent to a wide range of transmission and service signals, passing a wide frequency band and tolerating large signal amplitudes. Also, the network should present a noise-free interconnecting path.

The new element in network design considerations is the presence of a high-speed digital data processing center containing bulk memories. The presence of this processing center permits network designs that are faster acting, less expensive, more flexible and more easily maintained than former network designs. Certain functions formerly delegated to the network have been transferred to central control. Chief among these are the link busy-idle record and the calls-in-progress record. These changes make path hunting a central control task that can be integrated into normal call processing and can be done in microseconds instead of at a much slower rate as a network task. In addition, access requirements into the network are lessened and crosspoint costs are lowered. The change is also beneficial from the operational standpoint: network paths can be reserved and links can be made busy for call routing and for network diagnostic purposes without network equipment action.

The invention of the ferreed switch during the electronic switching development program made it possible for the switching network to be highly compatible both with existing outside plant and with the digital processing equipment. In addition, since the ferreed switches could be packaged in units of almost any size without undue equipment or electrical circuit restrictions, the sizes of the network building blocks could be determined directly by traffic and office growth considerations.

II. NETWORK FRAMES

With each new switching system, a number of new terms are adopted to fit its internal organization and its new type of switching apparatus.

The two largest functional groups in the No. 1 ESS network are the line link network (LLN) and the trunk link network (TLN). Figs. 1 and 2 show these two link networks and indicate the link wiring pattern. The same two figures show the breakdown of the link networks into frames, concentrators, grids and ferreed switches.*

A connection through the network consists of several path segments; each path segment is controlled by a separate network frame (see Fig. 3). Although a No. 1 ESS office network may have 300 or more network frames, these are of just three basic types.[1] They are the line switching frame, the junctor switching frame and the trunk switching frame. These frames are independent functional units and are highly autonomous in their operation. They receive all operating instructions in coded form over a peripheral bus[2] system directly from central control and return information to it at the completion of their operating cycles.

The interconnection of the frames to form functional groups and the further interconnection of these groups to form complete networks are treated elsewhere in this issue.[3] This article examines the three types of network frames, including descriptions of the network fabric in each frame, path selection within the network fabric, and the processing of frame input information to achieve a desired action.

III. LINE SWITCHING FRAMES

3.1 *Frames*

The line switching frames perform the first two stages of switching in the interconnecting process: connecting lines on their inputs to B links on their outputs. The line switching frames also perform the second of the two basic network functions, namely, traffic concentration.

To cover the range of No. 1 ESS offices, two designs of line switch frame are provided. One concentrates traffic by a ratio of 4-to-1 between input lines and B links and the second concentrates traffic by a ratio of 2-to-1. In any office, however, only one of the two types of frame is used.

The 4-to-1 line switching frame contains 16 line concentrators, each of which interconnects 64 lines to 16 B links through two stages of

* See Ref. 3 for a definition of all terms.

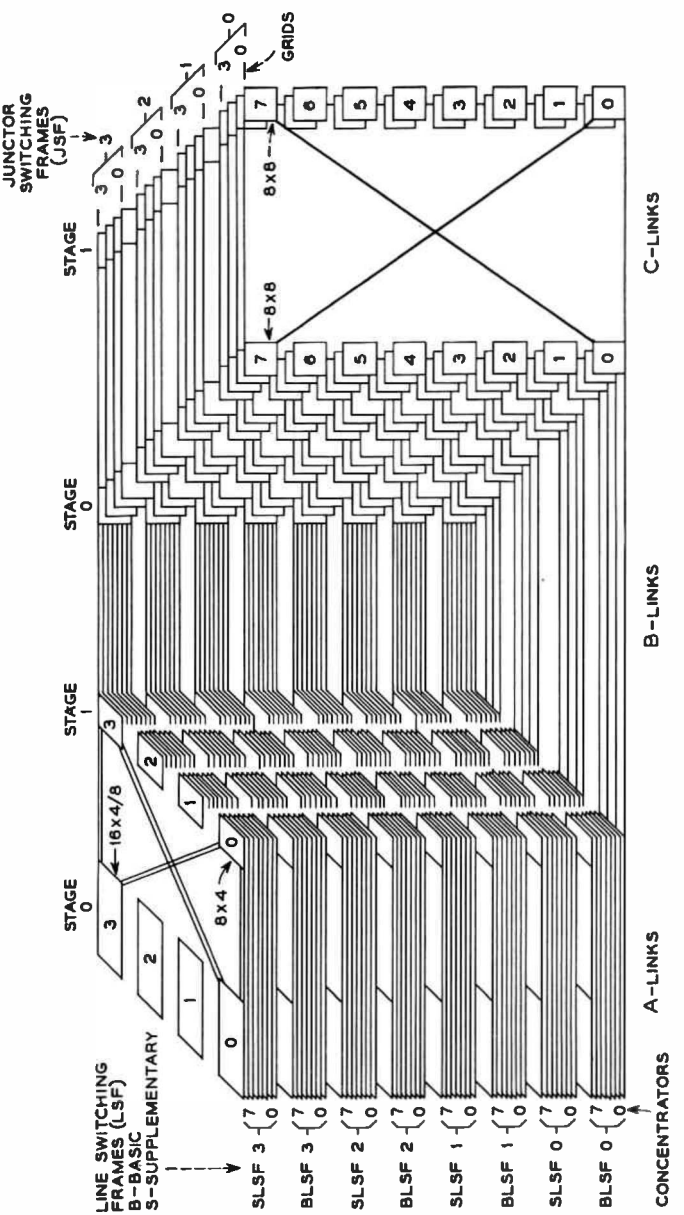


Fig. 1 — Wiring pattern for line link network with 4:1 concentration ratio.

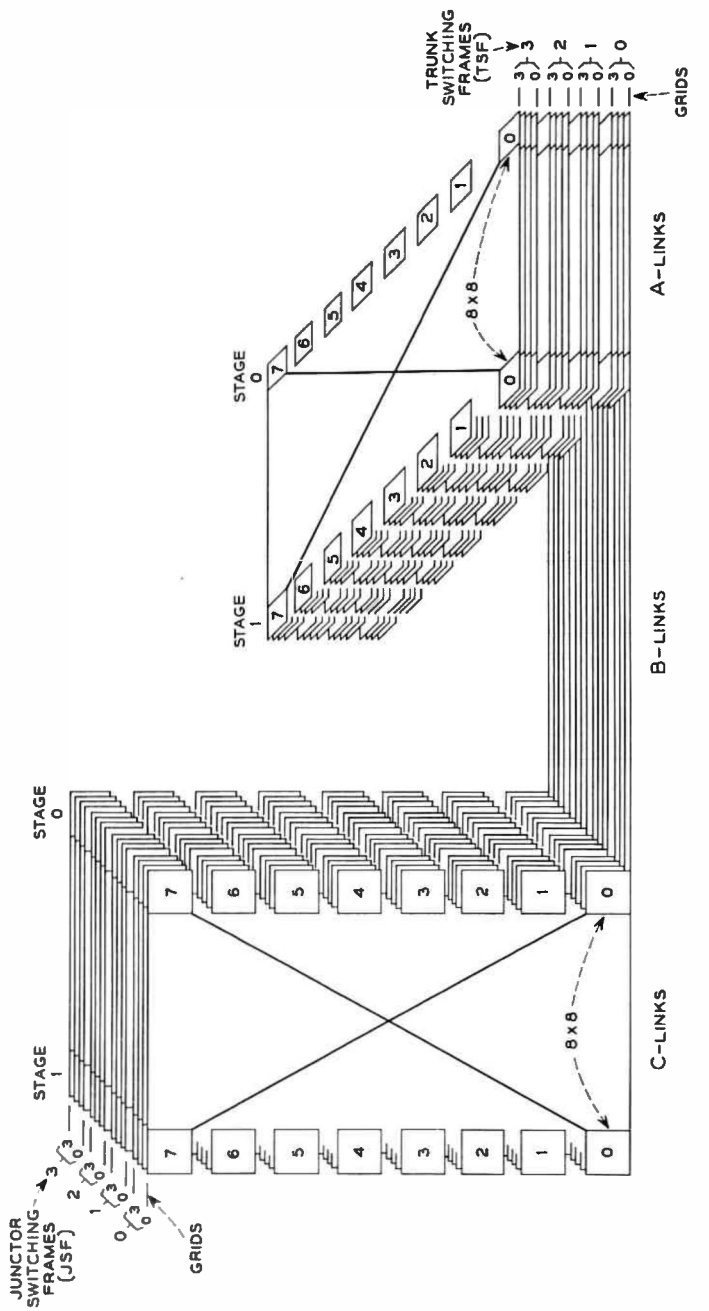


Fig. 2 — Typical wiring pattern for trunk link network with 1:1 concentration ratio.

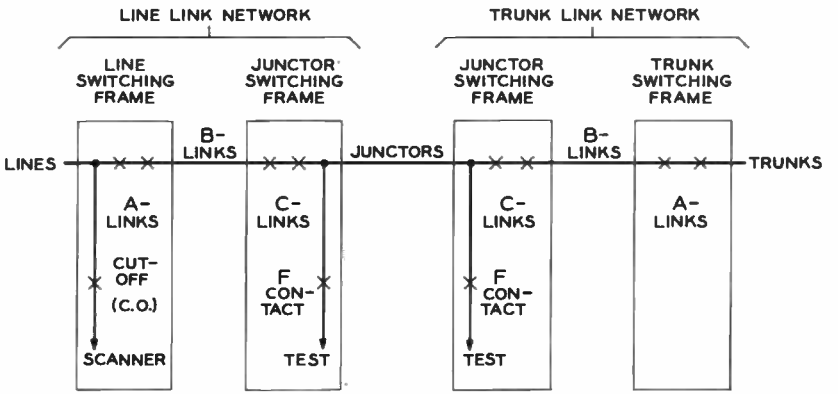


Fig. 3 — Network connections.

switching. The complete frame thus terminates 1,024 lines on its input and 256 B links on its output. For growth purposes, the frame can be separated mechanically into two halves, each containing eight concentrators. The basic half, however, contains all the controlling circuitry needed to control both the basic and supplementary halves. Only one line switch frame in an office would normally operate as a half frame. This half-frame operation provides line growth in steps of 512 lines. The 2-to-1 line switching frame contains 16 line concentrators, each of which interconnects 32 lines to 16 B links through two stages of switching. The frame thus terminates 512 lines on its input and 256 B links on its output. The frame is always supplied as a complete unit.

3.2 *Concentrators and Switches*

The 16 concentrators on a line switching frame are identical. A block diagram of a 4-to-1 concentrator is shown in Fig. 4. The 64 lines are assigned to four input or stage 0 switches, and the 16 output or B links are assigned to four stage 1 switches. The wiring between stages, called "A links," permits each input terminal to have access to each output terminal over a single path defined by a particular A link. Each of the 16 inputs on a stage 0 switch has access to only four of the eight outputs of the switch. The four outputs, however, in every case form a set that is wired one each to the four switches in stage 1. Thus full access of B-link outputs is achieved with a minimum crosspoint count per line.

Each input terminal also connects to a pair of contacts that give the line access to the line scanner for service request indication. These cut-

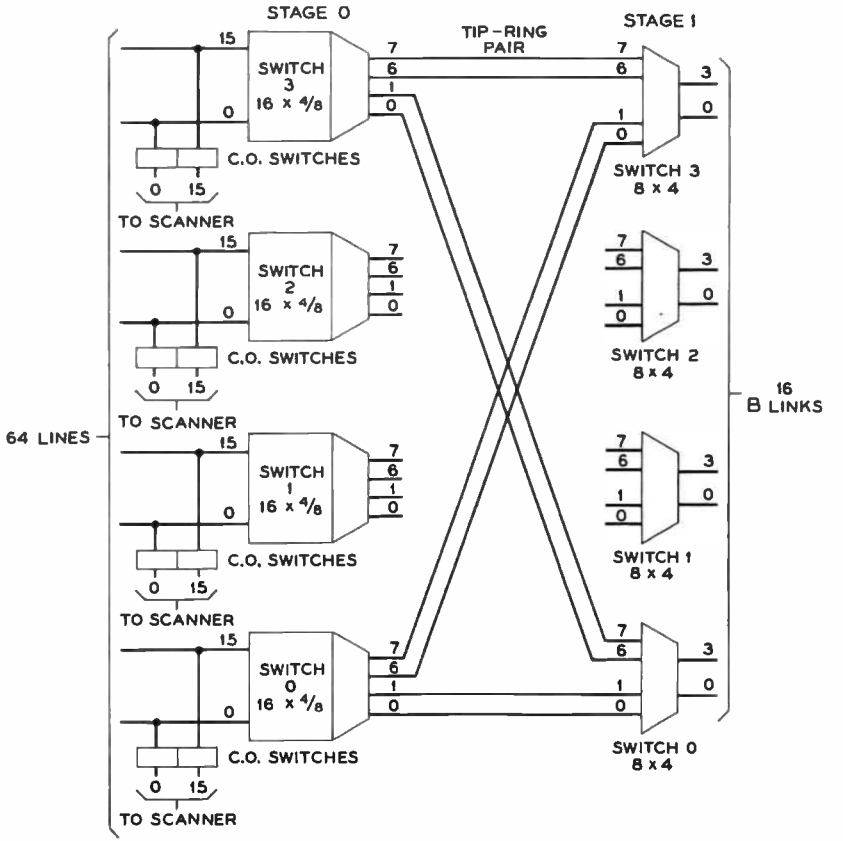


Fig. 4 — Line switching frame: tip-ring block diagram of a 4:1 concentrator.

off ferreed switches, bipolar in operation, are controlled by the line switch frame control equipment.

A block diagram of a 2-to-1 concentrator is shown in Fig. 5. This concentrator differs from the 4-to-1 concentrator only in the stage 0 switch. The 32 input lines are assigned to eight 4 × 4 switches. Full access to the B links is achieved with traffic concentration limited entirely to the stage 1 switches.

IV. TRUNK AND JUNCTOR SWITCHING FRAMES

4.1 *Frames*

The trunk switching frames perform two stages of switching in the interconnecting process, connecting trunks and service circuits on their

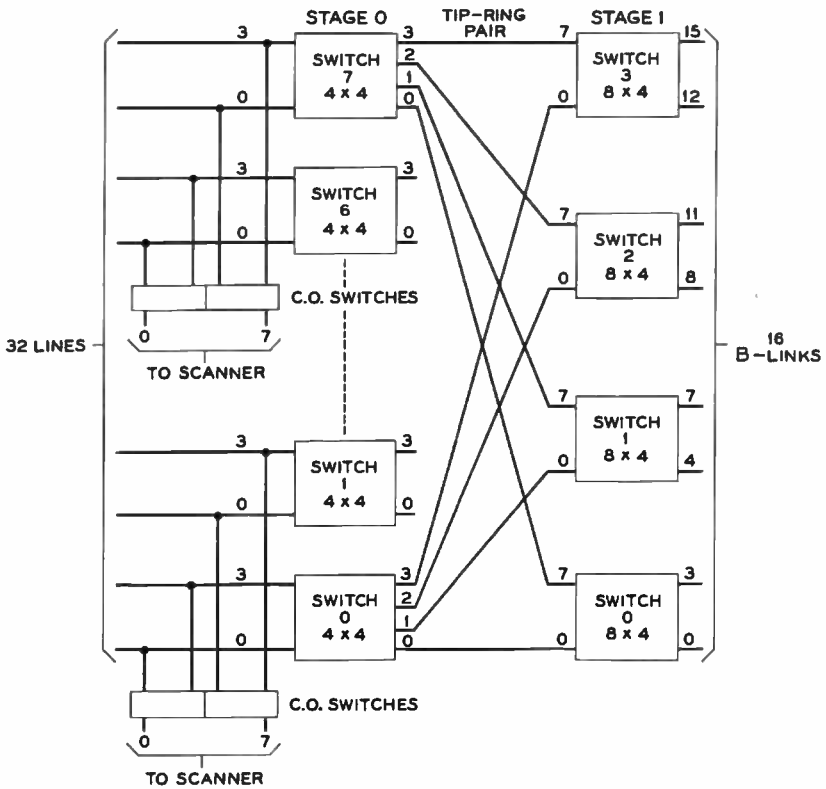


Fig. 5 — Line switching frame: tip-ring block diagram of a 2:1 concentrator.

inputs with B links on outputs. The junctor switching frames perform two stages of switching, interconnecting B links and junctors, the latter being the central link of every network connection. Both these interconnecting links are completely internal to the network connection.

Both types of frames contain four octal grids, each of which interconnects 64 inputs to 64 outputs through two stages of switching. The complete frames are always provided. This provides growth steps of 256 trunks and 256 junctors.

The junctors switching frame, in addition to the above switching grids, contains a pair of contacts per junctor that give the junctor access to a common test vertical. The test vertical ferreed switches, bipolar in operation, are controlled by the junctor switching frame control equipment.

4.2 *Octal Grids and Switches*

An octal grid with test vertical access is shown in Fig. 6. The 64 inputs are assigned to eight input or stage 0 switches, and the 64 outputs are assigned to eight stage 1 switches. The interstage wiring connects each input switch to each of the eight output switches. Each input terminal thus has access to each of the 64 output terminals through a single path defined by a particular link between switches.

V. PATH SELECTION

For path selection purposes, the ferreed switches within each frame are divided into two equal groups, and independent path selection

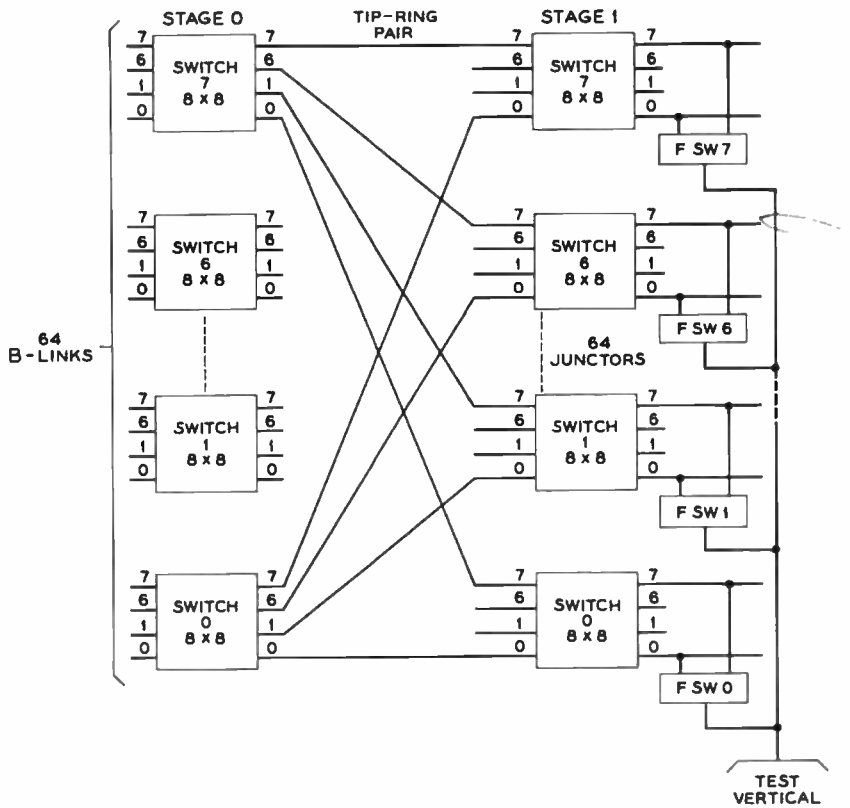


Fig. 6 — Junctor switching frame: tip-ring block diagram of a grid with test vertical access.

facilities are provided for each group. Two simultaneous paths can therefore be established per frame, one per switch group.

The magnetically latched ferreed crosspoints[4,5] are wired into coordinate arrays to form 8 × 8, 8 × 4, 4 × 4 and partial-access 16 × 8 switches.

The control windings are series connected along the rows and columns of the switch. One end of the control windings of all rows and columns is connected to a common multiple within the switch. By pulsing one row and one column via the common multiple, the crosspoint at the intersection is closed. At the same time, any crosspoints previously closed along the pulsed row or column are released. This operating method eliminates the need for specific release operations on the crosspoint. Crosspoints are released as a direct result of the operation of other crosspoints.

In all frames, the control winding interconnection pattern between switches on a frame parallels exactly the interconnection pattern of the tip-ring conductors on the frame.

The closure of a tip-ring transmission path between specific pairs of input and output terminals requires the momentary selective closure and high-current pulsing of the control winding path between these terminals. The pulse path is selected by means of 24-contact wire spring relays. Relay contacts are inserted in the control winding paths of every input, intermediate and output link. Other contacts steer the pulse to general switch groups and control the polarity of the pulse for special purposes. One contact per relay is used for test purposes.

In the junctor and trunk switching frames, the path selecting groups internal to the grid are four groups of eight wire spring relays each. The four relay groups define respectively the input level, the input switch, the output level and the output switch. This is shown in Figs. 7 and 8. The operation of one relay in each group defines a unique pulse path through the grid. Other contacts on these same relays are inserted into the links of the second grid of the selection pair. The function of the relay groups, however, is not always preserved in crossing between grids. This allows the maximum usage of the 24 contacts on each relay.

Within the grid, input switch selection has been made in the intermediate switch links. This performs the added function of disjoining the internal link multiple that results from the control winding multiple within this switch.

In the junctor switching frame, additional contacts are used within the grid to select the bipolar test vertical access ferreed F as shown in Fig. 8. This requires the use of additional contacts on the relay groups

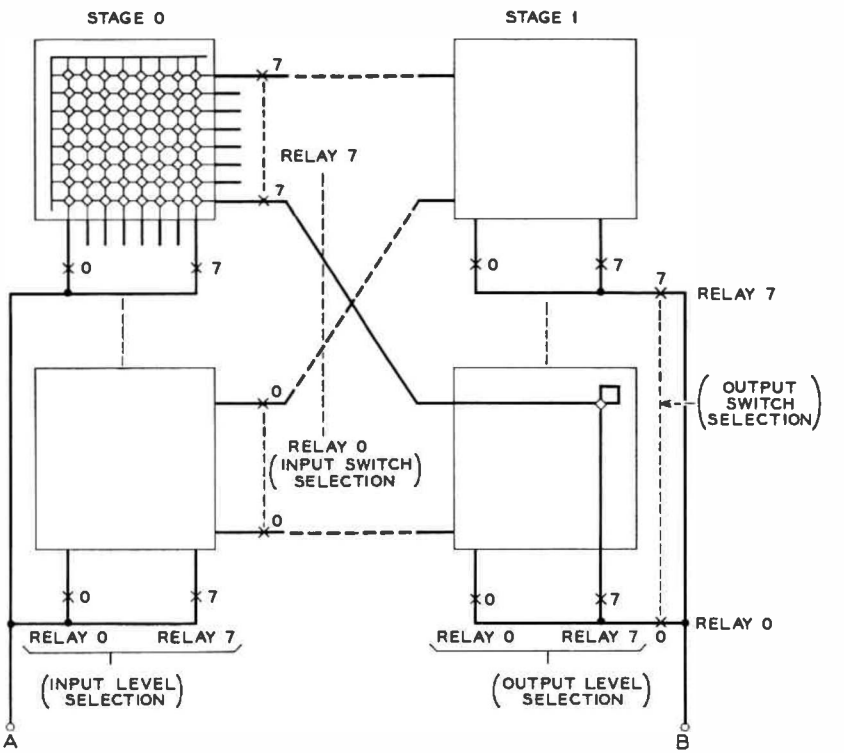


Fig. 7 — Trunk switching frame: path selection within a grid.

that select the output switch and output level. The grid with test vertical access has three possible pulsing points: A, B, and C. By pulsing between terminals A and C with one polarity, the two crosspoints in the grid and the F contact are closed. By reversing the pulse direction between these two points, the two crosspoints in the grid are closed and the F contact is opened. By connecting the pulser between terminals B and C in either of the two polarities, the F contact can be opened or closed without pulsing the transmission crosspoints. Similarly, by pulsing between terminals A and B, the transmission crosspoints can be operated without changing the state of the test vertical access ferreed.

External to the grid, the pulse is steered by means of additional wire spring relay groups as shown in Fig. 9. The first selection group steers the pulse to its normal pair of grids or the grids of the alternative half of the frame. The next relay group, the order group, controls the

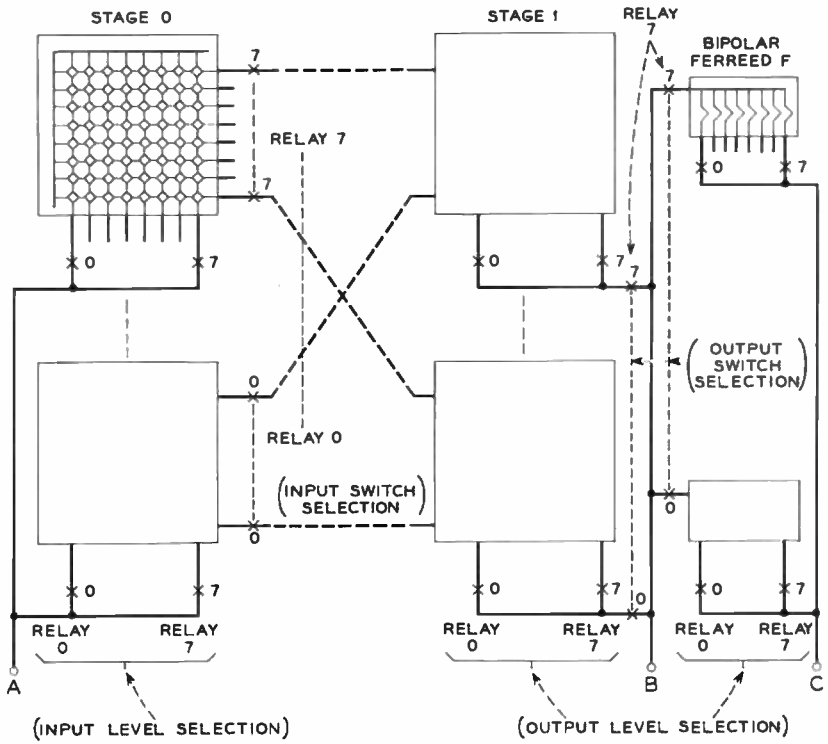


Fig. 8 — Junctor switching frame: path selection in a grid with test vertical access.

grid pulse points and pulse polarity. The third relay group controls the grid selection.

Path selection in the 4:1 concentration ratio line switching frame follows the same general pattern as in the junctor and trunk switching frames. The eight concentrators on the basic frame form one path selection group, and the eight concentrators on the supplementary frame form the second selection group. Within a concentrator, three relay groups perform path selection. These relay groups consist of 16 relays each. As shown in Fig. 10, one group defines the 16 input levels per input switch, the second, the 16 output levels per concentrator, and the third is a make-up group. This group is not too closely defined by equipment layout but essentially chooses input half-switches for a pair of concentrators.

Path selection in the 2:1 concentration ratio line switching frame is similar to the 4:1 concentration ratio line switching frame. Three relay

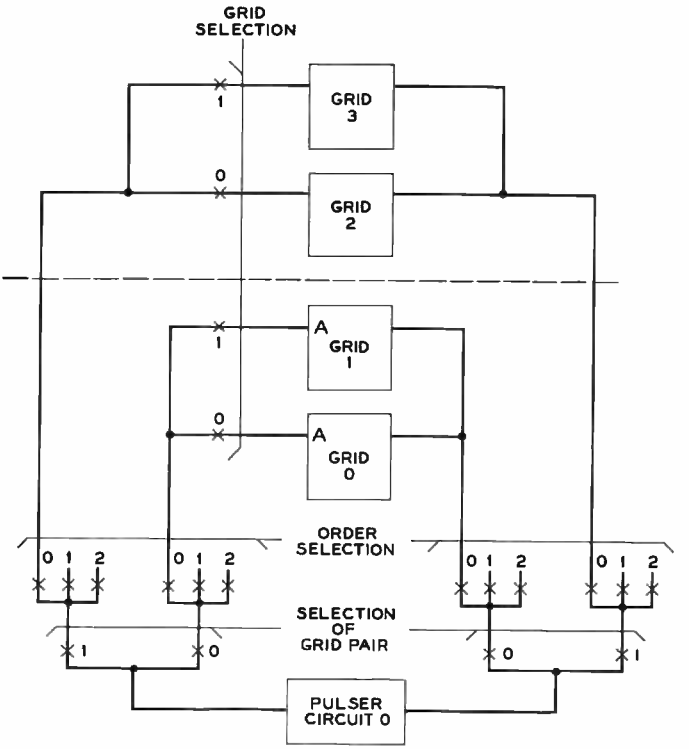


Fig. 9 — Path selection external to grids.

groups of 16 relays each are used for input level, input switch and output level selection as shown in Fig. 11.

In the 4:1 concentration ratio line switching frame selection plan, maximum use has been made of changing the functions of the relay groups among concentrators. In most cases, this has permitted 20 of the 24 contacts per relay to be used for path selection purposes.

External to the concentrator, additional selection groups steer the pulser to the appropriate concentrator. Another group, the order group, is a six-relay group. As can be seen from Figs. 10 and 11, there are three possible pulsing points into a concentrator: A, B, and C. A bipolar ferreed switch (cutoff) is used for the scanner access contact. In a manner similar to the junctor switching frame, the order relays, by controlling the polarity of the pulsing and the point of application, perform the following types of orders:

(a) close stage 0 and stage 1 crosspoints with the cutoff contact either opened or closed,

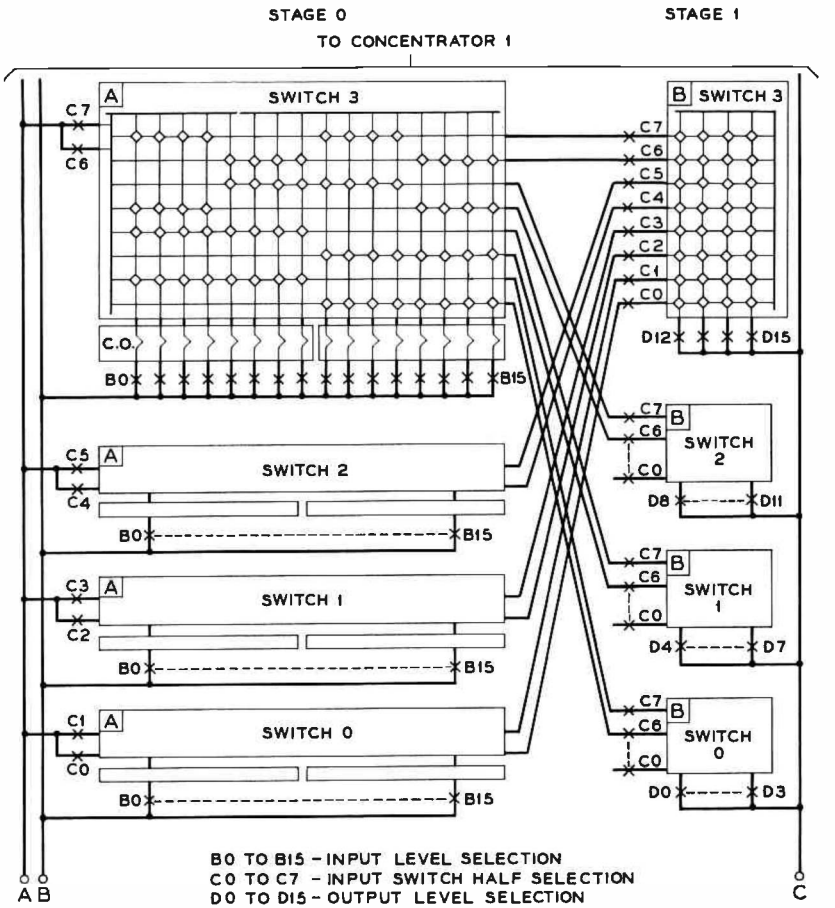


Fig. 10 — Line switching frame: path selection within a 4:1 concentrator.

(b) open stage 0 crosspoint with the cutoff contact either opened or closed, or

(c) open stage 0 and close stage 1 crosspoints with no change in cutoff contacts.

VI. INFORMATION PROCESSING

6.1 *Introduction*

The different types of network frames differ mainly in the internal organization of the network fabric. This results in differences in the

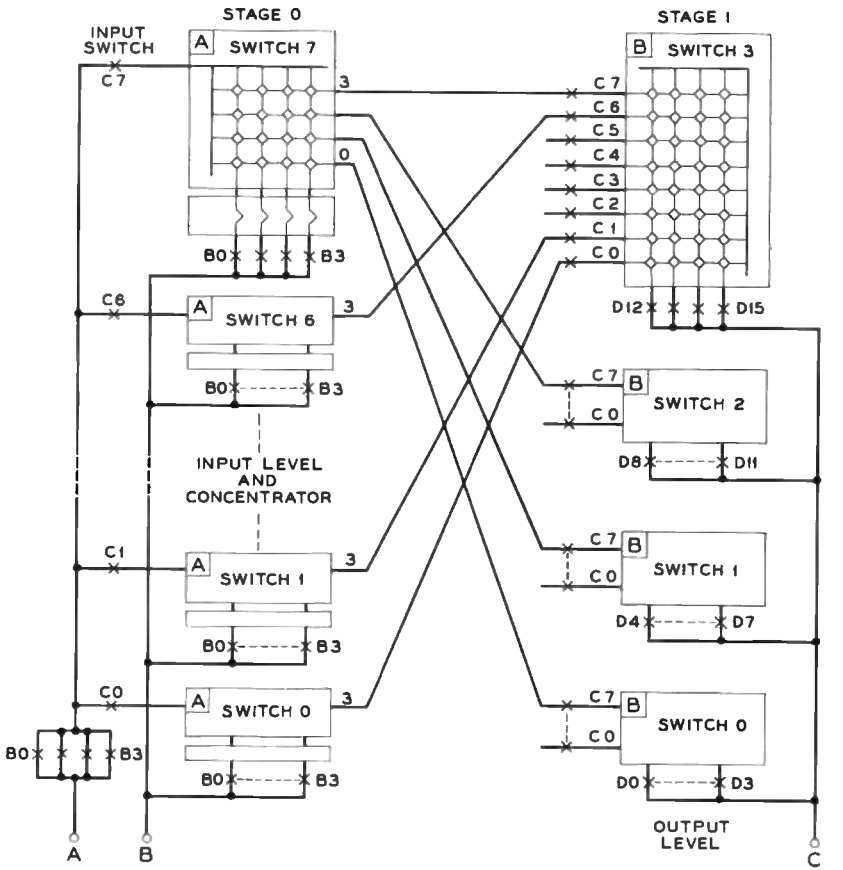


Fig. 11 — Line switching frame: path selection within a 2:1 concentrator.

information processing section of the frame, but these differences are minor. The same type of apparatus and circuit configuration is used in all types of network frames.

Reliability is achieved by means of duplication and redundancy. Duplication is achieved by providing two groups of circuits, each of which is capable of controlling the network fabric. Each of these circuit groups is assigned to control one-half of the frame fabric. Both circuit groups are functionally independent and under normal operating conditions may execute instructions simultaneously. Both circuit groups are normally active.

It is of prime concern that the probability of losing control of any portion of the network fabric be as small as possible. Therefore sufficient

circuit redundancy has been provided in each of the two major circuit groups that either of the two circuits can control all of the frame fabric. This provides standby facilities for controlling any part of the frame fabric.

The functional organization of a network frame is shown in Fig. 12. The cycle time of a network frame is 20 milliseconds. This time is measured from the instant an enable signal and peripheral bus addressing information are received and continues until the time the frame has executed this information and is prepared for a new instruction. With two active circuit groups in each frame, a network frame may establish new connections at a maximum rate of 100 new connections per second.

The high speed of the peripheral bus system in conjunction with the relatively slow-acting network frame requires that the bus information be stored in the network frame for a large portion of the frame operating cycle. The peripheral bus information, which is stored in a buffer register, is translated and checked. The checking circuitry determines if the state of the translator is valid or not. A nonvalid combination of bits on

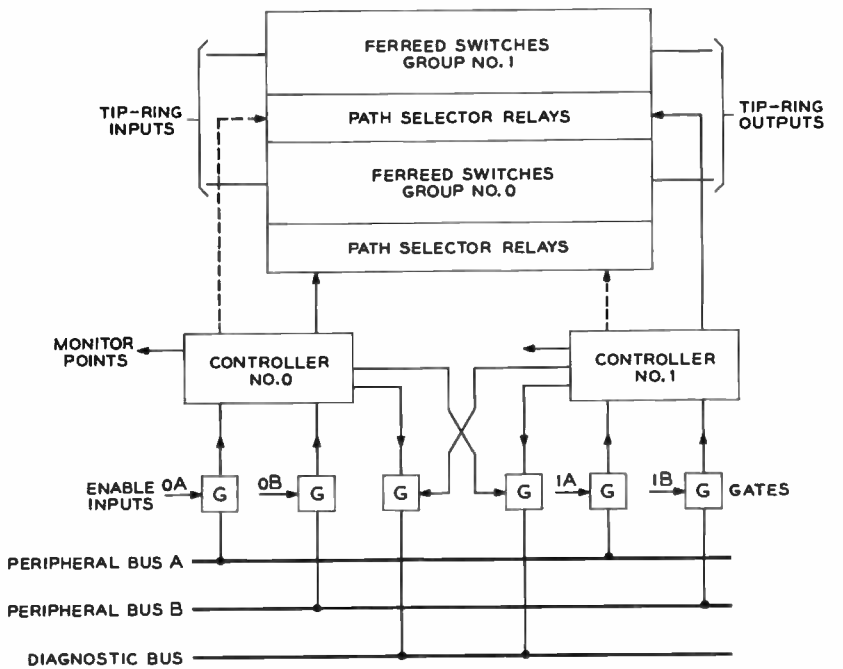


Fig. 12 — Functional organization of a network frame.

the peripheral bus or some malfunctioning of the buffer register or translator will cause the frame circuitry to stop processing the information. These checks will provide some assurance that erroneous action cannot be performed in the frame network fabric.

Each of the two major circuit groups in a frame has three monitor points which connect directly to the central control. These inform the central control about the status of the network control circuit at any time.

6.2 *The Peripheral Bus*

The peripheral bus consists basically of two 38-pair 26-gauge switchboard cables. These two cables are referred to as the 0 and 1 peripheral buses and interconnect the central control and all network frames. Each wire pair of both buses is assigned a bit position number ranging from zero to 37. A bit is said to be present in a position whenever an 80-milliampere, 0.5-microsecond current pulse occurs in that wire pair.

Each network frame expects bits to be present simultaneously in certain combinations on 36 of the bus bit positions. The format of the 36-bit binary word has been selected in such a manner that only a very simple translation has to be performed in the network frames. Bit positions 0 to 29 are used to identify a particular path. In the junctor and trunk switching frames, there are as many as 16,384 possible paths defined by these 30 bits. The grouping of these bits in several one-out-of-four and one-out-of-two subgroups is shown in Table I. Similarly, the line switching frame interprets the peripheral bus bits in several groups of one out of four and one out of two, as shown in Tables II and III. The peripheral bus information does not contain any check bits. The network frame circuitry contains facilities for checking the validity of the bus information. Each type of network frame expects a fixed number of bits to be present and checks that these appear in positions that satisfy the format for that type of frame.

Bits in positions 0 to 29 identify a path in a network frame. Bits in positions 30 to 35 inform the circuitry what kind of action should be performed on the selected path.

This multiple one-out-of-n grouping of the peripheral bus bit positions does require that the central control perform an elaborate translation of its internal binary information. This inconvenience and cost in the central control becomes relatively insignificant when one considers the simplicity of translation and checking that can be employed in each network frame. Bit position 36, referred to as the "reset signal," does

TABLE I—JUNCTOR SWITCHING FRAME: ASSOCIATION BETWEEN PERIPHERAL BUS BIT POSITIONS AND FRAME FUNCTIONS

			Function Selection		Grid Selection		Selection of Path within a Grid											
			1/2	1/4	1/2	1/2	1/2	1/4	1/2	1/4		1/2	1/4		1/2	1/4		
Bit. Pos.	37	36	35 34	33 32 31 30	29 28	27 26	25 24	15 14 13 12	21 20	11 10 9 8		19 18	7 6 5 4		17 16	3 2 1 0		
Grid			Order Group (see below for orders)		Group No. 6	Group No. 5	Relay Group No. 4		Relay Group No. 3			Relay Group No. 2			Relay Group No. 1			
0	FCG	reset			grid 0 & 1 or 2 & 3	grid 0 or 1 grid 2 or 3	output switch	output level	input switch	input switch		output level	output switch		input level	input level		
1							output level	output switch	input switch	input switch		output switch	output level		input level	input level		
2							output switch	output level	input switch	input switch		output level	output switch		input level	input level		
3							output level	output switch	input switch	input switch		output switch	output level		input level	input level		

Order No.	Designation	Function
0	remove NT, FCG, or verify	releases F contact and removes all connections to the test vertical
1	connect	closes 0 and 1 crosspoints and releases associated F contact
2	connect with FCG	closes stage 0 and 1 crosspoints and operates the associated F contact; connects FCG test circuit to the test vertical
3	test order	changes mode of operation of frame; does not change the state of any crosspoints
4	not used	
5	connect verify (LS)	operates an F contact and connects a resistor across the tip and ring of the test vertical
6	operate NT	operates an F contact and makes the test vertical available to external test circuits
7	connect verify (GS)	operates an F contact and connects a resistor between the ring conductor of the test vertical and ground

TABLE II—4:1 LINE SWITCHING FRAME: ASSOCIATION BETWEEN PERIPHERAL BUS BIT POSITIONS AND FRAME FUNCTIONS

	Line Switching Frame Contact Assignments																							
	1/2		1/4				1/2		1/4				1/4			1/4			1/4			1/4			1/4			
Bit Pos.	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Conc.	Order Group Relays								Gp. 4 Relays				Group 3 Relays								Group 2 Relays								Group 1 Relays							
0	0 not used						basic or supplementary		concentrator selector				input switch half								output level								input level							
1	1 connect* (close 0 and 1 open C.O.)												input switch half								output level								input level							
2	2 FCG (open 0, close 1 no change on C.O.)												input switch half								output level								input level							
3	3 test order												input switch half								output level								input level							
4	4 hi and dry (open 0 open C.O.)												input switch half								input level								output level							
5	5 connect test (close 0, close 1 close C.O.)												input switch half								input level								output level							
6	6 not used												input switch half								input level								output level							
7	7 restore C.O. (open 0, close C.O.)												input switch half								input level								output level							

* For "close 0 and 1, open C.O." read "close crosspoints in stage 0 and stage 1, open scanner cutoff ferreed."

TABLE III — 2:1 LINE SWITCHING FRAME: ASSOCIATION BETWEEN PERIPHERAL BUS BIT POSITIONS AND FRAME FUNCTIONS

	Line Switching Frame Contact Assignments (2:1 Ratio)																															
		1/2		1/4						1/4				1/4				1/4				1/4				1/4						
	Bit Pos.	35	34	33	32	31	30	29	28	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Conc.	Order Group Relays							conc. 0–7 or conc. 8–12		Group 3 Relays								Group 2 Relays								Group 1 Relays							
0 or 8	0 not used																															
1 or 9	1 connect* (close 0 and 1, open C.O.)																															
2 or 10	2 FCG (open 0, close 1 no change on C.O.)																															
3 or 11	3 test order									output level								input switch								input level and conc.						
4 or 12	4 hi and dry (open 0, open C.O.)																															
5 or 13	5 connect, test (close 0, close 1, close C.O.)																															
6 or 14	6 not used																															
7 or 15	7 restore C.O. (open 0, close C.O.)																															

* For "close 0 and 1, open C.O." read "close crosspoints in stages 0 and 1, open scanner cutoff ferreed."

not appear on the peripheral bus with any regularity. The use of this signal is mainly for the purpose of resetting network frames that may have been unsuccessful in executing their instructions, although it may be used for the purpose of generating an early timeout in any frame. Bit 37, also referred to as an "FCG" (false cross and ground) signal, is used only by the junctor switching frame. This signal appears at regular intervals on the peripheral bus. The signal is present at a different time than information bits 0 to 35. A certain time relationship does exist between this signal and the information bits. In the junctor switching frame, this signal is used as a time reference signal or clock.

Since the combination of bits present on either of the peripheral buses will appear in all network frames, all associated cable receivers will respond in all frames. This by itself will not cause any action in any of the frames. A further selection must be made as to which frame should respond. This selection of frame is accomplished by special signals called "enable signals." Each circuit has, in addition to its peripheral bus connections, connections to the central control via the central

pulse distributor[6] for this purpose. Since the network frame circuitry and the peripheral bus are both duplicated, four such enable signals per frame are possible, any one of which may be used to select one of the four possible access patterns. The presence of an enable signal indicates directly which of the four possible combinations of bus and circuit shall be used. The network circuitry will return to central control an opposite-polarity enable verify over the same pair of wires which carried the enable signal. The presence of this signal acknowledges the receipt of the enable and indicates that the frame circuitry has responded properly.

6.3 *Buffer Register*

The central control is capable of transmitting information data on the peripheral bus at a rate of one complete instruction every 11 microseconds. Each network frame requires 20 milliseconds to complete an instruction. Therefore it is necessary that the network frames record the bus information for the length of time necessary to execute the instruction. The bus information is stored in buffer register flip-flops. The flip-flop circuit, illustrated in Fig. 13, is set when transistor Q_2 is on. The output of this circuit is capable of sinking 100 ma. There is one set of registers associated with each of the two groups of controlling circuitry. Each of the two sets of registers in a frame has one such bistable circuit per bit position. Each flip-flop may be set from either of the two buses over separate connections. Two of the three primary windings provide for the separate bus connections. A positive register

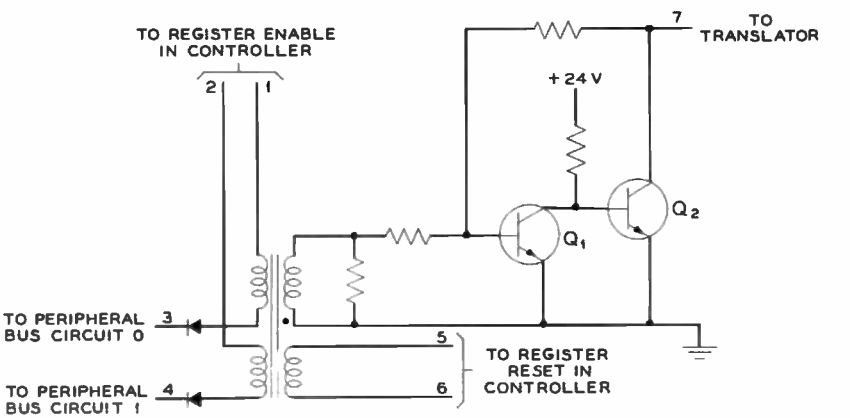


Fig. 13 — Buffer register flip-flop.

enable pulse generated by the network circuitry will appear either on terminal 1 or on terminal 2 and will provide the set current for each register flip-flop. The presence of an address bit, a negative pulse on either terminal 3 or 4, will terminate one of the two primary windings. A third winding provides means for resetting the flip-flops.

6.4 *Translator*

The principle employed in translating the multiple one-out-of-n peripheral bus information is the same in all types of network frames. The line switching frames translate the bus information into multiple one-out-of-sixteen selections, while both the trunk and the junctor switching frames translate the information into several one-out-of-eight selections. In the line switching frames, pairs of one-out-of-four bus information are translated to one-out-of-sixteen selections. The circuits of the one-out-of-sixteen and one-out-of-eight translators are shown in Figs. 14 and 15. They consist of 311-type reed relays which

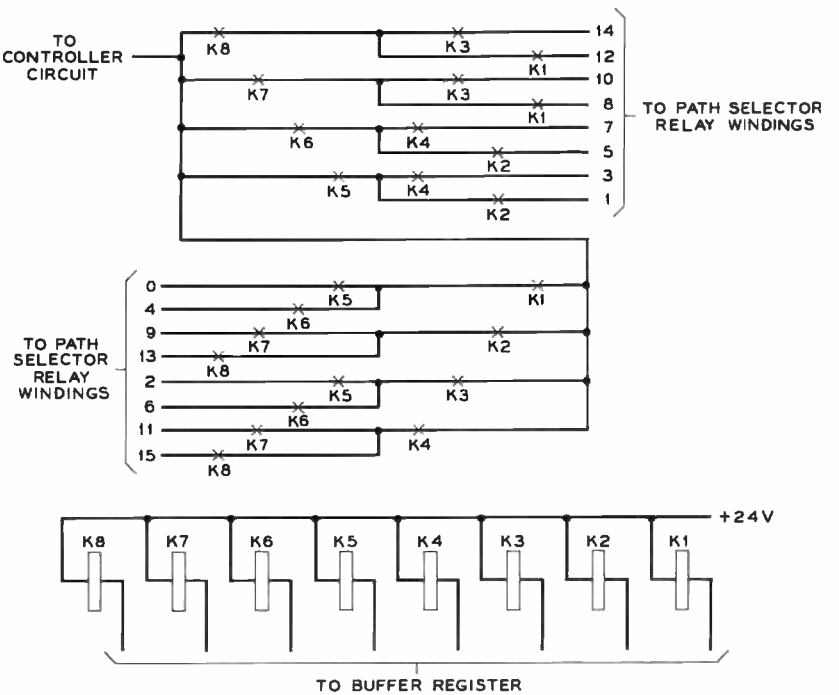


Fig. 14 — 1:16 translator circuit.

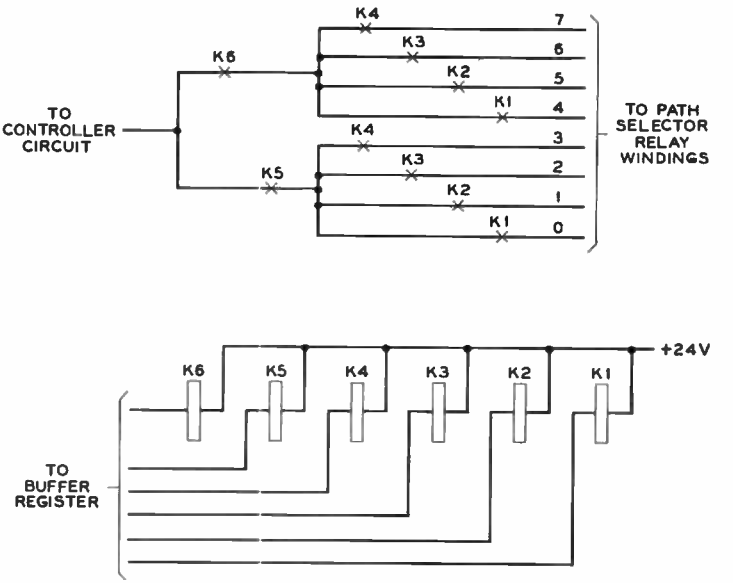


Fig. 15 — 1:18 translator circuit.

are operated directly by the register flip-flops. A fixed wiring pattern of the reed contacts results in the selection of a particular metallic path in each translator group. This metallic path is part of the operate path of a wire spring relay which is used in the ferreed pulse path selection. These relays are grouped and organized to correspond to the several groups of translators. A network frame contains two of each of all functional groups such as registers and translators. Each group of circuits is, under normal operating conditions, strictly associated with the control of one-half of the network fabric in that frame. An alternative access path between the two groups of circuits is provided by redundancy in the translators. Each buffer register circuit operates two reed relays in parallel, and two identical but separate metallic paths are established through each translator group. The selection of either of the two halves of the translator group determines if a circuit is to control its normal half or operate into the alternative path.

The path through a translator establishes continuity between an input terminal and the winding of a wire spring relay. The wire spring relay operate path is completed when cut-through occurs. At the time of cut-through, one path should exist through each translator group.

6.5 *Group Check Circuit*

Each network frame checks the validity of the peripheral bus information. This check is based upon the one-out-of-n grouping of the peripheral bus bit positions and the conversion of these groups into fewer but larger one-out-of-n groups by the translators. Each translator group should, at the time of cut-through, have only one path established between the common cut-through terminal and one of its output terminals. The absence of such a path or the presence of more than one path will cause the group check circuit to respond. Each wire spring relay in a relay group of the pulse path selector has a contact protection network connected across each of its two windings. One side of the output path for one group of relays is common and connects to battery through one winding of a transformer in the group check circuit, as shown in Fig. 16. At the instant of cut-through, a transient current will be present in the group check transformer winding. The waveform of this current is largely controlled by the protection network in shunt with the selected wire spring relay winding. At the same time, a reference current is present in a second winding of the group check transformer. The waveform of this current is controlled by an external *RC* network.

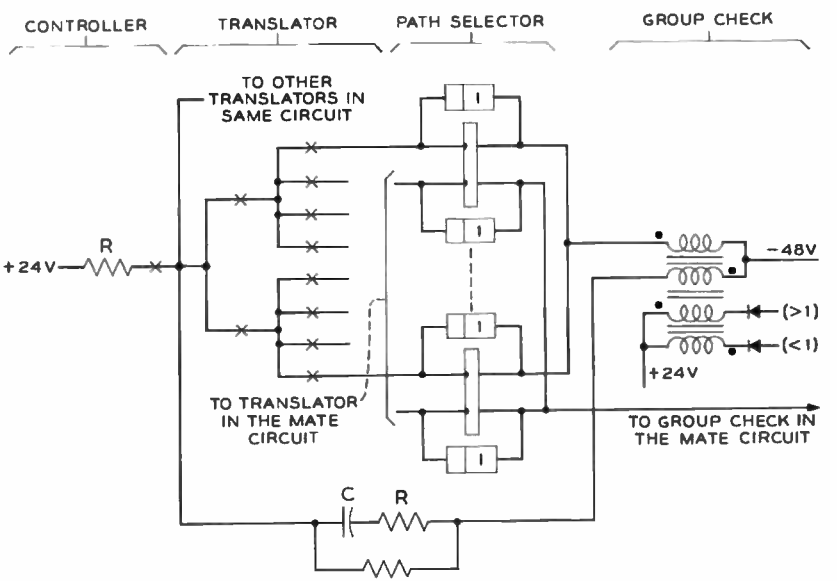


Fig. 16 — Check for single path through a translator by comparing the transient currents at the instant of cut-through.

This external network is designed to give a reference current which closely equals that which is present when one relay operate path exists. The effect of the two nearly equal transient currents is to cancel in the group check transformer and, as a result, no signal appears on either of the two secondary transformer windings. A group check failure signal will be present on both the secondary windings if no, or more than one, relay operate path exists. The polarity of the signals on the secondary windings depends on the type of failure that exists. This is used to detect and report the kind of failure that has occurred. Knowledge of the type of failure is important in diagnosing a frame trouble. One four-winding transformer is associated with each translator group. A group check failure of either kind will cause the wire spring relay voltage cut-through to open. This occurs within a couple of milliseconds and is considerably less than the minimum operate time of any of the wire spring relays.

6.6 *Sequence Control*

The network controller performs all essential timing and sequencing needed for the operation of the switching network circuit. The receipt of an enable signal from the peripheral bus circuit initiates the chain of events leading to the closure of a tip-ring path in the frame. A block diagram of the controller logic is shown in Fig. 17. In the following discussion, all times are measured from the receipt of an enable signal from the central control.

An enable signal may be received from the peripheral bus circuit on either the EN0 or EN1 lead. The corresponding flip-flop will set. The setting of the flip-flop will start the timing circuit, which generates a 2.4-microsecond timing pulse. The timing pulse is gated with the states of the enable flip-flops to enable the address registers to read the information on the peripheral bus (0 or 1) corresponding to the received enable (0 or 1). The trailing edge of the timing pulse is differentiated and gated with the enable flip-flops to return an enable verify signal to the peripheral bus (ENV0 or ENV1) corresponding to the enable (0 or 1) which was received. The differentiated pulse is also used to set the F and VCT flip-flops and reset the S flip-flop. Setting the VCT flip-flop operates the VCT mercury relay, which then cuts through +24 v to the wire spring relay coils through the one-out-of-eight and one-out-of-sixteen translators. The TCV lead activates the group check circuit at this time. The operation of the VCT relay is slowed to insure that the reed relays in the translators have operated before cut-through

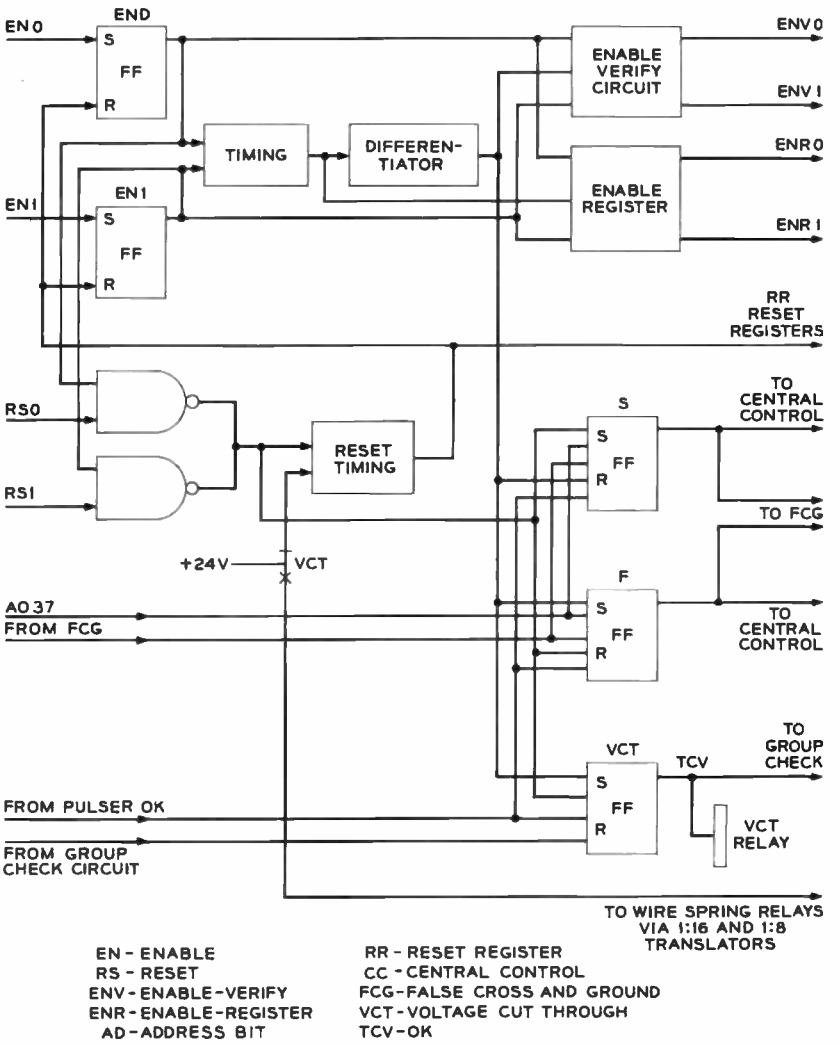


Fig. 17 — Sequence control diagram.

occurs. Under normal conditions, nothing further occurs in the network controller until the ferreed pulser has operated. The signal indicating satisfactory pulser operation starts the resetting of the controller by resetting the S, F and VCT flip-flops. Resetting the VCT flip-flop causes the VCT relay to release, removing power from the wire spring relays.

VCT relay release generates a signal in the reset timing circuit which resets the enable flip-flops and the address registers. The status of the S and F flip-flops is returned to central control. Central control expects that both the S and F flip-flops will be reset in 20 milliseconds. If either or both of these flip-flops are set, it is an indication that the controller has failed to complete a normal cycle.

One such case would be the occurrence of a group check failure. A group check failure causes the VCT flip-flop to be reset. The VCT relay releases, removing power from the wire spring relay. The VCT relay is operated for only 2 milliseconds in the case of a group check failure. This time is insufficient to operate the wire spring relays. The group check failure results in the S and F flip-flops remaining in the reset and set states respectively. When central control detects such a failure, or if it wishes to stop the controller from carrying out a previously transmitted order, it sends a reset signal on the peripheral bus (0 or 1). The reset signal is accepted only from the peripheral bus which corresponds to the previous enable (0 or 1). The external reset signal resets the F and VCT flip-flops and sets the S flip-flop. Resetting the VCT flip-flop initiates the resetting of the enable flip-flop and address register as previously described. The status of the S and F flip-flops indicates to central control that the frame is reset but that the reset was accomplished using the external rather than the internal reset.

The first enable received by a frame operates one of the enable flip-flops and reads information from the appropriate peripheral bus into the address register. Subsequent setting of the other enable flip-flop before the resetting of the first flip-flop will not cause reading of the other peripheral bus, because an EXCLUSIVE-OR circuit, part of the timing circuit, acts to prevent this. Thus the network frame is locked out after receipt of an enable until internally or externally reset.

On junctor switching frames, the S and F flip-flops are also connected to the FCG circuit. This circuit is activated only after a successful network cycle has been completed. The receipt of bit 37 sets both the S and F flip-flops, and interrogates the FCG detector circuit. A successful FCG check will reset both the S and F flip-flops.

6.7 *Ferreed Pulser and Path Check*

The ferreed pulser must be capable of producing a minimum pulse current of 9 amperes in the pulsing path. The 9-ampere minimum must be maintained over all combinations of battery voltage and ferreed load. The pulse width is nominally 300 microseconds at the 2.5-ampere points.

The load impedance varies from 10 ohms pure resistance to 12.5 ohms in series with 1.2 millihenries. Since the available battery voltage varies from 63.7 v to 79.1 v, some step-up in voltage is needed to develop the 9-ampere pulse. The pulser circuit is shown in Fig. 18. A step-up transformer with 4 to 1 secondary-to-primary turns ratio will raise the voltage sufficiently to develop at least 9 amperes into the highest impedance load at the minimum battery voltage. The primary current is obtained by discharging capacitor C_1 through a silicon control rectifier (Q_1) into the primary of the transformer (T_1). The recharging current of the capacitors is limited by a series resistor (R_1) and inductor (L_1).

The wire spring relays used to set up the pulsing path have a variation in their operate time. This makes it desirable to operate the pulser upon the completion of a pulse path rather than on a fixed time basis. To accomplish this, circuitry is included in the pulser which detects the presence of continuity on the pulse path. Two relays (relay 1 and relay 2), whose windings are in series with the battery voltage, transformer secondary and pulsing path, operate when continuity exists through the pulsing path. Relay 1 operates first and removes capacitor C_1 from the battery. The operation of Relay 2, delayed slightly by capacitor C_2, removes a shunt from the gate of the silicon control rectifier and

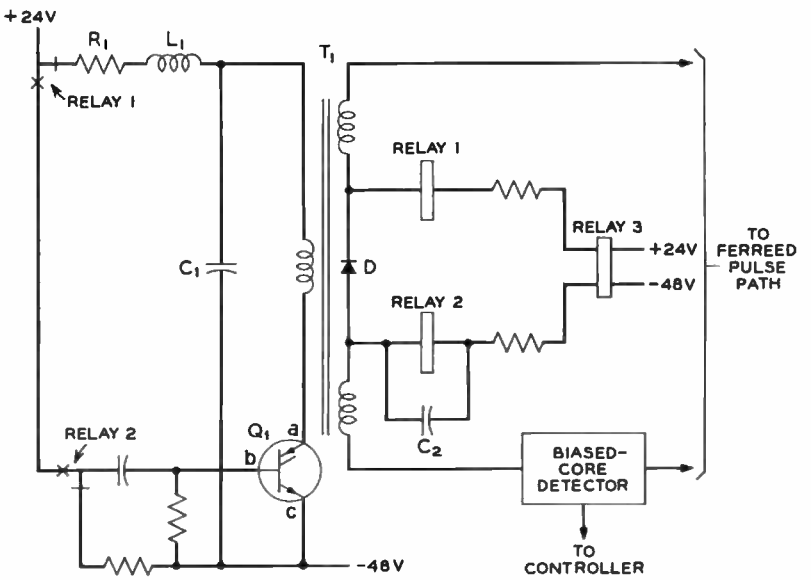


Fig. 18 — Ferreed pulser diagram.

triggers the silicon control rectifier, causing the capacitor to discharge into the transformer primary. The diode (D) in series with the transformer secondary conducts in the forward direction during pulsing and prevents any reversal of current through the load due to overshoot on the trailing edge of the pulse.

The windings of a third relay (relay 3) are also in series with the battery which is connected to the transformer secondary. Under normal conditions, with no short to ground existing in the pulsing path, the current in each winding of the relay will be the same and the relay will not operate. If, however, a short to ground exists somewhere in the pulsing path, the current flow in the winding connected to the −48-v supply will be twice that in the other winding and the relay will operate. The operation of this relay may be detected by the central control.

It is necessary for the pulser circuitry to give an indication to the controlling circuitry that the pulser has fired, so that the wire spring relays and other circuitry can be reset. Also, it is desirable to monitor the amplitude of the pulse to insure that it is of sufficient amplitude to operate the ferreed crosspoints. These two functions are combined in the biased-core detector. This is a square-loop core biased so that the pulser current must exceed 9 amperes to switch the core. A winding on the core goes to the controlling circuitry and signals when the pulser current is greater than 9 amperes.

6.8 *Test Vertical Functions*

Any path through the network fabric involves the use of at least one junctor. The junctor and associated junctor terminals are both physically and functionally the center of the network.

A special connection may be made to any pair of junctor terminals. This is accomplished by means of a test vertical, sometimes called the "ninth vertical." The test vertical and its associated controls are part of the junctor switching circuit. It consists basically of a tip-ring pair which may be connected to any pair of junctor terminals by means of a bipolar switch associated with each junctor terminal. The test vertical and the associated controls are organized in a manner similar to the frame fabric and control circuits. Therefore the junctor switching frame contains two test verticals, one associated with each of the two pairs of grids. One such tip-ring test access serves 128 pairs of junctor terminals. The bipolar test access ferreed switches may be operated or released by separate selected pulse paths, or they may be controlled in some cases by inserting the bipolar switch in the over-all pulsing path

of the junctor switching frame. A test vertical connection may be established or removed independent from or concurrent with the selection of a new network path. The circuitry which controls the state of the test vertical is basically an extension of the frame control circuitry.

The central control uses the test vertical either to perform a test or to establish a test condition on any pair of junctor terminals. The FCG check is a frequent and routine test which is performed on most new network paths. This test checks the newly established path for false crosses or grounds. Access to the path is by means of test verticals. The actual test is conducted by a current-sensing ferrod which is connected to the test vertical as indicated in Fig. 19. An instruction to the junctor switching frame to connect with FCG will establish the desired path through the fabric of the frame and connect the associated pair of junctor terminals to the test vertical. The current-sensing FCG detector circuit is then connected to the vertical. These actions take up most of

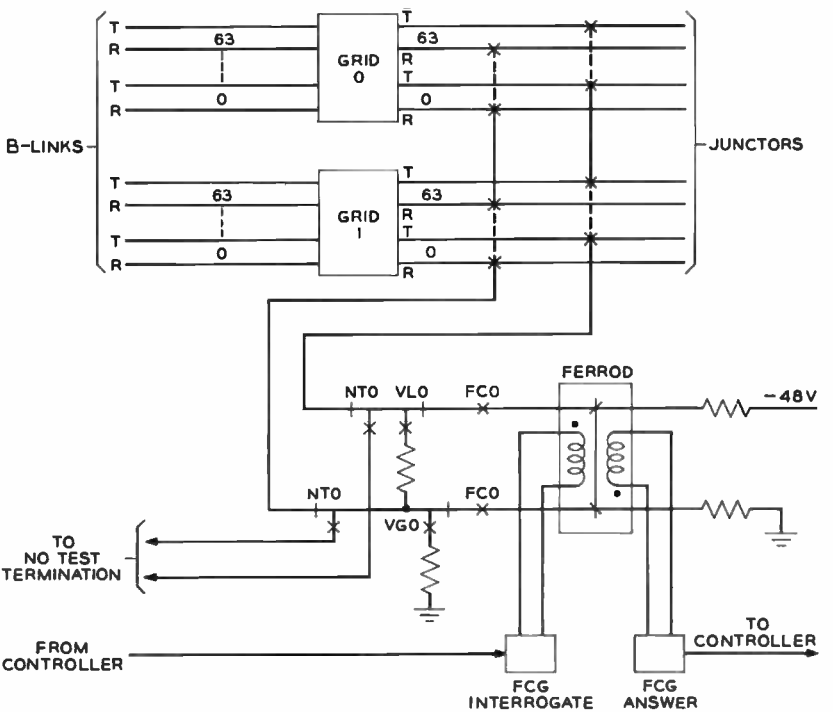


Fig. 19 — Test vertical access for one pair of grids.

the 20-millisecond frame cycle time. Within a millisecond of the end of the cycle, the FCG signal will be present on bus bit position 37. If the indication from the frame control circuitry is that all frame fabric actions have been executed successively, then the presence of this FCG signal will initiate the interrogation of the FCG ferrod. The success or failure of this test will be conveyed to the central control by means of the S and F monitor points of the frame control circuitry.

A "connect verify loop start" or "connect verify ground start" instruction may be performed by the junctor switching frame. These functions connect a resistor across the tip and ring conductors or between only the ring conductor and ground of the test vertical pair. These two test conditions are used by the central control to verify that it has restored the connection between a line terminal and its line scanner ferrod. With a network connection between a junctor terminal and a line terminal, central control can generate a request-for-service condition in any line scanner ferrod. Since the line scanner contains two types of ferrods, two separate test conditions may be established by means of the test vertical.

In addition to the primary functions of the FCG test and both of the restore verify test conditions, these tests also serve as additional aids in the systems network fabric diagnosis.

A fourth function which uses the test vertical is the operate-no-test function. A bridged test connection may be made to any pair of junctor terminals. This type of connection is traditionally referred to as a "no-test connection." This general-purpose type of test connection provides the means for connecting any external transmission testing facilities to any network path.

Each of these four test vertical functions requires a full network cycle. A test vertical may be used for only one of these four test conditions at any time.

A junctor switching frame may execute seven separate instructions; five of these involve the use of the test vertical. Initiation of any of these test vertical functions requires the presence of the peripheral bus bit 37. This FCG signal occurs at a time when all action in the network fabric is completed, and the receipt of bit 37 marks the time in the network cycle when the test vertical becomes connected to either of the two resistive verify terminating resistors or to the external no-test connection. An instruction to the junctor switching circuit to remove NT, FCG, or verify will release the selected bipolar ferreed and disconnect the test vertical from any of its previous test connections.

6.9 *Maintenance Facilities*

A number of test points and special circuits are provided in each network frame to facilitate automatic diagnosis. These test points are connected to strategic points in the circuit. These points may be connected to the central control by means of a diagnostic bus. The central control may request that any network frame be connected to this bus by means of an instruction transmitted to the frame. This diagnostic bus interconnects the central control and all network frames in a manner similar to the peripheral bus. Either of the two circuit groups in any network frame may be connected to the central control in this manner. A special instruction must be transmitted to a network frame requesting that one of its circuit groups be put in a test point access mode. By correlating the states and noting the sequence in the change of states of these test points, it is possible to diagnose malfunctions. Some circuit failures require that the circuit be removed from active control of the frame. This is accomplished by putting that circuit half in a quarantined mode of operation. A circuit half is quarantined by an instruction transmitted by the central control to the remaining active circuit half. Similarly, the circuit half may be restored to its active mode of operation by an instruction to the active circuit group. A quarantined circuit does not have access to any part of the frame network fabric. All control of the network fabric is now performed by the remaining operative circuit half. The quarantined circuit can respond and perform all its timing, sequencing and pulsing functions but will, in this mode of operation, use a dummy pulse path. This allows a circuit to be diagnosed either as an active circuit or while in a quarantined mode of operation. All modes of either circuit half of the network frame are under the control of central control. The monitoring channels that exist between the frames and the central control indicate the states of any circuit half at any time.

Manually operated controls are provided only for the removal of power from either side of the frame. These controls are mechanically interlocked pushbuttons. Manual removal of power will be reported to the central control by means of the identical status-indicating test points which exist in each circuit half.

VII. SUMMARY

Laboratory testing and preliminary system evaluation programs have demonstrated that all initial operating objectives of the switching network have been met very successfully. The result of system testing

programs has also confirmed the accuracy and usefulness of the automatic diagnosis facilities incorporated in the network control circuits.

The several codes of ferreed switches which utilize differentially wound ferreed crosspoints are probably the devices of primary interest in the switching network area. The switches are compatible with both the outside plant and the electronic central processors. The direct control of the ferreed switches by a combination of electronic and electromagnetic devices in duplicated circuitry represents at the present time a reasonable balance between cost and reliability.

The packaging of the network fabric with its associated controls in the same frame and the independent operation of the frames from a common peripheral communication bus present to the traffic engineer an extremely versatile system for building office switching networks. Future growth and changes in traffic patterns are accommodated with a minimum of effort and cost.

VIII. ACKNOWLEDGMENTS

Acknowledgment is made of the many contributions of T. N. Lowry, F. H. Myers, and R. L. Simms, Jr., in the circuit planning phases; and to R. A. Shaffer for his contributions in developing the ferreed pulser. The able assistance of R. W. Engroff, W. A. Lawrence, J. A. Romano, and F. J. Wilkinson in testing and maintaining the network is hereby acknowledged.

The authors are indebted to R. W. Ketchledge and A. Feiner for their many valuable suggestions and discussions.

REFERENCES

1. Ferguson, J. G., Grutzner, W. E., Koehler, D. C., Skinner, R. S., Skubiak, M. T., and Wetherell, D. H., No. 1 ESS Apparatus and Equipment, B.S.T.J., this issue, p. 2355.
2. Connell, J. B., Hussey, L. W., and Ketchledge, R. W., No. 1 ESS Bus System, B.S.T.J., this issue, p. 2021.
3. Feiner, A., and Hayward, W. S., No. 1 ESS Switching Network Plan, B.S.T.J., this issue, p. 2193.
4. Feiner, A., The Ferreed, B.S.T.J., **43**, Jan., 1964, p. 1.
5. Feiner, A., Lovell, C. A., Lowry, T. N., and Ridinger, P. G., The Ferreed — A New Switching Device, B.S.T.J., **39**, Jan., 1960, p. 1.
6. Freimanis, L., Guercio, A. M., and May, H. F., No. 1 ESS Scanner, Signal Distributor, and Central Pulse Distributor, B.S.T.J., this issue, p. 2255.

No. 1 ESS Scanner, Signal Distributor, and Central Pulse Distributor

By L. FREIMANIS, A. M. GUERCIO and H. F. MAY

(Manuscript received January 9, 1964)

The peripheral units of No. 1 ESS, described in this article, include the scanners and signal distributors. The scanners serve to collect information for the central control, and the signal distributors execute central control orders in the system. The peripheral units receive orders through the communication buses. The central pulse distributor serves to connect peripheral units to the communication buses so that a central control order, which is broadcast over the bus system, enters one particular peripheral unit.

I. INTRODUCTION

The No. 1 electronic switching system (ESS) includes peripheral units which serve as buffers between the central control with its associated stores and the outside world of lines, trunks, automatic message accounting (AMA) centers, and maintenance personnel. The peripheral units include the switching network, scanners, signal distributors, central pulse distributors, AMA recording equipment, teletypewriters and miscellaneous common systems circuits. This article describes the scanners, signal distributors and central pulse distributors. Associated articles describe the network[1] and other peripheral units[2] of No. 1 ESS.

II. FUNCTIONS OF SCANNERS, SIGNAL DISTRIBUTORS, AND CENTRAL PULSE DISTRIBUTORS

2.1 *Scanners*

Every telephone switching system embodies some mechanism for detecting service requests and supervising calls in progress. Input information of this nature is furnished to No. 1 ESS by the operation of scanners which sample or scan lines, trunks and various diagnostic points at discrete intervals of time as directed by the system. It might be said

that the scanners are the sensory organs of the system, since all external stimuli are received through them.

2.2 *Signal Distributors*

In addition to detecting and monitoring changes by means of the scanners, the system must be provided with means to operate and release relays in trunk, service and power control circuits. The signal distributors translate orders received from central control and distribute high-power, long-duration pulses to the various relays in No. 1 ESS. These relays[3] are controlled by polarized signals and are magnetically latched (held operated), thus providing a memory function in the end device.

2.3 *Central Pulse Distributors*

Many control functions in No. 1 ESS must be carried out at electronic speeds or at speeds exceeding the capability of the magnetic latching relays controlled by the signal distributors. Typical functions are outpulsing on trunks, operating miscellaneous flip-flops and the very important function of "enabling" various peripheral units (network controllers, scanners, etc.). Since the many peripheral units receive their orders from central control over a common bus system,[4] a particular peripheral unit must be connected to the bus when it is to receive an order. This is accomplished by sending a preliminary pulse to a particular peripheral unit before the order is transmitted on the bus system. This preliminary pulse, known as the "enable," is provided by the central pulse distributor, which translates a coded address received from the central control and transmits a pulse to the peripheral unit identified by this address.

III. SCANNER DESCRIPTION

3.1 *General*

Several categories of scanners are provided in No. 1 ESS. Primarily, they differ in purpose, location and the type of sensing element associated with the point to be scanned. For convenience and reliability, the scanners are provided in 1024-point modules which are physically located on network, trunk or junctor frames. In addition, one or more master scanners are provided for diagnostic and miscellaneous scanning functions.

3.2 *Ferrods*

The sensing element used in all scanners is the ferrod sensor (hereinafter called a "ferrod"), a current-sensing device operating on electromagnetic principles. It consists of a ferrite rod around which is wound a pair of solenoidal control windings. In addition, a single-turn interrogate winding and a single-turn readout winding are threaded through two holes in the center of the ferrite rod. The control windings are connected in series with the circuit to be sensed or supervised — for instance, a customer line. The ferrod[5] may be visualized as a transformer in which the magnetic coupling between the interrogate and readout windings is determined by the current in the control windings. This current in turn reflects the state of the circuit to be sensed, such as the on-hook or off-hook condition of a line. A diagram of the ferrod is shown in Fig. 1.

The magnetic state of the ferrite rod is sensed by pulsing the interrogate winding with a 0.5-ampere bipolar pulse. The positive half-cycle, of 3 μsec duration, switches the ferrite immediately surrounding the interrogate winding (single turn), provided the control windings (solenoid) are not energized. Ferrite switching induces a voltage in the readout winding. The induced voltage is approximately 200 mv when the control windings are not energized. On the other hand, when the control windings are energized, the ferrite rod is saturated and the coupling between interrogate and readout windings is greatly reduced, with the result that the readout signal is generally less than 20 mv. The negative half-cycle of the interrogate bipolar pulse resets the ferrite in the vicinity of the holes.

3.3 *Ferrod Types*

Three types of ferrods are used in No. 1 ESS. These differ in sensitivity and winding resistance. Their characteristics are shown diagrammatically in Fig. 1 and tabulated in Table I.

3.3.1 *Type 1B Ferrod*

The medium-sensitivity ferrod, type 1B, is used to detect call originations. It supplies ground and battery to the tip and ring of all loop-start lines through the contacts of a normally closed cutoff ferreed in the line switching frame. In the case of coin or PBX lines, the two windings of the ferrod are connected in series and supply battery to the ring conductor through the cutoff ferreed. The 1B ferrod is wound with resistance wire to limit the current in the presence of an accidentally grounded line.

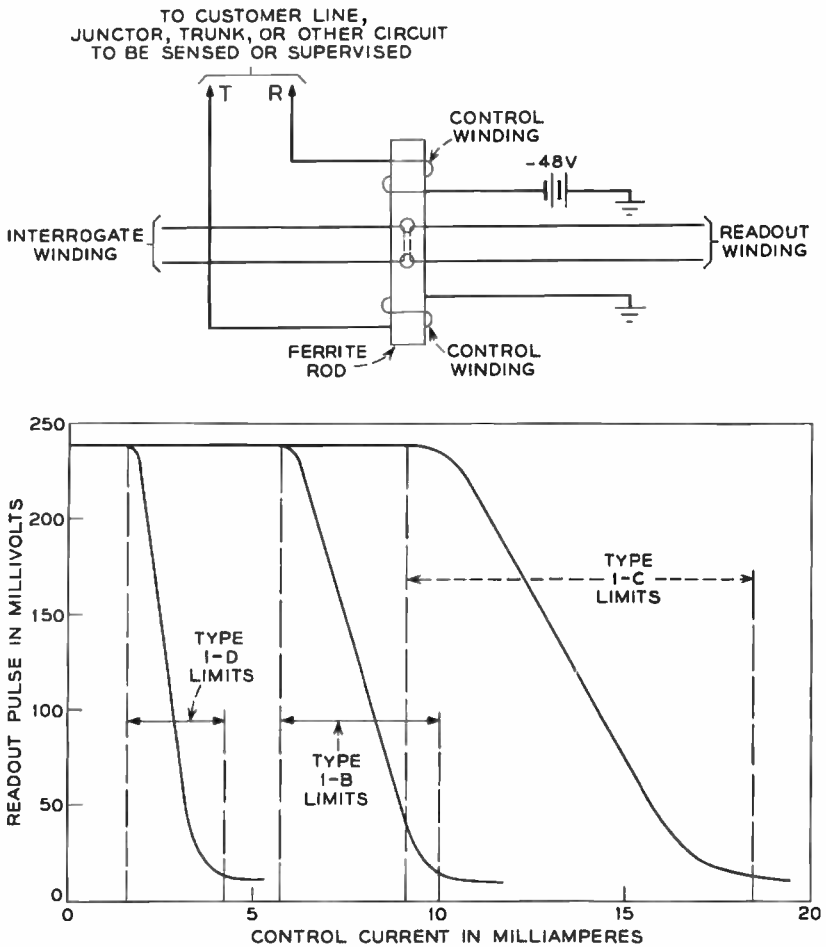


Fig. 1 — Ferrod schematic and typical characteristics.

Incorporated in the ferrod package is a conventional resistor-capacitor contact protection network to protect the cutoff contacts and to suppress electrical interference.

3.3.2 *Types 1C and 1D Ferrod*

The type 1C ferrod is the least sensitive ferrod. It is used in series with the talking battery feed inductors in junctor and trunk circuits to supervise the customer connection. A typical circuit using the 1C ferrod on the

TABLE I — ELECTRICAL SPECIFICATIONS OF THREE FERROD SENSOR CODES

Ferrod Sensor	1B	1C	1D 1E
Number of windings per ferrod	2	2	2
Resistance per winding	660 ± 10%	19 ± 10%	35 ± 10%
Turns per winding	1600	930	1300
Approximate inductance (both windings)	220 mh	70 mh	500 mh
Maximum current	100 ma	100 ma	100 ma
Maximum unbalance between windings	—	1.0 ohms	1.0 ohms
Nonoperate current	5.5 ma	9 ma	1.8 ma
Operate current	10 ma	18 ma	3.9 ma

* The 1E ferrod is electrically identical to the 1D.

local customer side of a trunk circuit is shown in Fig. 2. The type 1D ferrod is the most sensitive ferrod; it is used on the distant office side of trunk circuits.

3.3.3 *Ferrod Sensitivity*

The sensitivity of the various ferrods is controlled by varying the number of turns in the control windings and by adding a magnetic return path for the ferrite rod. The ferrod hole geometry determines the operate threshold and the slope of the output voltage characteristic. The several ferrod types are specified as having a 2:1 ratio of "no output" to "full output" control current, but individual ferrods exhibit a much lower ratio.

3.3.4 *Ferrod Package*

The ferrods are packaged in pairs in a molded ladder similar to that used for wire spring relays. Several ferrod assemblies are shown in Fig. 3.

3.4 *Scanner Operation*

The functional arrangement of the scanner is shown in Fig. 4. The ferrod sensor matrix consists of 64 rows, each containing 16 ferrods. To select a row for interrogation, the addressing equipment of the central control sends an enable pulse via the central pulse distributor. This signal opens the address register, permitting the X and Y signals to be received. One X input and one Y input are then selected and pulsed by the addressing equipment, which will hereafter be referred to only as a "central

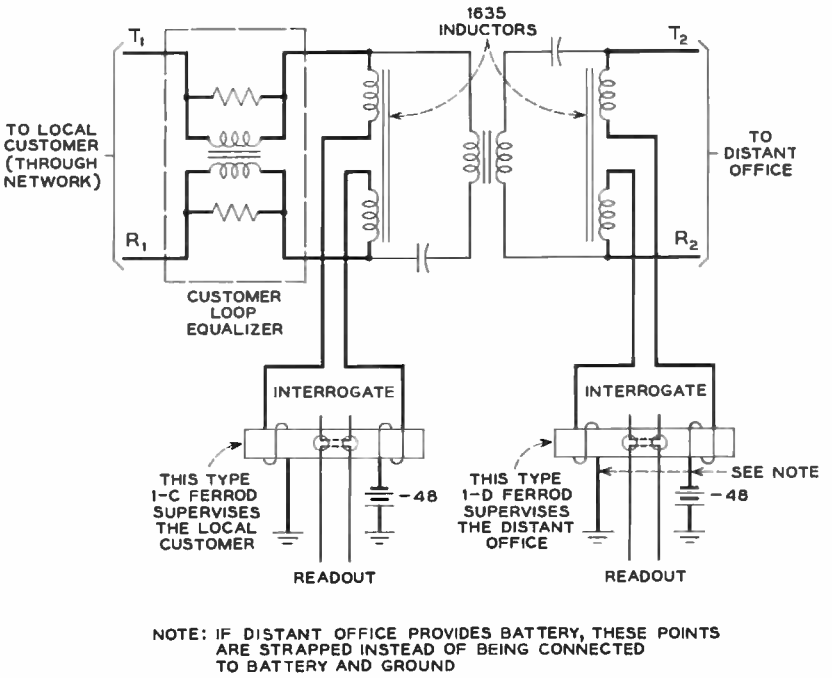


Fig. 2 — Trunk supervision using 1C and 1D ferrods.

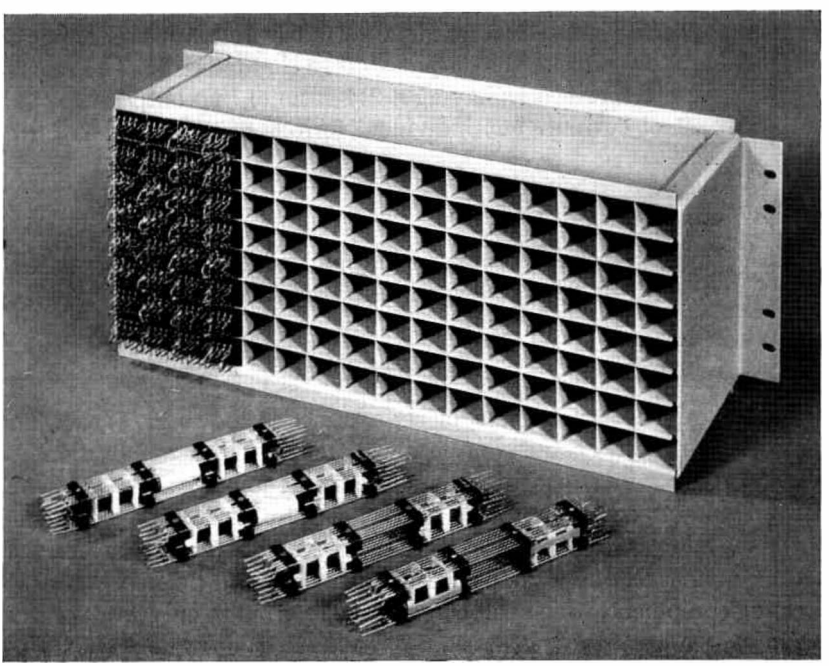


Fig. 3 — Typical ferrods and combined mounting shield.

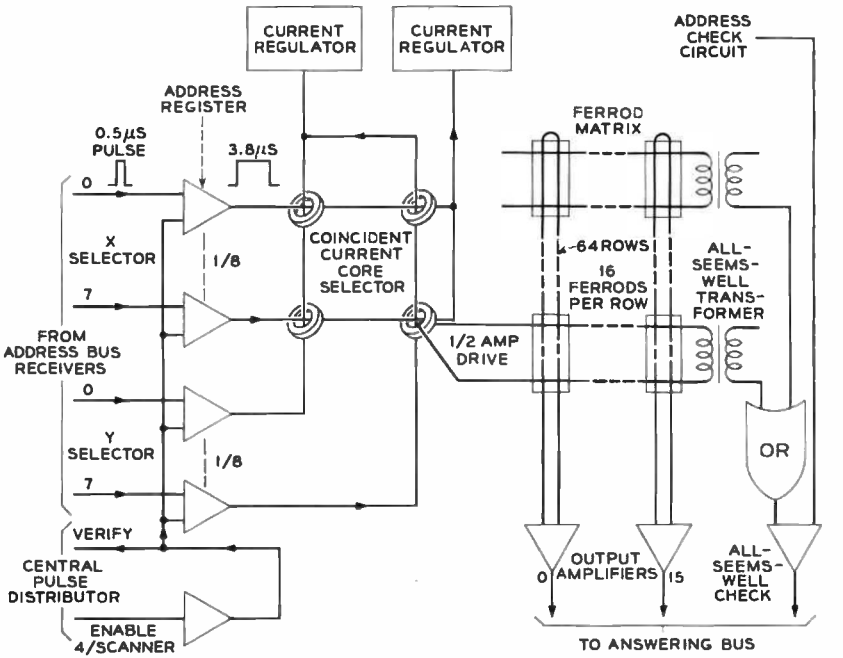


Fig. 4 — Functional diagram of a 1024-point scanner.

control." This in effect selects one of sixty-four cores, the output of which interrogates a row of 16 ferrod sensors. The outputs of all 16 ferrods are detected and transmitted simultaneously back to central control.

3.5 *Scanning Modes and Rates*

The kind of scanner used and the scanning mode, directed or cyclic, depend on the type of information the system desires at any particular time. Each subscriber line is cyclically scanned via a line scanner every 100 msec for originations. When an origination is detected, the subscriber line is connected to a dial pulse receiver. Scanning then occurs, using a master scanner on a directed basis every 10 msec to insure that all of the dial pulses are counted. After all the digits have been received and a connection has been established, the system supervises the call, using either a junctor or a trunk scanner, depending on whether the call is intraoffice or interoffice. Scanning now occurs on a cyclic basis every 100 msec until hangup is detected and the connection is removed. The sequence of events for an intraoffice call is shown in Fig. 5.

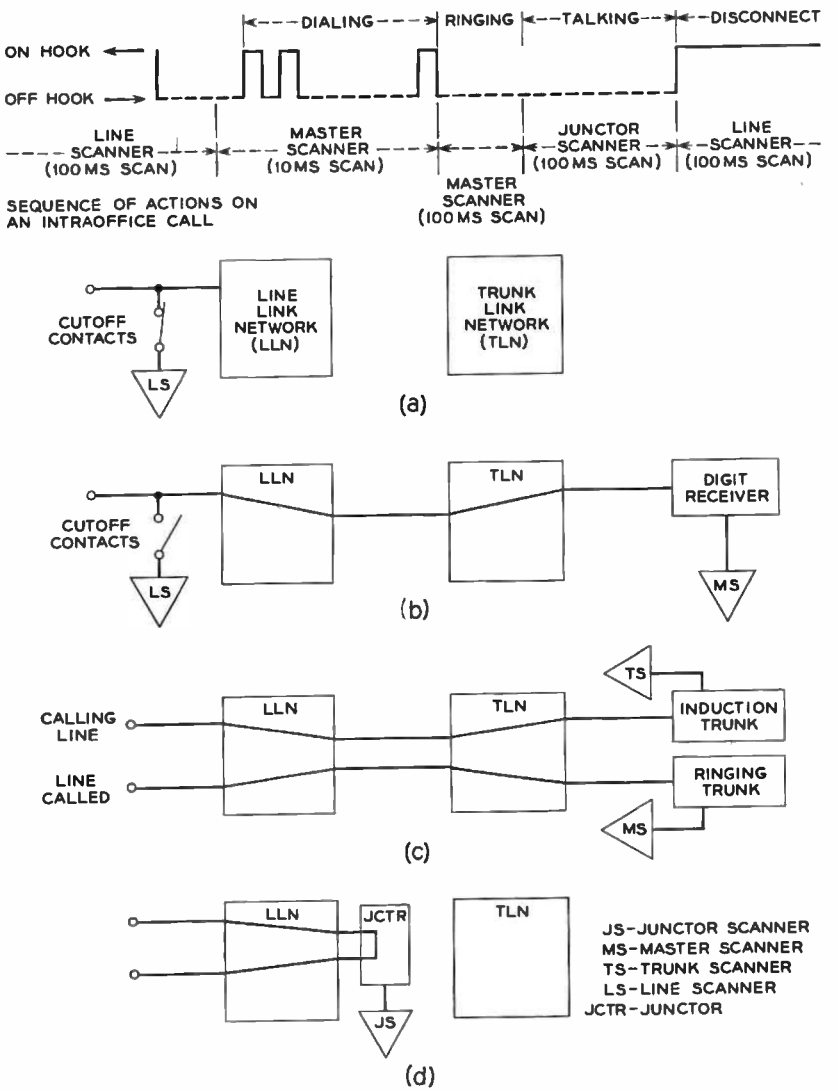


Fig. 5 — Connections for an intraoffice call: (a) line supervision, (b) digit reception, (c) ringing, and (d) talking.

The system uses a basic cycle which is repeated every 100 msec. This cycle is further broken into 5-msec intervals. At the beginning of each interval, the system accomplishes high-priority tasks, such as dial pulse reception and supervisory scanning of junctor and trunk circuits. During each interval, one-half of the digit receivers and one-twentieth of the

junctor and trunk circuits are scanned. Remaining time is used for low-priority work such as line scanning, setting up connections through the network, and maintenance routines.

3.6 *Use of Scanner Information*

There is no intelligence built into a scanner, so all decision making is done by central control. The results of a scan, the states of 16 ferrod sensors, are transmitted simultaneously to central control, where they are stored in flip-flops. These outputs are then compared with the last look information stored in the temporary memory (call store). If there is a change of state, the system, knowing which line (or test point) is involved, concludes that an order was successfully executed or that the next instruction of the stored program must be performed. In either case, the temporary memory is updated.

3.7 *Scanner Organization*

Each scanner consists of two major parts, the ferrod sensor matrix and the access and readout equipment for it. For maximum reliability the access and readout equipment is completely duplicated. Each set of equipment is referred to as a "controller." Ferrods which are assigned on a per-circuit basis are not duplicated.

Being an ESS peripheral unit, each scanner receives and transmits information on the peripheral bus system. This bus system is completely duplicated. Both scanner controllers have the ability to work with either bus. Associated with each scanner are four enable inputs. When the central control selects one of the four enables via the central pulse distributor circuit, it picks a particular controller-bus combination. Each controller-bus combination provides complete access to the unduplicated matrix.

3.8 *Address Register Considerations*

When the central control wishes to scan a particular point, it obtains from the call store the enable information for the particular scanner involved and the X and Y address of the point. This information is in binary form. Since there are many peripheral units (a large central office may contain 100 or more scanners), economic considerations suggest that the central control do as much translation of the binary address as is feasible. The advantage of doing this is obvious. After central control translates the binary word, it sends out addresses in the form of one-out-of-n codes in various numbers of groups, the number of groups and the

value of n being dictated by the size and type of peripheral unit. For a 1024-point scanner there are two address groups each containing eight inputs ($n = 8$).

Addresses (100-ma, 0.5-μsec pulses) are sent to peripheral units as single-rail signals. In practically all cases, these narrow pulses must be stored in order to be useful. Since the input information is sent single-rail, the use of a flip-flop as the storage element would require a reset signal immediately prior to the start of a new cycle or immediately after the execution of an order. The reset signal could, of course, be generated locally or externally. Since the cycle time of a scanner is relatively short (11 μsec), storage elements other than flip-flops become practical. In particular, those types which do not require resetting become extremely attractive. An LC type of pulse stretcher which inherently "resets" itself is used by the scanner in its address registers. This type of circuit is also used in numerous other applications in a scanner.

3.9 *Address Register Circuit*

The address register circuit is shown in Fig. 6. It consists of two 2-input AND gates followed by a single pulse stretcher. Each AND gate, one of the primary windings of the transformer, is associated with one of the input buses. The two AND gate inputs consist of an address input and an enable signal. Normally, the enable signal precedes the address to

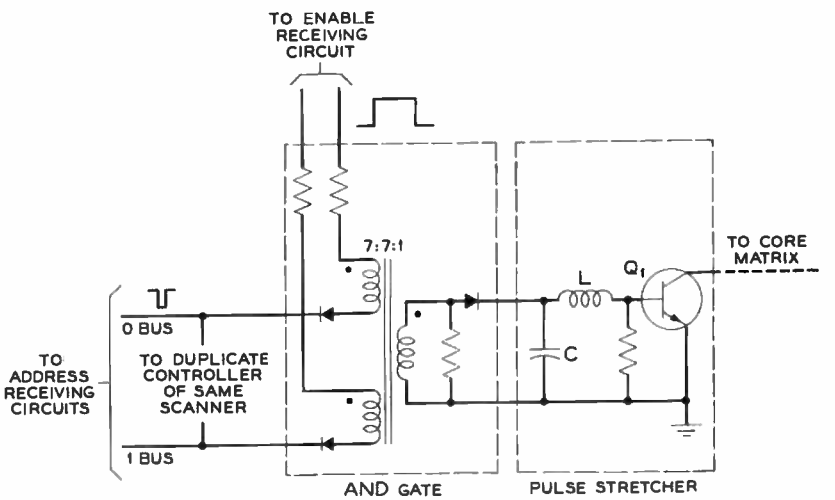


Fig. 6 — Address register circuit.

a controller. To insure sufficient time coincidence of the enable and the address pulses (0.5-μsec pulses), the enable signal is stretched to 2 μsec. Differences in arrival times between the enable and addresses are caused by variations in cable lengths and variations in operate times of transistor circuits.

Battery is connected to one side of the transformer primary by the stretched enable signal. Ground is then applied to the other side of the winding by the address receiving circuit. The resulting current is then transformed to the secondary winding, where it charges capacitor C. The discharge path for the capacitor is through the inductor L and the base-emitter junction of transistor Q_1 to ground. This turns on the transistor for the half-cycle during which the capacitor rings out. A nominal on time of 3.8 μsec is used.

3.10 *Method of Selecting One Row of Ferrod Sensors for Interrogation*

Selection of a particular row out of 64 rows of ferrods in a 1024-point scanner is accomplished using address register circuits previously described and a coincident-current core matrix as shown in Fig. 4. The cores are arranged in an 8-by-8 array. Each core is made of square-loop ferrite and is provided with four windings. Two of these windings are drive inputs and are associated with the X- and Y-address inputs. The third winding is the output, which drives a row of 16 ferrod sensors, while the fourth is the bias winding. When the central control selects one lead in the X group and one in the Y group, it is in effect selecting one row and one column of cores to be driven by 700-ma, 3-μsec pulses. The current pulse to each row and column is a half drive. The core at the intersection of a driven row and column receives full drive, switches, and produces a 0.5-ampere pulse which drives a row of ferrods. Each of the remaining cores in the driven row and column receive only a half drive. These cores do not switch, since half drive is not sufficient to overcome the bias.

The core matrix is duplicated but the ferrod sensor matrix is not. To provide access to the ferrod matrix from either core matrix, the output windings of respective cores in each matrix are connected in series and then to the ferrods. Such an arrangement requires that the standby core matrix present a low impedance to the driving core. This is normally accomplished by maintaining bias current in the standby matrix as well as in the active or driving matrix. The bias current is monitored by a series relay. When current ceases to flow in the bias loop of a matrix, one drive winding of each core in that matrix is switched auto-

matically to the role of a bias winding, using contacts on the released relay. Power is provided to this emergency bias loop from the battery associated with the working controller. The series relay also serves another function. It provides a high impedance in the bias loop, thus preventing unnecessary loading of the drive inputs.

3.11 *Output Circuitry*

It is the function of the output circuitry to convert ferrod outputs (typical amplitudes were given earlier) to signals which are usable by the central control. These signals are then transmitted on both duplicated peripheral reply buses simultaneously. By doing this, synchronous operation of both central controls is possible, since each central control normally receives scanner answers from a different bus.

Scanner output signals consist of either pulses (1's) or no pulses (0's). If the control windings of an interrogated ferrod have less than the nonoperate value of current, the ferrod output is detected and transmitted as a pulse (1). The output of an interrogated ferrod whose control windings have more than the operate value of current is detected and transmitted as a 0.

The output circuitry (see Fig. 7) consists of a 2-input AND-gate amplifier followed by a cable driver. Ferrod outputs provide one input to the AND gate, while a strobe circuit provides the second. Normally, neither of the two transistors in the amplifier is on. If the ferrod output signal has sufficient amplitude to turn on transistor Q_1, its collector current provides base current for transistor Q_2, turning it on. The strobe potential is applied approximately 0.75 μsec after the start of the ferrod output. The resulting current flow from the strobe circuit through the primary winding of transformer T_2 produces an output signal. If the ferrod output signal is insufficient to turn on transistor Q_1, no output signal will be generated when the strobe pulse is applied. However, the strobe input has a fast rise time, so there is an undesirable current path to ground provided by the junction capacitance of the transistor. To prevent this capacitive current from producing a false output, capacitor C is provided as a shunt path around the transformer primary winding.

The readout equipment is duplicated and driven by an unduplicated ferrod matrix. When a controller is ordered to interrogate a row of ferrods, the output amplifiers associated with that controller receive a strobe pulse permitting them to drive both sets of cable drivers.

Signals on the peripheral bus are nominally 0.5 μsec wide. The positive portion of the bipolar ferrod output is approximately 1.5 μsec in duration.

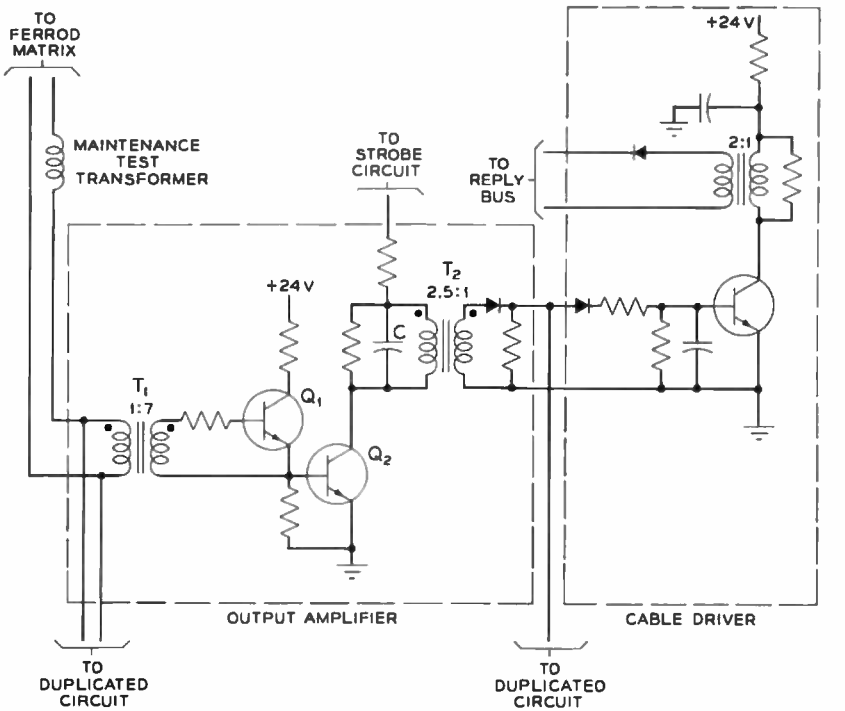


Fig. 7 — Output circuitry.

By sampling this relatively wide pulse with a 0.5-μsec strobe signal, a pulse suitable for bus system use is obtained. There is also a signal-to-noise advantage obtained by strobing. However, the output margins from a ferrod are such that the latter advantage is not an important one in the scanner design.

3.12 *Maintenance Features*

When a controller receives an enable signal over a pair of wires from the central pulse distributor, the signal is stretched to a width of 2 μsec as previously described. At the termination of this stretched enable signal, a new signal is generated by the controller and sent back over the previously enabled pair in such a manner that the enable receiving circuit is not reactivated. Doing this permits central control to verify that the enable signal reached the correct equipment.

During each normal cycle, an all-seems-well scanner (ASW-S) output

is produced and sent with the regular 16 answers to central control. If this pulse does not appear with the scanner replies, the replies are ignored. Central control then repeats the same instruction. If an ASW-S pulse is not produced after several retrials, the particular controller-bus combination used is marked "in trouble" in the call store and another combination is selected merely by using a different enable input to that scanner. When system time permits, the faulty equipment is located, using a stored diagnostic program.

The ASW-S circuitry provides a check on the access part of a controller. In essence, it is a 2-input AND gate. One input indicates that not more than one lead in each address group was selected. The second input indicates that a row of ferrods was interrogated. The absence of either input will manifest itself as an ASW-S output failure.

Normally, scanner outputs are not predictable since they depend upon the state of the ferrods. Therefore, the readout circuitry is routinely checked to determine if it is working properly. This is done using the maintenance test (MT) input which is part of the address bus and is handled like an ordinary address; that is, the enable pulse is first sent, followed by the MT order. In series with each of the 16 readout columns of a ferrod matrix is a secondary of an MT transformer (see Fig. 7), the primaries being in series. By means of these transformers, the MT order generates simulated nonsaturated ferrod outputs. Since the 16 columns receive the test pulse simultaneously, each of the 16 output amplifiers should produce 1's. The ASW-S signal is not produced during this cycle. The absence of an output indicates an open-circuit trouble. Short-circuit troubles in the output amplifiers can be detected by sending an enable and then pulsing only one pair in only one address group. Normally, neither the ASW-S circuit nor any of the 16 output circuits will reply.

A controller can be in any one of several modes, such as normal or power-off. Assigned to each controller are three scan points. When a controller is in a particular mode, it saturates or unsaturates these ferrods so that the unique code assigned to that mode is obtained. By scanning these points, central control can determine the present mode of the controller. These status ferrods are part of the master scanner.

3.13 *Power*

Power for the two controllers of a scanner is provided from an unregulated +24-volt central office battery over separate feeders. The average current requirement for both controllers is approximately 0.5 ampere. In general, ferrod control winding current is provided from the

−48-volt battery. However, there are numerous applications where the +24-volt battery is used instead.

As a maintenance feature, provision has been made to control power remotely by the central control. Out-of-service keys are also mounted on the frame housing the scanner so that power can be removed manually. To insure that power is not removed from both controllers simultaneously, mechanical and electrical safeguards are provided.

3.14 *Equipment Details*[3]

Scanner circuits are built on printed wire boards.[6] This is also true of the coincident-current core matrix. Several circuit packages are shown in Fig. 8. These printed wire boards are plugged into connectors arranged on mounting plates. All connections between circuits are made by wiring from one connector to another.

A master scanner frame is shown in Fig. 9. At the top of this bay are connecting terminal strips and card housings which contain bus and scanner control circuits. The next two mounting plates contain biasing relays, resistors and the control panel. Immediately below these are four ferrod mounting racks. Each rack contains 128 ferrod packages, each package consisting of two ferrod sensors, making a total of 1024 ferrod sensors.

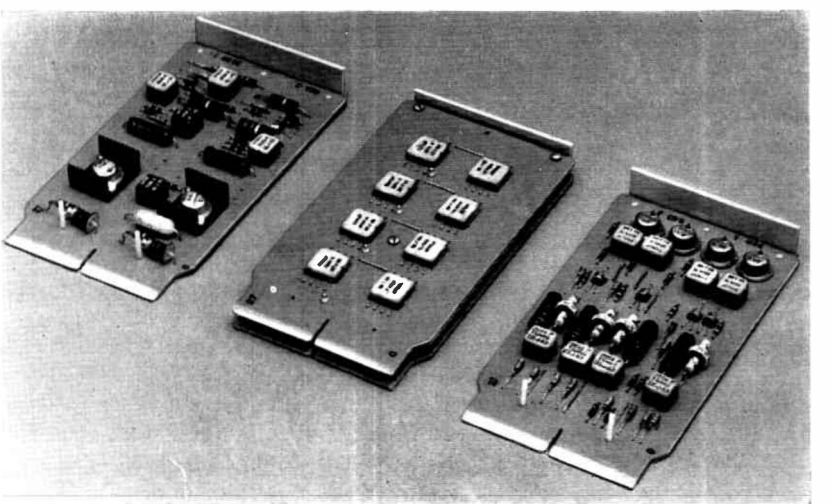


Fig. 8 — Current regulator, 4-by-4 core matrix and address register packages (left to right).

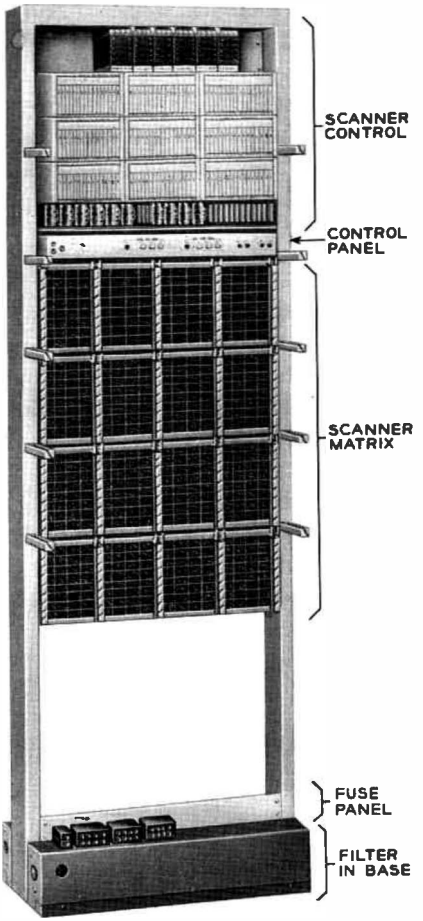


Fig. 9 — Master scanner.

IV. CENTRAL PULSE DISTRIBUTOR AND SIGNAL DISTRIBUTOR DESCRIPTION

4.1 *General*

The central pulse distributors and the signal distributors provide central control with access to many points within the system requiring action signals. Wherever fast access at electronic speeds is required, the central pulse distributor is employed. The signal distributor is used to operate relays in trunk and junctor circuits which do not require access at electronic speeds.

Functionally, both units are large decoders. They are called "peripheral" because they can be located anywhere within the central office and reached from the central control via the peripheral bus. The information is transmitted on the peripheral bus in a coded form. It consists of several groups of 1-out-of-n codes. It is the basic function of both types of distributors to translate this information into 1-out-of-m form, where m is the total number of outputs of the respective distributor.

The decoding and associated functions have to be done in the most economical way which meets the requirements of speed, output level and reliability. The associated functions include receiving and temporarily storing the address information, checking it for errors, and blocking further action while notifying the central control if errors are found.

If errors in the address are not found and the signal indicated by the address is sent out, both distributors in most cases receive a positive indication from the activated point that the decoded signal has been properly received and acted upon. A failure to receive such indication is reported to the central control.

After a failure report, or as a matter of preventive maintenance, the central control may transmit maintenance or diagnostic orders to the respective distributor either to assure that the unit is in proper operational order or to localize a failure.

4.2 *Decoding and Verifying*

The number of outputs required in a distributor is between 512 and 1024. In these sizes, economy can be achieved only if the least expensive devices are provided for each output point. At the present state of the art, there is no electronic device that can economically compete with a contact on a large relay as a decoding element. For this reason the decoding in the signal distributors, where the required access cycle is about 20 msec, is performed by relay contacts.

Where the required access rates exceed the capabilities of relays, electronic devices have been employed. There are several possible arrangements that could provide a decoder capable of operating at electronic speeds in the microsecond range.

In the No. 1 ESS, a diode-transformer gate was chosen as the decoding element in the central pulse distributor (see Fig. 10). One of the reasons is that the transformer provides a balanced output. A pulse from such an output point can be transmitted over twisted pair to remote locations without interference. Also, the pulse provided by an electronic pulser can be shaped so that the least amount of noise is generated in the adjoining cables.

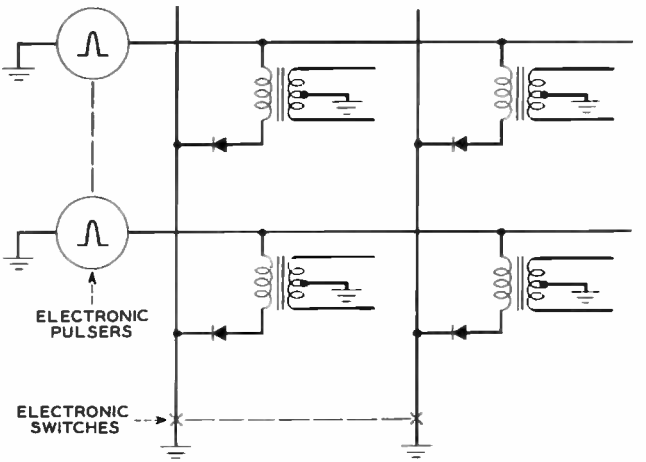


Fig. 10 — Transformer-diode selection matrix.

Another advantage of the transformer-diode arrangement is that bipolar pulses can be easily generated and transmitted by employing a three-winding transformer (see Fig. 11). In many cases, this is done to control the operation and release of a relay over a single pair of wires by using a receiving device that can recognize the two polarities. In the ESS such a device, called a "bipolar flip-flop" (see Fig. 11), is widely employed. This flip-flop assumes one state upon receiving a positive pulse and the other upon receiving a pulse of opposite polarity.

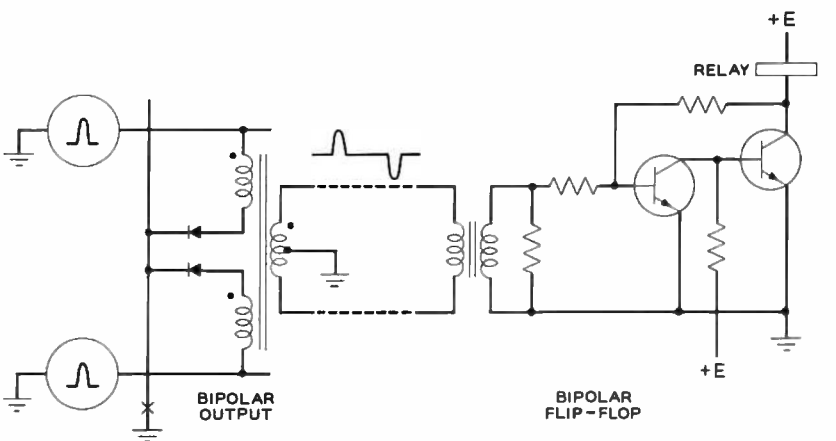


Fig. 11 — Bipolar operation.

The use of a three-winding transformer also provides for the verifying requirement. Whenever the signal transmitted from the central pulse distributor is to enable a peripheral unit for receiving an address from the central control on the peripheral bus, an answer that the enabling has been properly initiated is expected. Such an answer could be transmitted on a separate cable pair. However, the cable pair that transmits the enabling order is vacant at the time the verify answer should come. By employing the third winding on the transmitting transformer, the verify answer can be received over the same cable pair (see Fig. 12).

4.3 *Address Storage*

All the signals on the peripheral bus are in the form of nominal 0.5-μsec pulses. In the signal distributors, the information received from the bus has to be checked for errors before it is acted upon. Some kind of information storage is therefore necessary. The length of storage depends upon the time required to carry out the function of the unit.

In the signal distributor, the information has to be stored until the magnetically latching output relay has definitely operated or released. This time is somewhere between 10 and 20 milliseconds. A bistable transistor circuit (a flip-flop; see Fig. 13a) is employed to perform the storage function in the signal distributor. This flip-flop must have the proper input gating circuitry to permit reception from either of the two buses.

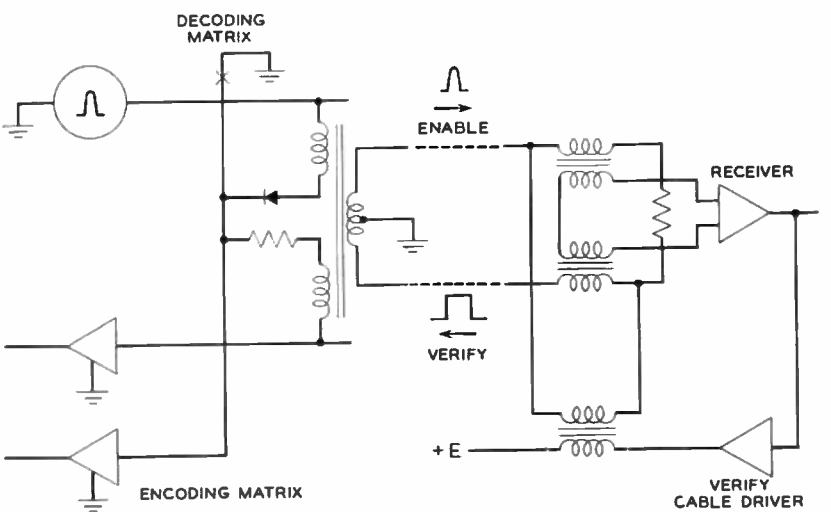


Fig. 12 — Enable-verify operation.

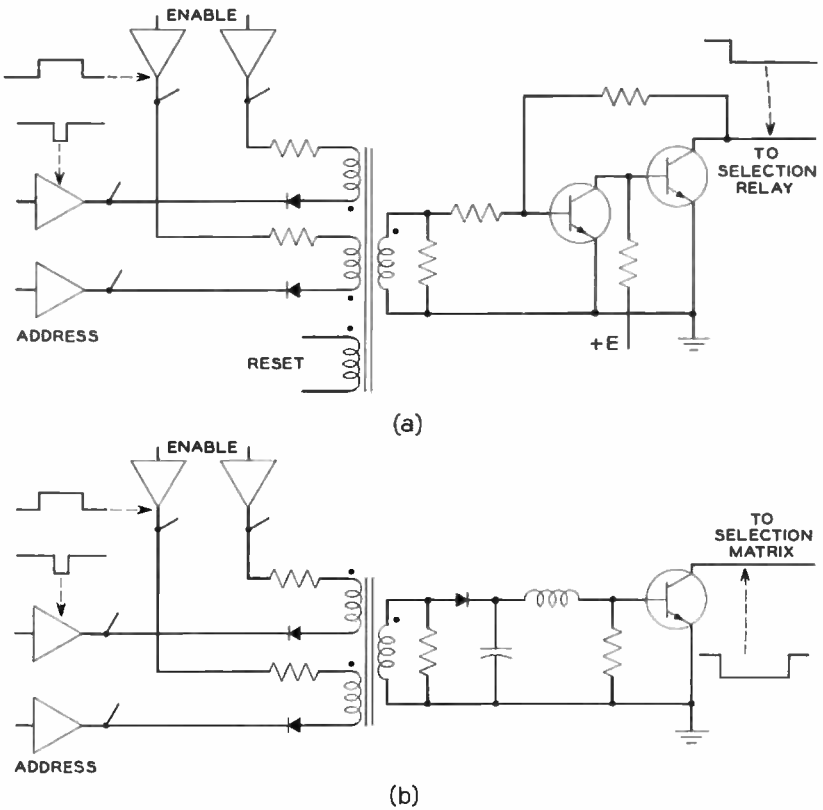


Fig. 13 — Address storage registers.

It also should be capable of operating the primary decoding relays directly, and should use the minimum of components. A special flip-flop has been designed that employs transformer-diode gating for the input signals and has the primary relay as a collector load without buffer stages.

The primary relays operate the multicontact wire spring relays that actually establish a path within the decoding matrix to the output relay.

In the central pulse distributor, the address need only be stored for about 2.5 μsec to perform the error checks and produce a nominal 0.5-μsec output pulse. A device simpler than a flip-flop can be employed. A storage circuit that simply stretches and amplifies the input pulse is used (see Fig. 13b). The input gating is quite similar to that in the signal distributor storage circuit. The stretching is accomplished by a diode-capacitor-inductor combination, which provides a rather exact timing

virtually independent of variations in the input pulse and supply voltage. The transistor acts as an on-off switch and amplifier. The transistor in the register cell controls the decoding matrix.

4.4 *Error Check*

The address information received by the distributors from the central control may contain errors. Errors may be caused by noise in the transmission path. There may be translation errors in the central control, or a faulty component in the sending or receiving circuitry may cause an error. Finally, errors may be purposely introduced in the address to check the error checking circuitry.

The error checking circuitry must find the errors before the action indicated by the address is carried out. The address is transmitted in groups of 1-out-of-n codes. For instance, the address to the central pulse distributors is in the form of 1-out-of-8, 1-out-of-8, and 1-out-of-16 code, a total of 32 pairs. A 1024-point signal distributor receives its address in a 1-out-of-8, 1-out-of-8, 1-out-of-4, 1-out-of-2, and 1-out-of-2 code.

For an address to be correct, each group should contain one and only one active signal. A signal completely missing in a group would constitute an error, as would two or more signals within a group.

In the signal distributor, the check for errors can be performed by employing the proper combinations of spare contacts on the selection relays, as shown in Fig. 14.

The final selection relays have only make contacts. If a signal is missing within any group and no relay within that group operates, a path cannot be established through the selection matrix. If more than one relay within a group operates, a double path will be established and a wrong output point activated. To prevent this, the contacts of the relays are arranged in a combinational network that will provide a path whenever more than one relay operates.

The presence of such a path within any group will prevent the application of the final output current and notify the central control about the nature of the error.

The central pulse distributor has been designed to find an error in the address within less than a microsecond, and electronic circuits have to be employed. The error check is performed at the outputs of the register cells in the three groups.

It is a simple matter to check that at least one register cell within a group has been activated. An OR gate is provided with as many inputs as there are cells in that group and one input connected to each cell. The output of the OR gate indicates that there is at least one activated cell

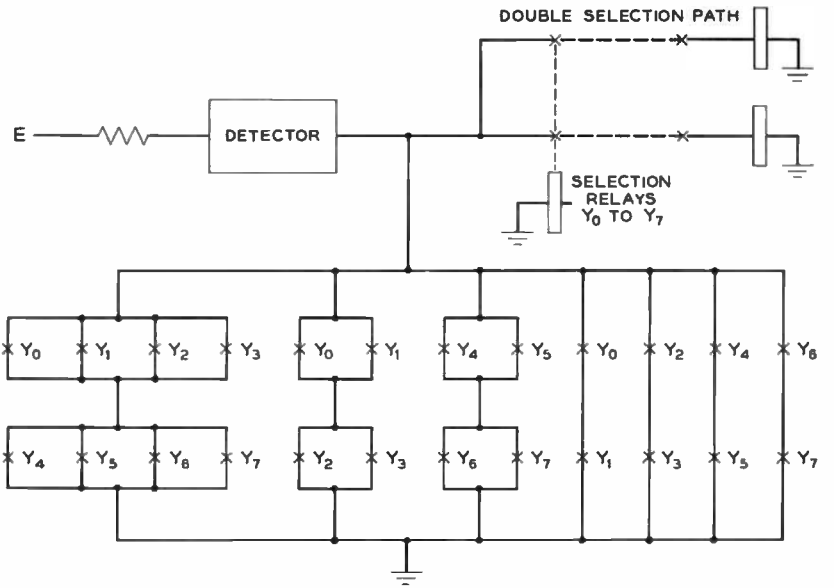


Fig. 14 — Error checking circuits for more than one signal in SD.

in that group. To ascertain that there is no more than one cell activated within that group, a rather elaborate arrangement of logic gates is necessary. For a 1-out-of-n group with $n = 16$, such an arrangement becomes prohibitive. In the central pulse distributor, therefore, a digital-to-analog conversion method is used (see Fig. 15). Let each addressed cell within a group control a unit of current $-i$. Let these currents be added in a low-impedance discriminating circuit of sufficient accuracy to distinguish between one and two units of current. Then the outputs from the discriminating circuits properly combined with the outputs of the OR gates will indicate that there is an error in the address, block the final action, and notify the central control about the type and location of the error.

4.5 *Final Outputs*

If no errors in the address are indicated, the distributors proceed in producing the final output. In the central pulse distributor, the three groups of register cells have already prepared a path in the three stages of selection matrix (see Fig. 16). A properly shaped pulse is now applied at the apex of the matrix and finds its way to the selected output. As

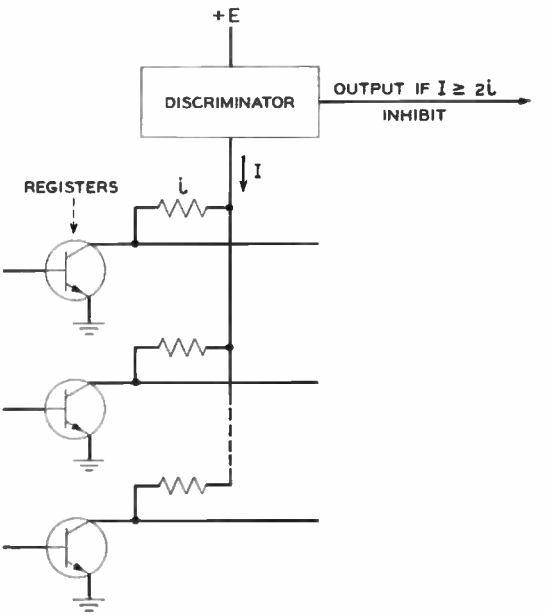


Fig. 15 — Error checking circuits for more than one error in CPD.

it proceeds through the final selection stage, a measurement of the pulse amplitude is made. If the amplitude is within the proper limits, an all-seems-well (ASW) signal is sent to the central control, indicating that the final action has been successfully completed. Absence of the ASW signal indicates either a missing or excessive output pulse. This causes the central control to initiate a trouble localizing or maintenance routine.

The final output element in the signal distributor is a magnetic latching relay. The pulses from the signal distributor have to be long enough and of the proper polarity either to operate or release the relay. The relay then will maintain this state until the next signal. There is no holding current required, which constitutes an appreciable power saving. The latching feature is obtained by making the armature of a mildly remanent material. The duration of the release current, however, must be sufficiently controlled to prevent a reversal of the field during release. A spare contact on the relay is used to verify the operation of the relay. It cuts in and out a shunt resistor, and the change in current is monitored at the apex (see Fig. 17) of the selection matrix. The proper change is interpreted as a successful operation and terminates the pulse to the relay.

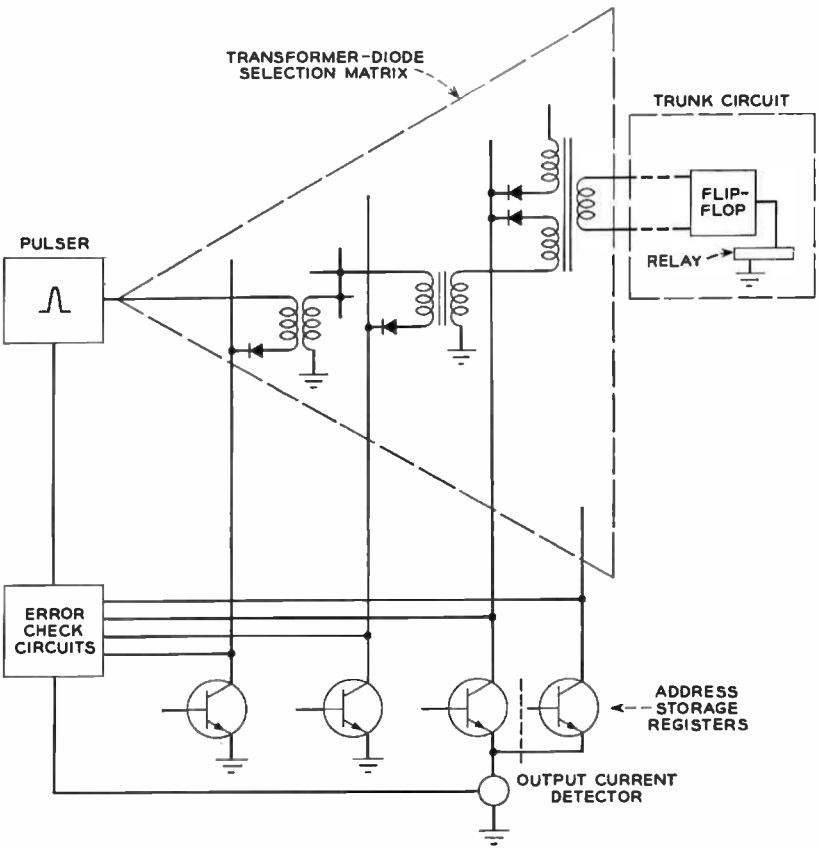


Fig. 16 — Selection in the CPD.

4.6 *Duplication*

The design of the distributors must provide sufficient reliability. A situation where failure of a component would disable the whole system cannot be tolerated.

To insure such reliability, duplication is employed. The central pulse distributors are always provided in pairs. The outputs are either connected together in such a way that either unit can send a signal to the remote point, or, when more than one output is needed for a function, the outputs are equally distributed between the two units. In the signal distributors, the control circuits are duplicated in such a way that either one of them can control all the outputs.

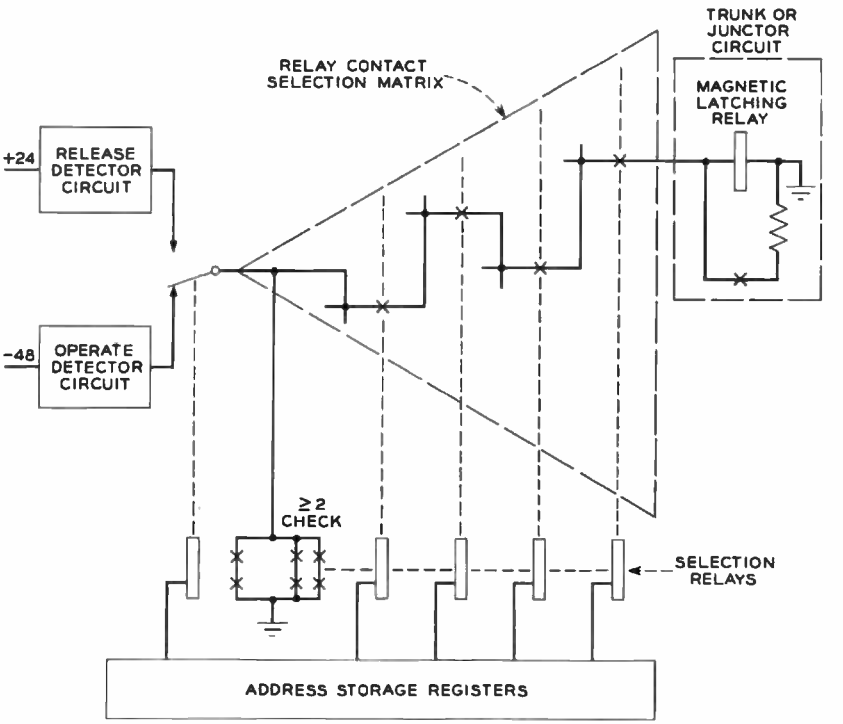


Fig. 17 — Selection and verification in the SD.

4.7 *Equipment Arrangement*

The signal distributors are incorporated in junctor and trunk frames with the magnetic latching relays located in the individual trunk and junctor units. The duplicated control equipment, comprising electronic packages, mercury-wetted contact relays, and wire spring relays, occupies one half of an ESS rack and is shown in Fig. 18.

The central pulse distributors are in their own frames, which also include peripheral bus signal repeaters. A typical central pulse distributor is shown in Fig. 19.

Equipment design details are described in associated articles.[3,6]

4.8 *Conclusion*

The basic principle of No. 1 ESS has been to minimize the varieties of subsystems, while providing the necessary functions in the most econom-

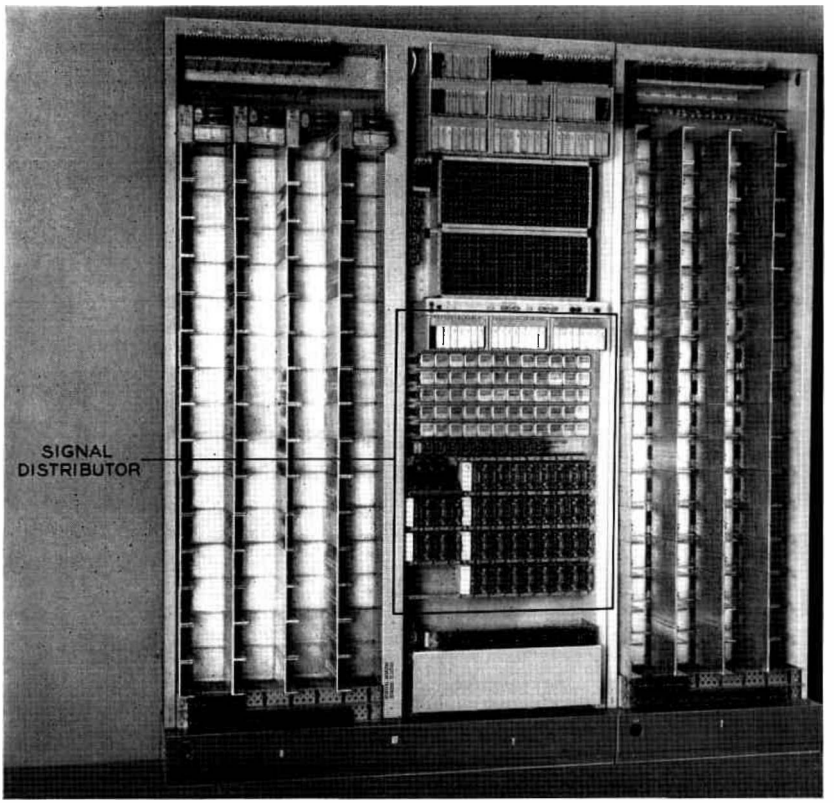


Fig. 18 — Signal distributor in trunk frame.

ical manner. The division of the action subsystems into two categories — signal and pulse distributors — which was dictated at the time of development by the existing state of the art, does indeed provide the necessary functions in the most economical manner.

V. ACKNOWLEDGMENTS

The design and development of the peripheral units described above required the efforts of many people outside the electronic switching area. Particular thanks are expressed to J. A. Baldwin, who collaborated in the ferrod conception, and to H. J. Wirth and R. F. Glore, who contributed the mechanical design of the ferrod. Thanks are expressed to F. P. Balacek, T. G. Grau and A. K. Spiegler for their development of the magnetic latching relay. Within the electronic switching area of Bell

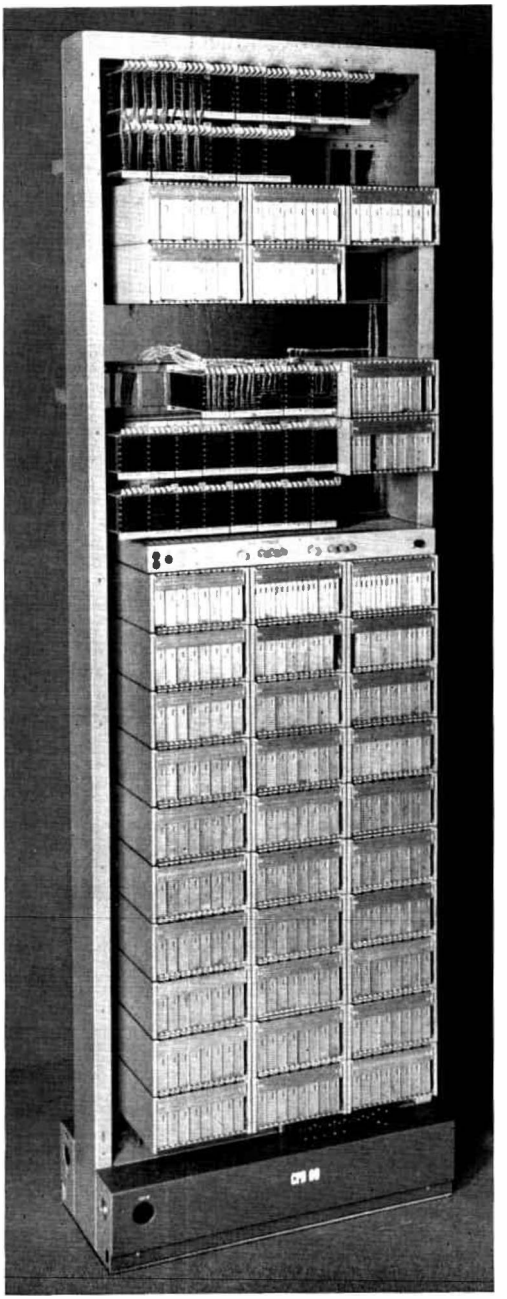


Fig. 19 — Central pulse distributor.

Laboratories, the fine equipment engineering and the unstinting cooperation of many other members of technical staff and technical aides made these developments possible.

REFERENCES

1. Danielsen, D., Dunlap, K. S., and Hofmann, H. R., No. 1 ESS Switching Network Frames and Circuits, B.S.T.J., this issue, p. 2221.
2. Dougherty, H. J., Ridinger, P. G., Raag, H., and Stockert, A. A., Line, Trunk, Junctor and Service Circuits for No. 1 ESS, B.S.T.J., this issue, p. 2321.
3. Ferguson, J. G., Grutzner, W. E., Koehler, D. C., Skinner, R. S., Skubiak, M. T., and Wetherell, D. H., No. 1 ESS Apparatus and Equipment, B.S.T.J., this issue, p. 2355.
4. Connell, J. B., Hussey, L. W., and Ketchledge, R. W., No. 1 ESS Bus System, B.S.T.J., this issue, p. 2021.
5. Baldwin, J. A., A Magnetic Device for High-Speed Sensing of Small Currents, AIEE Transactions, Part I, Communications and Electronics, March, 1961.
6. Chevalier, J. G., and Eisenhart, R. K., No. 1 ESS Circuit Packs and Connectors, B.S.T.J., this issue, p. 2441.

No. 1 ESS Master Control Center

By H. J. DOUGHERTY, H. RAAG, P. G. RIDINGER and A. A. STOCKERT

(Manuscript received January 21, 1964)

The No. 1 ESS master control center (MCC) serves as the interface between the switching system and operating telephone company personnel. It includes facilities for system test and control, alarm indication, maintenance of lines and trunks, recording customer billing information, and writing information on program store memory cards. The MCC is thus the central maintenance, control, and administration point of No. 1 ESS; it can be functionally divided as follows:

(a) maintenance teletypewriter (TTY) facilities, the primary communication facilities between the system and operating company personnel. Other TTY channels provide for communication between the system and a number of other operating company areas (such as traffic measurement, line assignment, etc.). The description of all of these TTY channels is included.

(b) alarm, display, and control circuitry to provide continuous indication of system status and permit maintenance personnel to control the system.

(c) trunk and line test (TLT) facility provided for maintenance of trunks, lines and service circuits. Included are facilities for dc loop testing, transmission testing, circuit make-busy, handling permanent signals on lines, etc.

(d) automatic message accounting (AMA) facility, providing a magnetic tape record of all data related to billing customer calls. The tape is processed in an accounting center to determine the customer charges.

(e) program store card writer used for periodically updating the program store translation information. This machine is described in a companion paper.

1. TELETYPEWRITER FACILITIES IN NO. 1 ESS

1.1 *Introduction*

To maintain and administer the No. 1 electronic switching system (ESS), communication facilities must be provided to exchange information be-

tween the switching system and operating company personnel. The flexibility and self-diagnostic features of No. 1 ESS are exploited fully[1,2,3] and result in an especially large volume of information exchange for effecting changes in subscriber service, reporting trouble conditions, etc. This makes it essential that the communication facility be flexible and convenient to use. The teletypewriter, together with the No. 1 ESS stored program and large memory capacity for storage of input and output messages, meets these requirements. Other major factors which influenced the adoption of the teletypewriter as the principal communication link with No. 1 ESS were:

(a) the teletypewriter is a standard device and relatively reliable because of the vast amount of development work already done in this field;

(b) the teletypewriter, with a standard English keyboard, lends itself readily to the large number of input-output messages required in maintaining and administering a large system such as the ESS;

(c) no particular training is required to use a teletypewriter for the vast majority of input messages to the ESS;

(d) the translation of English or English mnemonics to binary and vice versa is a relatively easy task for the ESS.

Keys and lamps are also provided as input-output devices for certain special or highly repetitive but simple functions.

1.2 *Maintenance Teletypewriters*

The two primary teletypewriter channels provided in the ESS are associated with the master control center (MCC). One of these teletypewriters serves as the basic communications channel between the ESS and maintenance personnel for normal everyday use. This machine is permanently mounted as part of the MCC; see Fig. 1. The second teletypewriter associated with the MCC may be located near the MCC or in some remote maintenance center or maintenance bureau. This second teletypewriter will always be located at some remote attended point if the central office is to be unattended at any time. This machine, besides serving as a communications channel for the maintenance men, also serves as the alarm broadcasting facility for the unattended office.

The normal output messages from the system will consist of alarm indications of various types, messages indicating troubles within the system, results of any self-diagnosis resulting from detected troubles within the system, and traffic overload conditions, as well as answers to questions asked of the system by the maintenance men.

Input messages to the system are of two general categories. First, the maintenance man may interrogate the system concerning a number of

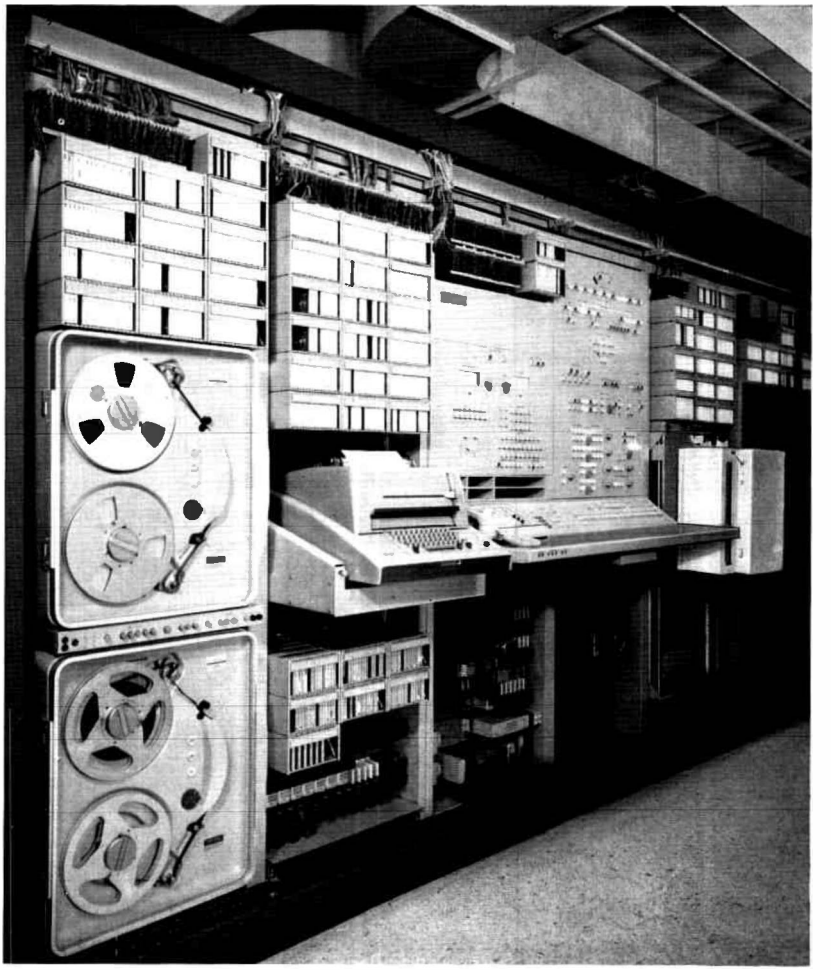


Fig. 1 — No. 1 ESS master control center.

specific areas. That is, he may ask the system to print out the contents of a particular memory location or to trace a call through the network from a particular line or trunk. He may also ask the system for the information in its memory concerning a particular line or a particular trunk. That is, he may give the system, for example, a directory number and ask the system to print out the equipment number that the directory number is associated with and all other information the system has pertaining to that directory number. Second, the switchmen may order the system to perform a diagnostic test on some part of the system.

1.3 *Service Order Teletypewriters*

A teletypewriter arrangement is also provided as the input channel for service order information. In this arrangement a teletypewriter tape reader is located in a remote service order bureau or any other location convenient for operating company personnel. This ESS teletypewriter channel accepts service order information in the field-oriented format used by any operating company. On receipt of the teletypewriter signals, the ESS edits the service order information insofar as it is able. For example, if a service order indicates that a directory number is to be assigned to a new customer's line and, in fact, the directory number is already in use, the system will detect this error. When any service order information appears to be in error, the ESS will ignore the information and will indicate the existence of the error and the service order number on the MCC teletypewriter. If the information is acceptable, the system stores the information until it is activated by operating company personnel. The service order is activated when telephone company personnel dial the service order number over a special telephone line, thus instructing the system to start using the translation information.

Provision is also made for another teletypewriter to be connected to this service order input channel and located near the main frame in the central office. When provided, this teletypewriter will receive the same service order information as received by the ESS from the remote point. The copy from this machine can then serve as the service order for main frame cross wiring.

1.4 *Traffic Usage Teletypewriter*

Another teletypewriter arrangement is provided in the ESS for indicating the contents of a large number of traffic usage registers. These traffic registers in the ESS memory are functionally analogous to the mechanical traffic registers used in all electromechanical switching systems. This traffic usage teletypewriter may be located at any point remote from the ESS. The ESS is programmed to identify each of these registers on the remote teletypewriter and to indicate its contents in summarized form. These register counts are reported periodically on the remote usage teletypewriter: the contents of some registers will be reported every half hour. In other cases, depending upon their function, some registers' contents will be reported every four hours.

This teletype channel and the associated ESS program are also arranged to permit traffic administrative personnel to interrogate the system concerning the contents of specific registers. Certain traffic overflow conditions encountered are also reported on this teletypewriter.

1.5 *Line Trouble Teletypewriter*

The final teletypewriter arrangement provided in ESS serves three purposes. This one-way channel is utilized by the system to transmit information to a remote maintenance bureau concerning permanent signals on lines, the results of automatic line insulation testing (ALIT) and the results of tests performed on pressurized cable contactor pairs. Permanent signal information is normally transmitted by the system periodically on a timed interval. However, in the event a large number of permanent signals occur in a short time, a printout is initiated as soon as information on them has been collected. The results of the ALIT tests and cable contactor tests are printed whenever failure information is available.

1.6 *Call Store Buffer*

In communicating with peripheral units, central control is able to transmit or receive a multibit word every 11 μsec.[1] By contrast, the teletypewriter can send or receive information at only one bit every nine msec. To bridge this speed barrier, a call store buffer is provided. The functional form of the buffer and its relation to the teletypewriter channels are shown in Fig. 2. It consists of two stages: a first stage provided on a per-channel basis and a second stage common to all channels.

The first stage of the buffer includes the teletypewriter channel buffer (TCB) and the teletypewriter channel control (TCC) registers. These registers are used to assemble and process both input and output messages. The TCB register can store up to 60 TTY characters. This corresponds to one line of type from the teletypewriter. As each incoming character is received from the teletypewriter, it is deposited in the TCB register by the control program. After the complete message has been assembled, it is withdrawn from the TCB register, converted to ESS machine language, and delivered to the proper client program. Records necessary for administering the TCB register are stored in the associated TCC register.

The second stage of the buffer is used as an assembly area for outgoing messages and is designated the "outgoing message buffer area" (OMBA). Records for the administration of the OMBA are kept in its administration area (AA). The OMBA has a capacity of 200 24-bit words. It contains two kinds of entries: a client request (CR) register for every output message and a data entry for clients requiring data storage. The destination channel, priority, and other output message information is stored in the CR register.

An output message from a client program is stored in the OMBA until

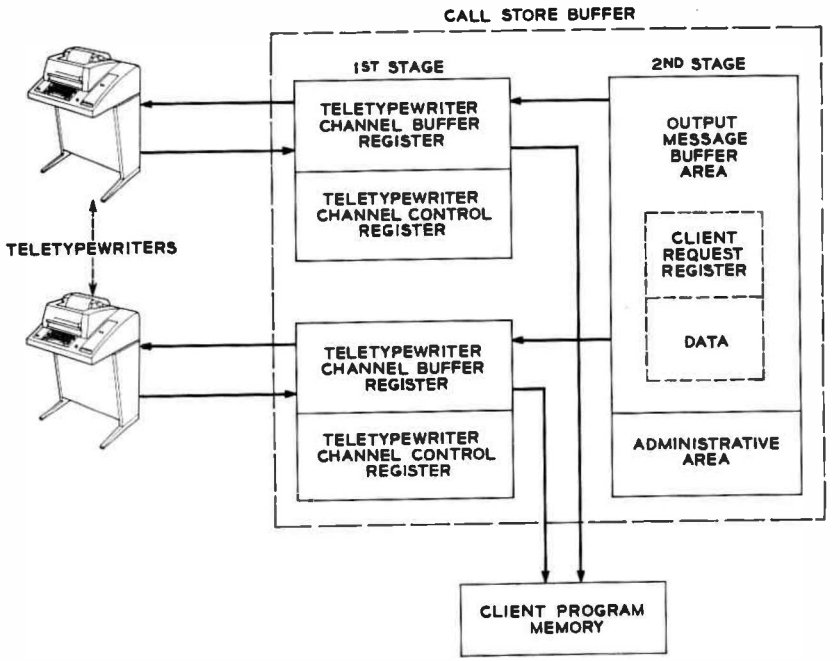


Fig. 2 — Call store buffer.

the desired destination channel is available. The message is then converted into TTY characters and loaded into the TCB register. Every 100 msec, the control program transmits a character stored in the TCB register to the teletypewriter. As this process continues, new characters are placed in the TCB register until the output message is completed.

1.7 *Input Message Control*

A special group of characters is used to instruct the program as to the proper treatment to be given each input message. Since these characters have control significance, they may not be used in the body of an input message. A partial list of the control characters and a brief description of the function of each are given below.

Dash (-): the dash is used as a part of the message identification code which precedes each input message to specify its destination and treatment. The message identification code consists of as many as 13 noncontrol characters and two dashes. The second dash indicates the end of the message identification code.

Execute (.): a period is typed at the end of each line to tell the pro-

gram to execute that portion of an input message. The contents of the TCB register are then transferred to the client's memory and the carriage is returned to the left margin in preparation for the next input or output.

Checkput (?): the checkput is a printback of an input message that has just been typed. It is used to check that the No. 1 ESS has properly received the input message. The checkput is requested by typing a question mark at the end of each line of input message. If the printback is satisfactory, a period is typed at the end of the line to tell the program to execute the message.

Backspace (←): the backspace character is used to correct typing errors. Each backspace that is sent causes the program to discard the last character stored in the TCB register. The program also moves the carriage one space to the right so that the correct character can be printed to the right of the incorrect character.

Begin again (↑): this character is used to correct a full line of type. In this case, the program discards the entire contents of the TCB register and then returns the carriage to the left margin so that a new line may be typed.

Abandon message (@): input messages that have not been terminated by typing the execute character may be discarded by typing the abandon message character. In this case, the program returns the channel to the idle state and ignores the preceding message.

After each input message is typed in, a two-character acknowledgment is printed out by the program. The acknowledgment appears on the same line as the input and indicates what action has been or will be taken by the No. 1 ESS. Finally, after each complete input or output message, the program sends at least three nonprint characters to the stunt box (see Section 1.8), which repositions the carriage and unlocks the keyboard for the next input or output message.

1.8 *The Model 35 Teletypewriter*

The Teletype Corporation Model 35 teletypewriter has been adopted for use with No. 1 ESS. This newly developed machine has a four-row keyboard similar to the standard office typewriter. It operates at 100 words per minute using a new seven-bit code based on the American Standard Code for Information Interchange (ASCII) recently approved by the American Standards Association. Sixty-four code combinations are assigned to letters of the alphabet, numbers, and symbols; thirty-five combinations are used for control purposes; and the remaining twenty-nine combinations are unassigned.

The line signals generated and received by the Model 35 teletypewriter are shown in Fig. 3. Each seven-bit character is transmitted with a start bit, an unused bit, and two stop bits. Since the nominal bit interval is 9.09 msec, one character may be sent every 99.99 msec.

Like its predecessors, the Model 35 teletypewriter is equipped with a stunt box which is able to decode certain teletypewriter characters and activate a set of contacts. In No. 1 ESS, these contacts provide a convenient method of controlling external relay equipment. For example, the stunt box is used to actuate audible and visual alarms in the maintenance center when alarm messages are transmitted to the remote maintenance teletypewriter.

The Model 35 teletypewriter may also be equipped with an answerback circuit. When triggered by a stunt box contact, this circuit sends a unique sequence of characters back over the signal line as an indication that the machine is able to transmit and receive information.

1.9 *The Transmit-Receive Unit*

Since the No. 1 ESS is a word-organized system, information is transferred between functional units in parallel form. To permit communication with the teletypewriter which sends and receives serial information, a transmit-receive (TR) unit is interposed between the central control and the teletypewriter. The TR unit accepts parallel information from central control and transmits it serially to the teletypewriter. Conversely, the TR unit receives serial information from the teletypewriter and converts it to a parallel form acceptable to central control.

The functional form of the TR unit is illustrated in Fig. 4. Serial-to-parallel or parallel-to-serial conversion of each TTY character is carried out by means of a shift register. Since full-duplex operation is not required, the same shift register is used both to transmit and receive information.

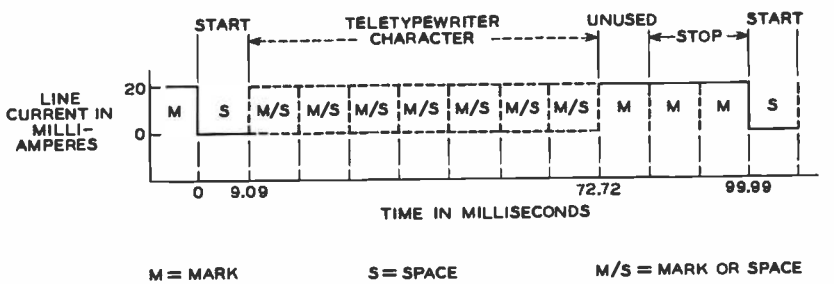


Fig. 3 — Model 35 teletypewriter line signals.

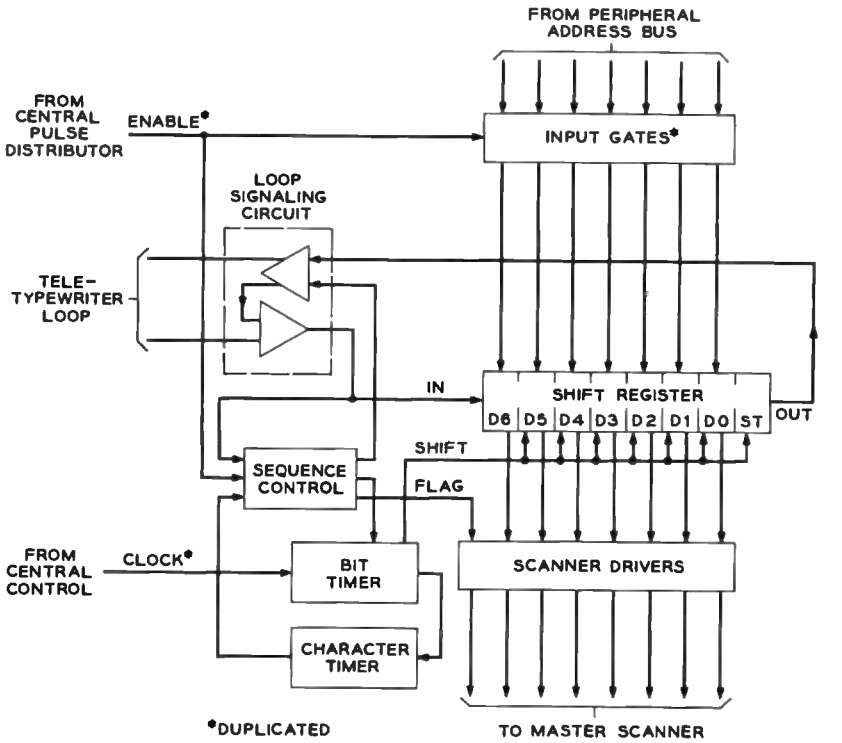


Fig. 4 — Transmit-receive unit.

The TR unit must also generate timing signals which define the duration of each bit and the end of each seven-bit character. This timing information is derived by counting down a clock signal supplied by central control every 71.5 μsec. These clock signals drive a binary counter (designated the "bit timer") which is arranged to recycle after 127 clock pulses or 9.085 msec. Each time the bit timer recycles, it increments another binary counter (designated the "character timer") which in turn recycles after the start bit and seven character bits have moved into or out of the shift register.

Parallel information from central control is delivered to the TR unit via the peripheral address bus (PADB).[4] Central control is able to transmit to the TR unit over either PADB by enabling the appropriate set of input gates. The enable signals are supplied from the central pulse distributor[5] over two private wire pairs.

Serial information from the teletypewriter is temporarily stored in the shift register and is available to central control via scan points in the

master scanner.[5] The teletypewriter sends and receives information in the form of mark and space signals which appear on the signal line as 20 ma of current or zero current respectively. These signals are generated and detected in the TR unit by a transistorized loop signaling circuit. The detector has a decision threshold of 10 ma and a low-pass filter in its input to reject high frequency noise.

The operational sequences normally carried out by the TR unit are initiated by external signals received from either the central control or the teletypewriter. Resulting actions inside the TR unit are governed by the sequence control.

1.9.1 *Normal Operation*

1.9.1.1 *Receiving Sequence.* A receiving sequence is initiated by a start signal from the teletypewriter, which is detected by the loop signaling circuit. The sequence control times for half a bit interval and then samples the line signal. If the line signal is now a mark, the preceding space signal is interpreted as a hit due to line noise and the TR unit returns to its normal state. On the other hand, if the line signal is still a space, it is accepted as a legitimate start signal and the TR unit flags central control. At the end of the start bit and each successive bit, a shift pulse generated by the bit timer gates the line signal into the register and shifts each bit in the register to the next higher stage. At the end of the seventh character bit, the character timer recycles and the sequence control deactivates the flag ferrod, indicating to central control that a full character has been received. The sequence control then times for two more bit intervals, after which it returns to the idle state. During the receiving sequence, the TR unit inhibits the input gates from the PADB to prevent an improper read-in from the central control due to a false enable signal.

Central control scans the TR flag every 25 msec; after detecting the flag rise and fall, it interrogates the scan points which monitor the seven character bits stored in the shift register. The full TTY character remains stored in the shift register from the end of the seventh character bit until the end of the next start bit. After the character is read out, it is placed in a buffer register in the call store until the full input message is assembled.

1.9.1.2 *Transmitting Sequence.* Output messages are transmitted by the central control to the TR unit at the rate of one character every 100 msec. A transmitting sequence begins with an enable signal from central control. The seven character bits are gated into the shift register and an enable verify signal is returned to central control.

The sequence control now takes charge of the TR unit and begins to transmit information serially to the teletypewriter. The output of the last stage of the shift register is gated to the loop signaling circuit and the start bit transmitted by sending a space signal for 9.085 msec. Then each of the seven character bits is gated to the loop signaling circuit by a series of shift pulses generated by the bit timer.

The loop signaling circuit transmits a mark or space signal, depending on whether a one or a zero is stored in the last stage of the register. After each bit is applied to the signal line it is gated back into the first stage of the register. Thus, by recirculating the transmitted character, central control is able to check the TR unit's ability to properly transmit information.

After the last character bit is transmitted, the character timer recycles and the shift register is disconnected from the loop signaling circuit, leaving the signal line in a continuous marking condition. The sequence control times two more bit intervals and then returns to the idle state. During the transmitting sequence the input gates from the PADB are again inhibited to protect the information in the shift register from mutilation by a false enable signal.

Although teletypewriter signals can flow in only one direction at a time, there is still the possibility of simultaneous seizure of the TR unit by both the teletypewriter and central control. In this case, the TR unit favors the teletypewriter by permitting the start signal to override the enable signal from central control. The TR flag is activated and central control is able to defer its output message until the input message is completed. This same provision permits the maintenance man to interrupt central control at any point in a transmitting sequence simply by depressing the break key at the teletypewriter. This action produces a long space signal which fills the shift register with zeros and is recognized by central control as a request to send an input message. After the input message is received, central control resumes sending the output message previously interrupted by the break signal.

1.9.2 *Trouble Detection*

Central control periodically checks the TR unit's ability to transmit information by comparing the recirculated character with the character previously transmitted. If a discrepancy is found, central control activates a central pulse distributor point which places the TR unit in the quarantine mode. This mode is designed to permit fault recognition and diagnosis. While in quarantine, the TR unit is divorced from the teletypewriter, although the recirculation path is maintained and continu-

ous marking current is supplied to the signal line. Five of the seven scan points that normally monitor the shift register are reassigned to critical points in the character timer and sequence control. The other two scan points continue to monitor the fourth and seventh stages of the shift register. Central control may then initiate a transmitting sequence and observe changes of state in the sequence control, recycling of the character timer, and signal flow through the shift register.

The ability of the TR unit to receive information is also checked on entering the quarantine mode by momentarily opening the input to the loop signaling circuit. The TR unit interprets this action as a valid start signal and executes a receiving sequence. At any time during a diagnostic routine, central control may monitor the full content of the shift register by returning the TR unit to its normal mode.

Fuse alarms, power removal, and abnormal supply voltage conditions are also reported to central control by means of scan point indications. During idle periods, central control may check the supply voltage monitors by artificially inducing high- and low-voltage conditions.

1.10 *The 105A Data Set*

Most of the teletypewriters associated with No. 1 ESS will be located at points remote from the central office. The permissible distance between the teletypewriter and its associated TR unit would normally be limited by the loop resistance of the connecting cable pair. To overcome this restriction, the dc teletypewriter signals are converted to ac tones by 105A data sets which are interposed between the teletypewriter and the TR unit as shown in Fig. 5.

Each data set consists of a frequency-shift modulator and demodula-

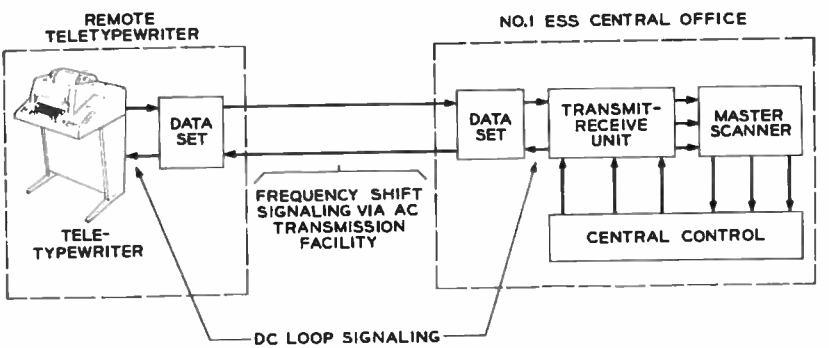


Fig. 5 — Remote teletypewriter channel.

tor. The modulator converts dc loop current or the absence of current into a continuous mark or space frequency. The demodulator detects the mark or space frequency and converts these signals to dc loop current or the absence of current. The data sets are arranged to transmit and receive in two different frequency bands, thereby allowing transmission in both directions simultaneously.

The strength of the carrier signal received by each data set is continuously monitored as an over-all check of the communications channel. If the received carrier falls below a predetermined level for more than 400 msec, the data set will change to its disconnect state and send a 750-msec space signal to the other data set. After timing the long space signal, this set will also change to the disconnect state. The loss of carrier is reported to the No. 1 ESS by means of a scan point in the master scanner activated when the data set is in the disconnect state. Loss of carrier on the remote maintenance channel is also reported as a major alarm by the data set in the attending maintenance center.

A special provision in the remote maintenance channel is the means to report a catastrophic failure of No. 1 ESS even though the system is unable to transmit an output message. In this case, the emergency-action timeout circuit in central control releases a relay which opens the transmission path between the two data sets. The subsequent loss of carrier is detected and reported as a major alarm by the data set in the maintenance center.

II. ALARMS, DISPLAYS, AND CONTROLS IN NO. 1 ESS

2.1 *Introduction*

The primary medium of communication between No. 1 ESS and maintenance personnel is the maintenance teletypewriter channels described in the previous section. When the system maintenance program detects a trouble, it diagnoses the unit and reports the location of the failure to the maintenance personnel via the maintenance TTY. After repairing the unit, maintenance personnel use the maintenance TTY to instruct the system to return the unit to service.

TTY communication, however, is not dependable when the system loses its self-organizing capability. When this occurs, the need to exert control over the system is imperative. To provide for this need, controls and displays are provided in No. 1 ESS equipment frames and on the alarm, display, and control panel at the master control center. An alarm system is used to alert maintenance personnel and direct them to the proper location for receiving data about the nature of the failure.

2.2 *Office Alarm System*

For any one floor in a building, the office alarm system (see Fig. 6) consists of an aisle pilot unit with a major alarm lamp for each equipment aisle, a group of three lamps for each main aisle, a vertical lamp holder near the exit with one lamp for each of the other floors in the building, and a panel containing four audible alarms and relay control equipment. Of the three lamps in the main aisle group, one is the main aisle pilot. This lamp indicates trouble in an equipment lineup off that main aisle. The remaining two are the "other floor lamps." These indicate the existence of troubles on other floors. The audible signal for major alarms is a tone bar and for minor alarms is a telephone ringer.

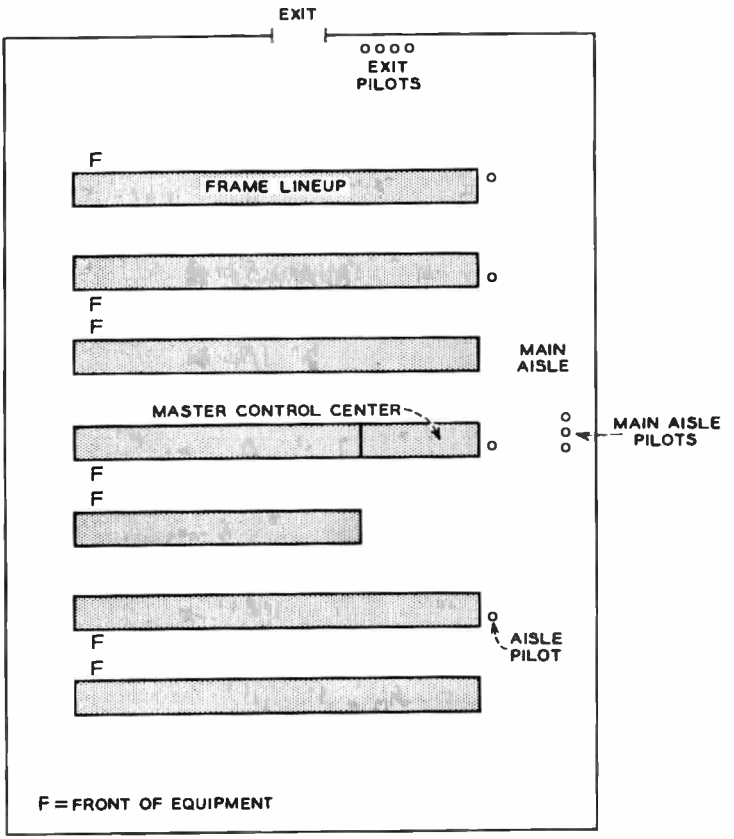


Fig. 6 — Office alarm layout.

The power room equipment does not contain a series of locating lamps, since it has its own alarm circuit providing both major and minor alarms. It is tied into the office alarm system, however, to the extent that for major alarms it lights the appropriate other floor lamp and rings a distinctive audible alarm on all floors. For a minor power alarm, it lights the appropriate other floor lamp and rings the regular minor alarm bell on the floor where power alarms are normally supervised. Either major or minor power alarms cause a separate exit lamp on each of the floors to light.

The office alarm system also contains an alarm circuit to indicate a failure in the supply that powers the alarm circuits themselves. A failure in the fuses that supply the alarm circuits causes a distinct audible alarm to be sounded, immediately indicating to the maintenance man the precise nature of the failure.

Each No. 1 ESS subsystem reports fuse failures via the office alarm system. Lamps located on the frame indicate the nature of the trouble. The maintenance programs, the most powerful diagnostic facility in No. 1 ESS, have access to the office alarm system at the MCC. When the program completes diagnosing a unit, it reports the failure to the MCC, which causes a signal to be placed on the alarm system. The maintenance personnel retire the alarm at the MCC and receive the dictionary printout[3] from the teletypewriter.

No. 1 ESS alarms are transferred to a distant office using a teletypewriter loop. The teletypewriter at the receiving office is integrated into the office alarm system by using teletypewriter stunt box contacts to activate remote alarms.

2.3 *Alarm, Display and Control Panel*

The alarm, display and control panel (see Fig. 7) is the centralized control point for No. 1 ESS. Lamps provide an over-all indication of the system status. Keys and pushbuttons provide means of exerting direct control over the system and its program. For convenience, this panel is divided into four sections. Each section will be described separately.

2.3.1 *System Status Display*

The lamps in this group display the status of individual units as classes of units. Whenever a trouble occurs, the maintenance man can quickly ascertain the seriousness of the failure by glancing at the lamp displays. Each central control has two lamps: one, labeled "TBL," lights when the unit is in trouble; the other. labeled "ACTIVE," lights when the unit

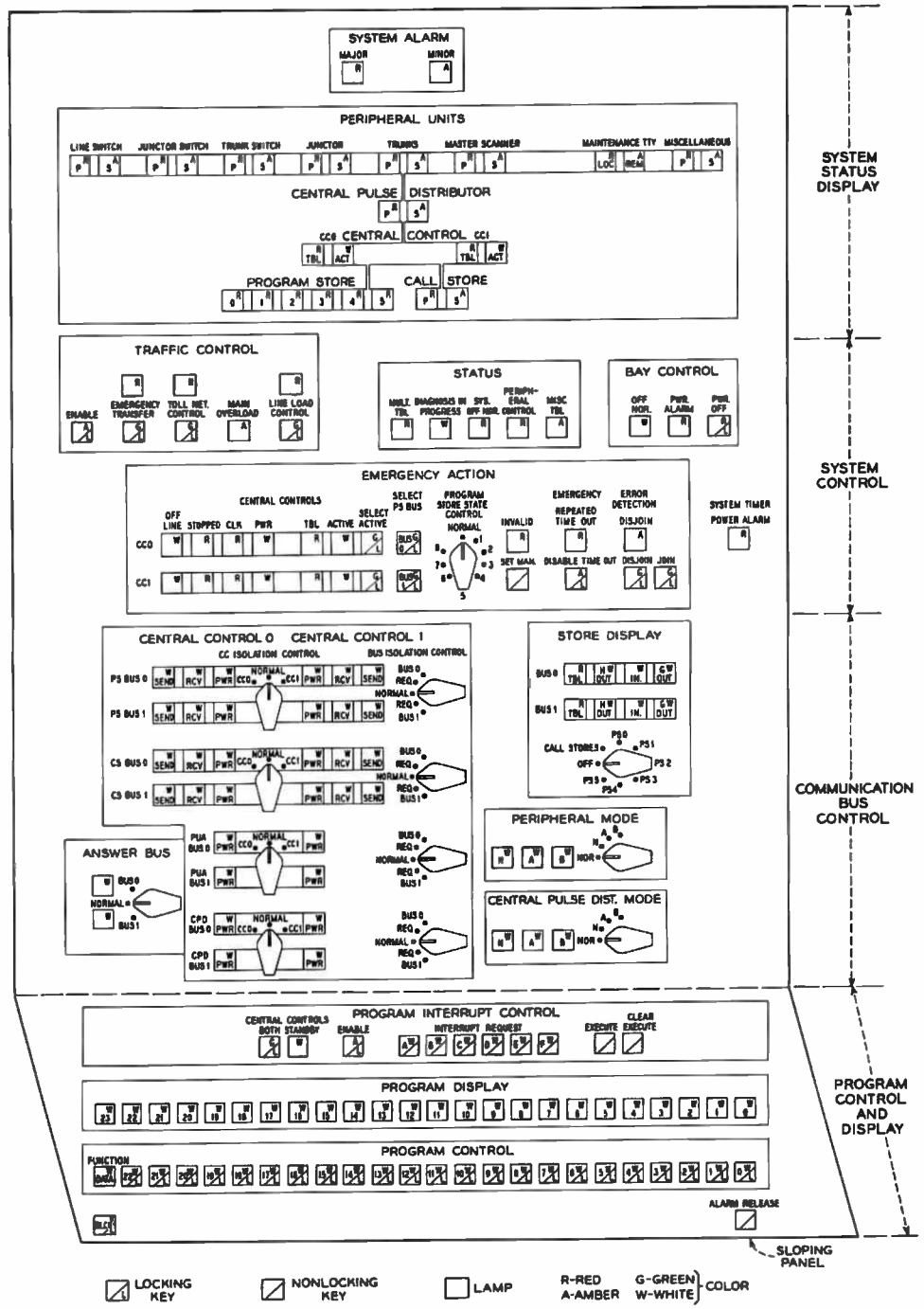


Fig. 7 — Alarm, display, and control panel.

is in active status. Each program store has a lamp that lights when the store is out of service. Two lamps are provided for line switch frames. The primary lamp P lights when both scanner controllers or both network controllers are in trouble in some line switch frame. The secondary lamp S lights when one scanner controller or one network controller is out of service in some line switch frame. Similar considerations apply to the other groups of units. In the call stores, the primary lamp lights when both copies of some information block are no longer available. The secondary lamp lights when only one copy of some information block is unavailable.

2.3.2 *System Control*

The system control is used to restore rudimentary self-adaptive capability to the system when the maintenance programs[3] cannot effectively restore order. The first indication that the system has lost program control comes when an emergency-action alarm is given. This alarm indicates that the system program has not passed prescribed check points within a given time interval. If this alarm is repeated several times (as indicated by the "REPEATED TIME-OUT" lamp), the system is deemed to have lost control. For instance, the system clock may have failed and a switch of central controls cannot be effected. When maintenance personnel are faced with this situation, they must take control over the system. Control is assumed via the "emergency-action" controls. Via these controls, the maintenance man can force a configuration of central controls and program stores, the basic data processing units in the system. He then starts the program. The program then attempts to recover an operational system. If the program cannot reestablish control it will again time out. Maintenance personnel then force a different configuration. This is repeated until the program establishes control.

Associated with the emergency action controls are lamps to provide feedback to the maintenance man. Points monitored include the clocks, power, active status, stopped, and trouble flip-flops in each central control.

2.3.3 *Communication Bus Control*

The keys and lamps in this group are used to control the interframe bus systems[4] and display their status. The lamps in this group display how the central controls (CC) are associated with the duplicated buses for program stores (PS), call stores (CS), peripheral units (PU) and central pulse distributors (CPD). For each CC, lamps indicate whether the

CC is transmitting to and/or receiving from the appropriate bus system. Each CC can also be isolated from any or all bus systems. Such control is desirable when the systems must be "split" for special test purposes. The buses can be isolated from each CC by operating the appropriate "BUS ISOLATION CONTROL" switch. Before control is applied the switch passes through an intermediate "REQ" position. This position, which is monitored by a scan point, gives the program an opportunity to mark the bus in trouble and take it out of service. This allows an orderly shutdown of the bus system.

2.3.4 *Program Control and Display*

Since the "software" in No. 1 ESS is significant in the operation of the system, some control over it is necessary. There exists a class of programs — system initialization, for instance — instituted by the maintenance man in emergency circumstances. The system can be forced into a number of these programs by operating the "program interrupt" keys.

The "program control" keys are used to insert data into the system. Conversely, the system can be requested to display data on the "program display" lamps. The program control keys are also used to control blocks of programs when the system program is being modified.

III. TRUNK AND LINE TEST PANEL

3.1 *Introduction*

The trunk and line test (TLT) panel, in conjunction with the MCC TTY, provides central office maintenance personnel with a facility for maintaining interoffice trunks, lines, and service circuits (e.g., dial pulse receivers, multifrequency transmitters, etc.).[6] The inherent ability of No. 1 ESS to perform logical actions under direction of a stored program suggests that the facility design should include a minimum amount of hardware. In general, the simplicity and flexibility of design in the TLT panel was obtained by exploiting the following:

(a) Compared to electromechanical switching systems, No. 1 ESS permits simplified trunk circuit design and, consequently, more trouble-free operation. Also, the types of trunk circuits required in the largest numbers are designed as plug-in modules for easy maintenance.

(b) No. 1 ESS is programmed to make a number of per-call checks on trunk circuits and service circuits in the process of handling a call. This arrangement allows the system to report circuits causing trouble via the maintenance TTY.

(c) If No. 1 ESS encounters trouble with a particular service circuit or trunk in the process of handling a call, it can in many cases complete the call by using another circuit and then determine the nature of the trouble and report it to the maintenance man via the TTY.

(d) Part of the simplicity in trunk circuit design stems from the fact that outgoing trunk (OGT) test jacks are not provided on a per-trunk basis. Test access to any trunk or service circuit is obtained by a switched connection through the switching network.

(e) Entire test sequences can be stored in the system program for various circuits and be rapidly initiated by the maintenance man from the TLT panel. The system can also be told to mark a trunk or service circuit busy in the memory, and consequently a make-busy key or jack per trunk is not required.

3.2 *General*

The TLT panel in conjunction with the local maintenance TTY adjacent to it serves as the major tool for the central office personnel for maintaining trunks and lines. The keys and lamps on the TLT panel are used for those tests and messages which occur most frequently in the course of maintaining trunks, lines, and service circuits. The same messages could as well be transmitted by the TTY, but the TLT panel provides a more rapid means of communication.

The block diagram in Fig. 8 shows the basic relationship between the TLT panel and the system. The equipment arrangement is shown in Fig. 9. A TOUCH-TONE key set, mounted on the key shelf, is used to transmit all information to the system. The output of the TOUCH-TONE key set reaches the system via the master test line (MTL) which appears as an ordinary TOUCH-TONE line on the switching network with a special class of service. The keyed information may consist of a trunk or service circuit number, a trunk number and a directory number to be outpulsed, etc. Once a trunk, line, or service circuit has been identified to the system via the MTL and the maintenance man has received appropriate supervisory or status indications (e.g., busy, idle, etc.), he can then exercise any of the test and maintenance features provided in the TLT panel.

3.3 *Trunk and Line Test Panel Features*

Trunk make-busy — If the maintenance man identifies an outgoing trunk by keying the proper identity number over the MTL, he can then, if the trunk is idle, operate a make-busy key to instruct the system to mark that particular trunk "maintenance-busy" in its memory. As long

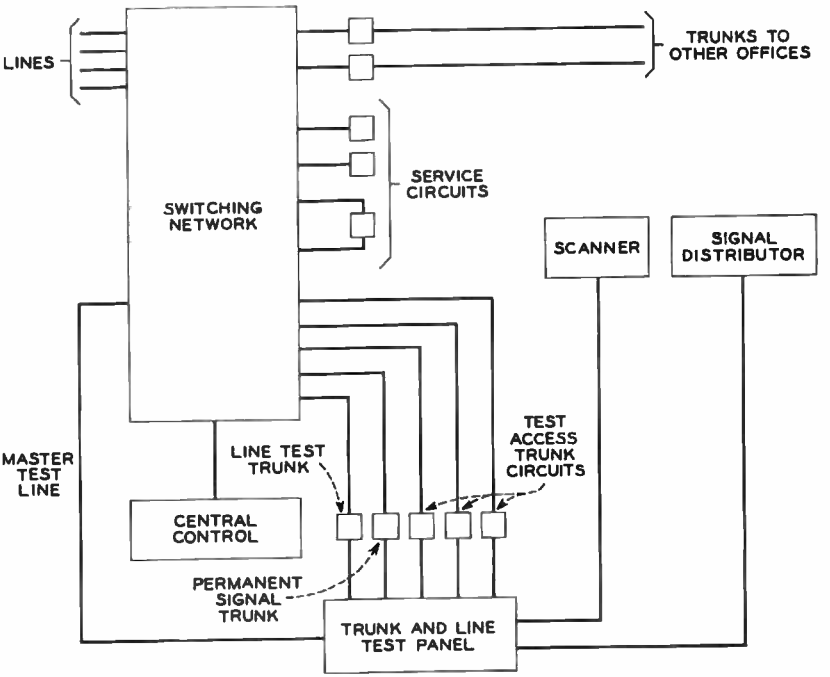


Fig. 8 — Block diagram showing basic connections between TLT panel and ESS.

as the trunk is marked "maintenance-busy," the system will not use the trunk for normal traffic. However, a connection can be established between the trunk and the TLT panel for test purposes. If the maintenance man identifies a trunk that is already in the "maintenance-busy" state, he can operate another key which instructs the system to restore that particular trunk to service. Whenever the system reacts to a make-busy or remove-busy order from the TLT panel, it also prints a message on the TTY identifying the trunk which was made busy or restored to service.

If the maintenance man wants to mark a trunk maintenance-busy but finds that it is traffic-busy, he can instruct the system to mark it maintenance-busy as soon as it becomes idle and to notify him by a message on the TTY. Thus, he can continue with other work until the trunk is available.

Test access trunks — Once a trunk or service circuit has been identified, the system can be instructed to connect it to any one of the three test access trunks. Thus, if the maintenance man identifies an outgoing trunk and causes the system to outpulse to a test terminal or test facility

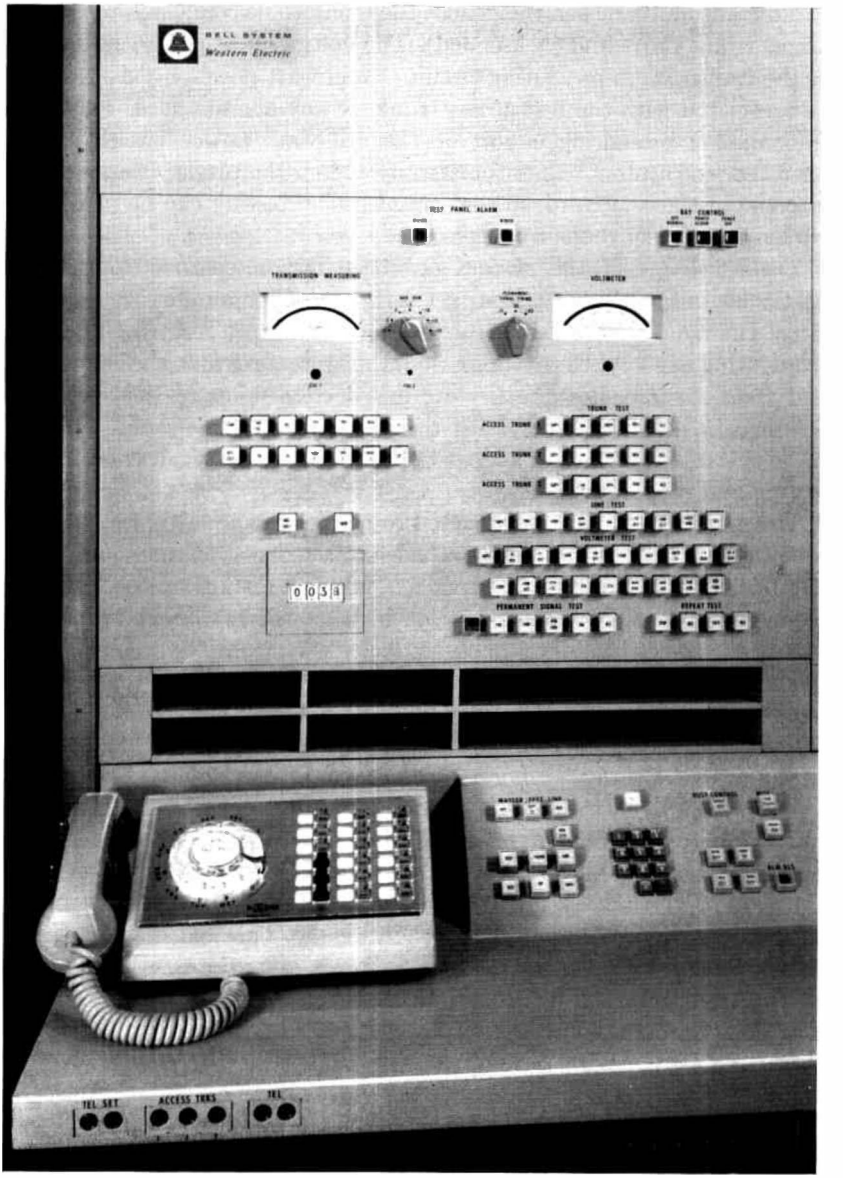


Fig. 9 — Equipment arrangement of TLT panel.

in a distant office, he can then cause the trunk to be connected to the test access trunk (key- and jack-ended). The voltmeter test facilities as well as the transmission measuring circuit or milliwatt power circuit can then be associated with the test access trunk by key operations. If either the transmission measuring circuit or the milliwatt power circuit key is operated, the system will automatically switch the proper resistive pads into the test access trunk so that transmission testing can be performed within 0.1 dbm of the correct reference level.

Trunk retest — If the system has been instructed to outpulse on a particular outgoing trunk to a test terminal, the maintenance man can retest the trunk to the same test terminal by simply restoring and then reoperating one key. In this operation, the system releases the connection and then reestablishes another outpulsed connection to the same test terminal in the distant office via the same trunk. The system will continue this retest operation as long as the maintenance man continues to restore and reoperate the key.

The system can also be instructed to perform a similar repeat test automatically. That is, if a test call is established to a permanently busy test terminal or an incoming trunk test line in a distant office, the maintenance man can, by key operation, instruct the system to repeat the test on the same trunk for a maximum of thirty-two times. Each time the system establishes a new test call, it will monitor the signals or tone being returned from the test terminal and report when a failure is detected.

Substitute trunk test — Loop pulsing trunk circuits in ESS are designed with a bypass state. That is, under system control, the trunk circuit can be completely bypassed or removed from the trunk. This feature makes it possible to switch a substitute test trunk circuit in place of a loop-type trunk circuit suspected of being faulty. The network connections for this operation are shown in Fig. 10. This feature can be used to help isolate troubles in a trunk. If the maintenance man cannot successfully establish a call through either the regular trunk circuit or the substitute trunk circuit, he has reasonable assurance that the trouble is not in the regular trunk circuit.

Line testing — The maintenance man can identify a customer line in the same way that he identifies a trunk or service circuit. Once the system has received the identity of a line from the TLT panel, it reports the busy-idle status of the line and displays the line's class of service (e.g., coin, noncoin, PBX). If the line is idle, the system will connect the line to the TLT panel via the line test trunk. The line test trunk is key ended at the TLT so that the voltmeter test circuit can be associated with the line. Keys are also provided for ringing the line, exercising coin

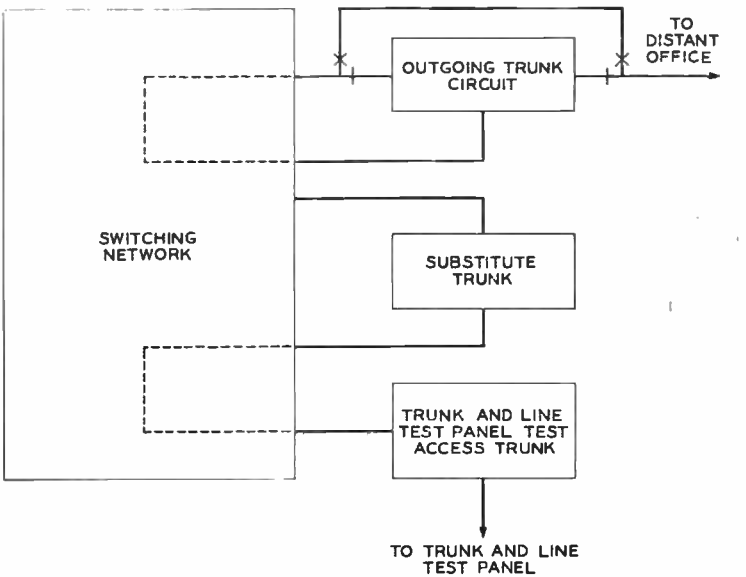


Fig. 10 — Network connections for substitute trunk.

control if it is a coin line, applying receiver off-hook (ROH) tone, and testing the ferrod associated with the line.

Monitoring — By key operation, the system can be instructed to establish a monitoring connection to any identified traffic-busy trunk, service circuit, or line.

Permanent signal holding trunk — When a permanent signal persists on an ESS line, the system sequentially connects the line to a recorded announcement, ROH tone, an operator trunk, and the permanent signal holding trunk appearing at the TLT panel. If the permanent signal condition ceases any time during this sequence, the line is restored to normal and the balance of the sequence is omitted. When the system connects a permanent signal line to the permanent signal holding trunk, it alerts the maintenance personnel by lighting a lamp associated with the holding trunk at the TLT panel. The system also indicates whether the line is serving a coin station, a PBX, etc. If the maintenance man takes no action on the permanent signal, the lamp will start flashing after a timed period and an audible alarm will be sounded.

The holding trunk is key ended at the TLT panel, and by key operation the maintenance man can challenge on the line, ring, or apply ROH tone. If these actions fail to clear the permanent signal, he may test the line with the voltmeter test circuit and instruct the system to disconnect the line from the holding trunk. When the system releases the network

connection to the line, it causes the directory number of the line to be printed on the TTY and continues to scan the line. There is no network connection to the line during this scanning operation. However, when the trouble causing the permanent signal is cleared, the system will automatically restore the line to service and notify the maintenance man via the TTY that the line is now free of trouble.

Only one permanent signal holding trunk is provided, because the ESS has the ability to establish a queue of permanent signal lines which have been subjected to all parts of the permanent signal sequence except for being connected to the holding trunk. Consequently, if permanent signals exist on more than one line, the line that has been in the permanent signal queue the longest will be connected to the holding trunk as soon as the maintenance man releases the holding trunk. The system will notify maintenance personnel if the number of lines waiting in the permanent signal queue increases beyond a certain number, and on request will print via the TTY a complete list of all line directory numbers which are in a permanent signal state.

Test progress and errors — The system is arranged to inform the maintenance man of the progress on any test call that he has instructed the system to perform, and in addition alert him to many kinds of errors. For example, if an irregular code is keyed to the system or if any of the control keys are operated incorrectly, the system will flash a lamp at the TLT panel.

Register listing feature — All information transmitted to the system from the TOUCH-TONE key set via the master test line is stored by the system in a register until the MTL is released. The contents of this register include the identity of a trunk and the outpulsed number, etc. The maintenance man may at any time interrogate the system as to the contents of the MTL register by operating a key. The system will then cause the contents of the register to be printed on the TTY.

The preceding description illustrates the types of features that are included in the TLT panel. All of the features provided in the TLT panel, including those described here, provide the operating company personnel with a flexible and versatile tool for testing and maintaining lines, service circuits, and trunks.

IV. AUTOMATIC MESSAGE ACCOUNTING IN NO. 1 ESS

4.1 *Introduction*

The automatic message accounting (AMA) facility in No. 1 ESS collects and records all pertinent data related to the charging of customer calls. This information is later transported to an accounting center where

it is used to determine the charges to be included on each customer's telephone bill. The entire operation can be divided into four parts:

(a) collection of the data on all calls for which charging information is required,

(b) assembly of data into coded format,

(c) data recording on a medium suitable for transportation to the accounting center, and

(d) data processing in the accounting center.

The first three of these function are performed by No. 1 ESS.

4.2 *General*

The major features of the No. 1 ESS AMA are:

(a) maximum use of the system data processing capability to minimize the amount of circuitry and hardware,

(b) data recording as a completely assembled entity for each call, and

(c) use of magnetic tape as the recording medium.

When the call processing program determines that an AMA record is required on a particular call, it stores the pertinent data in an AMA register located in call store (see Fig. 11). This information is then as-

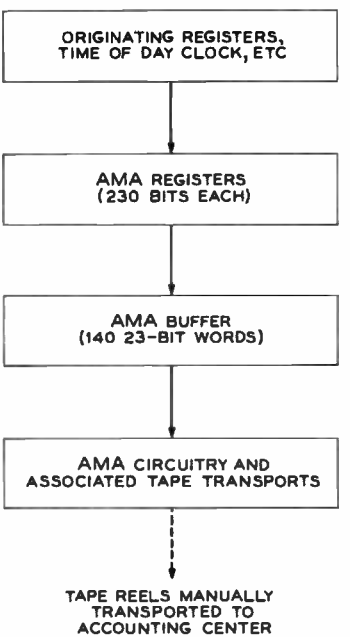


Fig. 11 — Simplified flow of AMA data.

sembled in binary-coded decimal format and stored in a temporary buffer storage area. When the buffer reaches its capacity, the recording procedure is initiated. An AMA program causes the data to be transferred one word at a time to the AMA circuit for recording on magnetic tape. Each tape is properly identified with labels for processing control.

Normally, each central office is provided with two AMA circuits and one block of buffer storage. One circuit serves as the active unit while the second serves as a standby. Each midnight, the system automatically switches the two units. Thus all AMA data during a 24-hour period are normally recorded on one magnetic tape. This complex of two AMA circuits and one block of buffer memory is capable of handling as many as 70,000 busy hour calls or about 100,000 calls on each reel of tape. In very large offices, four AMA circuits and two blocks of buffer memory will be provided. The two additional AMA circuits and the additional memory are required only when the traffic in a given office is greater than about 70,000 calls per busy hour or when the additional machines are desired so that tape changes will not have to be performed as frequently.

4.3 *Accumulation of Charge Information*

Before any chargeable call can be connected through the network, an AMA register must be associated with the call.[2,7] Once an AMA register has been associated with a call, all pertinent data available from the originating register are stored in the AMA register. The answer time and disconnect time are also stored when available as readings from the ESS time-of-day clock in hours, minutes, seconds, and tenths of seconds. When the call is completed, the information is organized into a binary-coded decimal format and stored in a buffer area.

Each AMA register has a 230-bit capacity (10 call store words), enough call store memory for almost any type of AMA entry. Facilities also exist for internal memory linkages with other registers to obtain extra storage area for types of calls requiring additional information. The extra memory is needed for calls such as those requiring operator assistance and the use of the traffic service position switchboard, or those calls requiring more than one billing entry for services such as dial conference and add-on.

Normally, only completed calls are recorded by No. 1 ESS. However, the call processing program is arranged to place a number of special marks in the originating register which also allow incomplete calls to be recorded. For example, if a calling line is arranged for service observing,

an indication is placed in the originating register that this call may be service observed. If the call is being observed, an entry is made in the AMA register and the call is entered as a detailed entry regardless of whether or not it was completed. Complaint observing entries are also provided for message rate lines. In this case, a special mark indicates that charge information for all AMA calls originating from a particular message rate line is entered on tape as a detailed entry.

4.4 *Assembly of Information into AMA Entry*

The information in the AMA register must be coded in a standard binary-coded decimal (BCD) form for use in the Bell System data processing centers. Because of the large number of entry types in the ESS, AMA formats have been specified to allow for recording all types of calls in a minimum of call store space. The information in an AMA register is arranged in a format appropriate to the particular type of call after all disconnect timing has been completed.

At midnight, before the active unit is switched into the standby state, all completed AMA entries are placed on the tape so that the total AMA record for the day is on one tape. When an AMA register has been held through two consecutive midnights, a trouble alert report is printed by the maintenance teletypewriter and the call can be manually checked for validity.

4.5 *AMA Buffer and Control*

The AMA buffer provides intermediate storage for the AMA data in the call store. Buffering is required to collect sufficient AMA data to fill an AMA block of 100 words, each word consisting of five 4-bit characters. When the AMA buffer is filled, the transfer sequence is started for recording this block on magnetic tape. (See Section 4.10 for a detailed description of the recording process.) Each 100-word block contains from seven to 20 call entries, depending upon the type of calls being recorded. Overlap of an entry from one tape record block to the next is permitted.

The buffer actually consists of 140 call store words, but it unloads only 100 words at a time. The extra 40 words permit the full contents of an AMA register to be placed in the buffer even though only part of the call fits into the 100-word block. As the program empties the buffer, that space becomes available for loading with new data.

When handling normal AMA traffic, the AMA block contains approximately 11 call entries. Recording on magnetic tape is done at a rate of one 100-word block per second. This provides a recording capability of

25,000 to 72,000 complete AMA entries per hour. Large offices with predominantly message-rate traffic may be equipped with two AMA buffers and four AMA circuits. These would increase the total AMA entry capability to between 80,000 and 144,000 entries per hour.

4.6 *Tape Format, Labels, and Codes*

The AMA data is written on the tape in a format which usually includes the following:

(a) tape header label,
(b) call entries,
(c) tape trailer label, and
(d) tape mark.

The tape header label is recorded once every day at midnight to indicate the start of a day's call records. Its contents include the originating area code or building number, the ESS office identification number or office number, the date, and the tape transport number.

Each call entry consists of a start-of-entry code, a type-of-entry code, and the data field. The type-of-entry code indicates the precise type of call and, consequently, the quantity and nature of the data to follow in the data field.

The tape trailer label is also recorded once every day at midnight to indicate the end of a day's call records. It includes the total number of call entries and the total number of 100-word blocks of call data recorded on the tape since the last header label.

The tape mark is a special character which indicates to the accounting center the end of the useful information on the tape. This label is recorded at the request of the maintenance man just prior to his removing the tape from the transport, or is recorded automatically when the system detects the physical end of the tape.

A transfer label is provided for use in special situations. This label is recorded on both tapes whenever it is necessary for the system to switch from one AMA circuit to the other because of trouble. However, switching is postponed as long as the traffic load will permit to allow immediate repair to the faulty circuitry. In many cases, this will allow all of one day's call records to be recorded on a single tape even though trouble occurred.

All data for a particular call are recorded on tape within a single entry. This single-entry arrangement employs the modified American Standards Code for Information Interchange (ASCII), as indicated in Table I. The billing data for each call are recorded as 4-bit BCD numbers in

TABLE I — AMA TAPE CODING IN NO. 1 ESS

Track and Code							
B7	B6	B5	B4	B3	B2	B1	
1	1	1	0	0	0	0	0
0	1	1	0	0	0	1	1
0	1	1	0	0	1	0	2
1	1	1	0	0	1	1	3
0	1	1	0	1	0	0	4
1	1	1	0	1	0	1	5
1	1	1	0	1	1	0	6
0	1	1	0	1	1	1	7
0	1	1	1	0	0	0	8
1	1	1	1	0	0	1	9
1	1	1	1	0	1	0	label identifier
0	1	1	1	0	1	1	noncheck dummy
1	1	1	1	1	0	0	start-of-entry code
0	1	1	1	1	0	1	check dummy
0	1	1	1	1	1	0	tape mark
1	1	1	1	1	1	1	write/read head check code
0	0	0	0	0	0	0	interblock gap
parity	fill		data				

groups of varying length. The previously mentioned labels (header, trailer, and transfer) always include the label identifier code, which is recorded once, twice, or three times to identify the type of label. The noncheck dummy code is used to fill out blank spaces in records where the blank is not the result of a trouble condition. This code is placed in an AMA entry before it leaves the AMA register. The check dummy code is also used to fill out blank places. However, the check dummy code is written by the AMA circuit in the event of trouble or in case a word is not received at the proper time from the AMA memory buffer.

4.7 *The Tape Recorder*

The digital tape recorders used in the AMA circuit have been specifically developed for No. 1 ESS use. Their design was based on three basic requirements:

(a) a recording capability of about 1000 7-bit characters per second on 0.5 inch magnetic tape,

(b) long-term reliability, and

(c) economy.

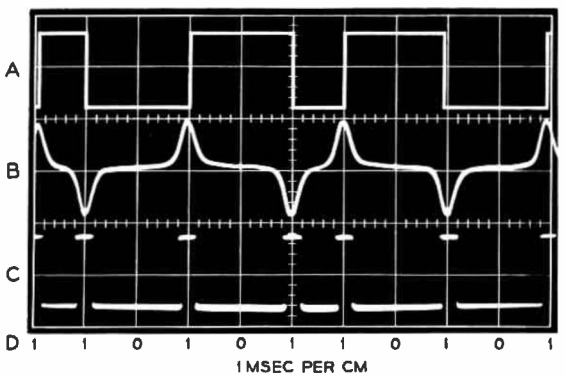
This relatively low data recording rate permits just one tape recorder to keep up with the busy-hour AMA traffic of the vast majority of central offices and allows the design of a simple and rugged recorder. Neverthe-

less, at least two AMA circuits are always provided, named AMA0 and AMA1, for reliability and continuity of recording. Each employs a tape recorder and the necessary control circuitry.

The tape recorder is a two-speed tape drive mechanism designed for start-stop operation. Tape speed for recording is 5.25 inches per second ± 5 per cent. A fast forward speed of 30 ips is also provided for winding unused tape on the take-up reel. There is no rewind feature. Both start and stop times are less than 100 msec. There are no belts, pulleys or gears. Tape is driven by the directly-coupled capstan motor shaft. Tape wow and flutter are less than 2 per cent.

Recording is done with a seven-track write head in the non-return-to-zero (NRZ) mode with ± 100 ma head current ± 10 per cent. A seven-track read head is spaced 2.25 inches from the write head and yields an output voltage of ± 1.3 mv ± 25 per cent (Ampex 832 magnetic tape); see Fig. 12. Provided also are a write-read tachometer head for checking tape speed and a seven-track erase head for erasing all previously recorded data before the tape moves under the write head.

Regular or heavy-duty 0.5-inch tape is used. Reels are of the 10.5-inch precision type with 2400 feet of tape per reel in "A" wind (oxide in). Tape skew at 5 ips is less than 140 μsec. Recording density is 200 bits per inch. Four tracks are used for data recording; one records an odd parity bit, and two are for fill bits that record "1's" when the circuit is recording.



A - WRITE HEAD INPUT - 200 MA PEAK-TO-PEAK
B - READ HEAD OUTPUT - 2 MV PEAK-TO-PEAK
C - READ CIRCUIT OUTPUT - 5V PEAK-TO-PEAK
D - RECORDED MESSAGE

Fig. 12 — AMA circuit waveforms.

4.8 *Functional Block Diagram*

A simplified AMA circuit block diagram is shown in Fig. 13. Communication with central control (CC) is accomplished over different channels for control, timing, data and alarm signals. Control signals received from the central pulse distributors (CPD) control the AMA circuit state, interrogate it as to certain alarm conditions, and signal the arrival of data for recording. The AMA circuit timing is based on the 0.5-msec and 5-msec central control clocks transmitted to it over private duplicated wire pairs. Data for recording are received over 21 wire pairs of one or the other peripheral address bus (PADB). The peripheral answer bus (PANB) is used to send to central control either trouble reports, tape readouts or data point signals for diagnostic checks. Scanner drivers report AMA circuit conditions, including:

(a) AMA circuit state,
(b) lack of tape tension,
(c) end of tape,
(d) incorrect tape speed, and
(e) power alarm.

The sequence control circuit tests for the proper command signal sequence from central control, provides the internal timing of the AMA circuit, and governs the writing of characters on tape.

The input register and translator store the data word while its parity is being checked by the parity checking circuit, divide the words into five characters, and store them along with a parity bit for each character until gated for recording on tape.

The read amplifiers send each read character to the check register, where its parity is checked. If a character fails the parity test, it is stored in the check register for gating to CC over the PANB.

The tape motion check circuit measures tape speed by timing the interval for tachometer pulses to travel from under the tachometer write head to the tachometer read head, with the two heads spaced 0.050 inch apart. If tape speed is not correct, a scan point is activated in the master scanner.

The alarm timeout circuit stops the motors and the recording sequence should central control start the motors and not order them stopped within 1 second. This provides self protection from major internal faults and from continuous neglect by the system.

Each AMA circuit employs 153 circuit packs with 645 transistors and 1693 diodes, and dissipates about 200 watts of +24-v power and 30 watts of −48-v power. Each tape recorder draws 0.8 ampere of 3-phase 208-v power and requires about 1 ampere of −48-v power for control.

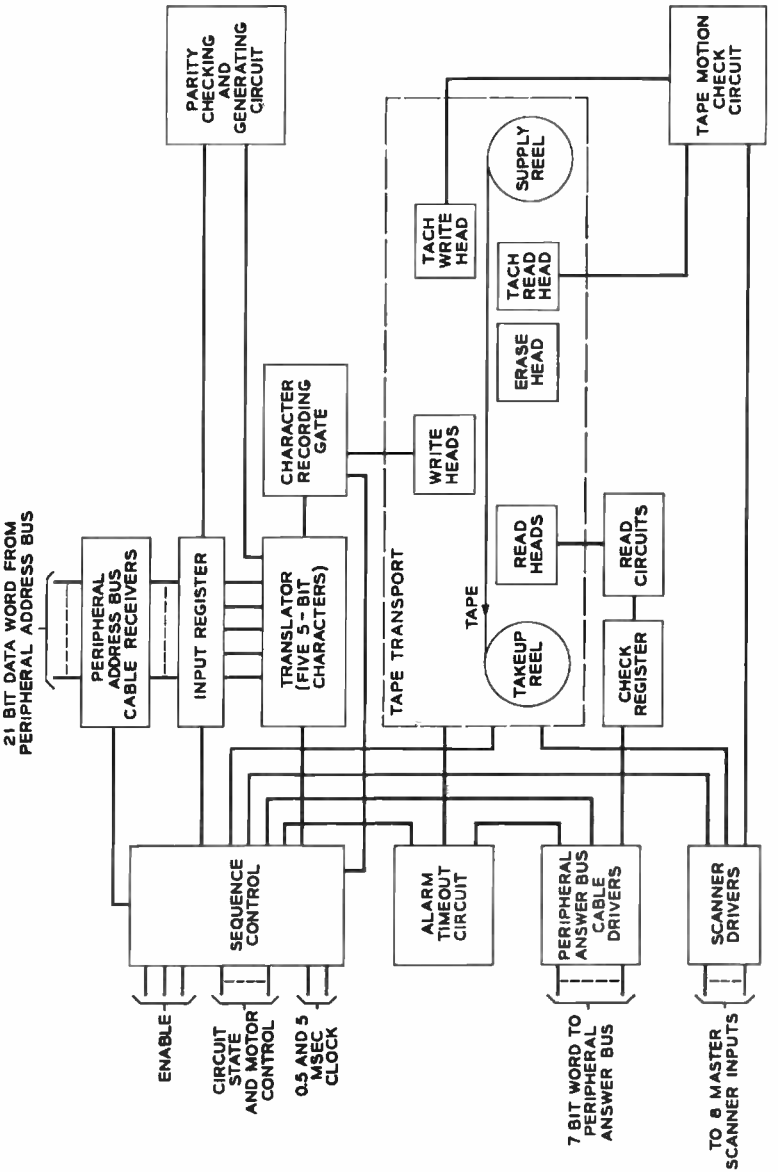


Fig. 13 — Simplified AMA circuit block diagram.

4.9 *AMA Circuit States*

By means of manual pushbutton control at the frame and system control via CPD signals, it is possible to order the AMA circuits into a number of states.

The three manual control states are:

normal — used for data recording and system diagnosis of troubles,

manual control — used when tapes are changed,

power off — used to remove all power from the circuit.

Mechanical interlocks on the control insure that at least one AMA system is in the normal state.

In the normal manual control state central control can place either AMA circuit into any one of five states:

active — ready for or actually recording data. Normally, during one 24-hour period only one AMA circuit is in active state and records all AMA data.

standby — ready for recording of data, but not expected normally to be put into the active state during this 24-hour period.

quarantine — as a result of system-diagnosed troubles the AMA circuit has been isolated from the system.

maintenance 1 — used to diagnose portions of the AMA circuitry for faults. Bypassing the write amplifiers, tape, and read amplifiers, incoming data are shunted to the check register.

maintenance 2 — used to report to central control various internal circuit conditions of the AMA sequence control.

Finally, if any fuse blows, all dc and ac power is removed from the circuit, a major alarm is initiated, and the circuit is in the power alarm state.

4.10 *Normal Operation*

By means of the proper CPD signals CC selects and places one of the two AMA circuits in the active state. It then sends a CPD signal which starts the tape transport motors and waits for all scan points to read zero, indicating no alarms and active state. This will occur in less than 150 msec, after the tape speed has stabilized to five inches per second. CC then sends the CPD a signal to begin the write sequence with the receipt of the next 5-msec clock pulse. The AMA sequence control circuit, in turn, times 1 msec, during which the enable signals and the data word for recording must be received. The first enable pulse (EN0) resets the input register. The next enable pulse (EN1) could be sent about 11 μsec later, but under normal conditions is omitted. The third enable pulse

(EN2) is sent another 11 μsec later, but still within the same millisecond interval, and informs the AMA circuit that during the next 2 μsec a data word will be sent, specifying the PADB bus. The data word consists of 20 data bits and 1 odd parity bit. After being gated to the input register it is checked for parity by the parity check and generating circuit. The 20 bits are arranged into five 4-bit characters in the input translator circuit, and the parity check and generating circuit calculates and stores an odd parity bit for each of the characters. At this stage there are five 5-bit characters ready to be recorded. The first character and two fill bits are recorded 1.0024 msec after the start of the write sequence, and others follow at 1-msec intervals. The fifth character is thus recorded 0.0024 msec after the "second" 5-msec clock pulse. Then the EN0 enable pulse is received again, resetting the input register. This EN0 also is used to gate to CC the parity of the 21-bit word just recorded (good or bad) and the fact that the check dummy character was not (or was) recorded. The check dummy character is defined in Section 4.6, and its use is explained in the next section. After this, another EN2 enable pulse is received and the cycle repeats. This 5-msec cycle is repeated for 100 21-bit words, yielding a 500-character AMA block.

At the end of the 500 characters, CPD signals the end of the write sequence. Another CPD signal orders the tape motors stopped. Within 100 msec the tape will slow to 5 per cent of normal speed. It should be noted that the AMA block is longer than 500 characters by the number of the check dummy characters recorded. Typically, the entire process of starting the tape transports, recording 500 characters and stopping takes less than one second. During this time about 3.5 inches of tape is used.

In addition to writing circuitry, the AMA contains tape reading circuitry, used only as a running check on the writing process. The write and read heads are separated by 2.25 inches. This means that the AMA circuits cannot write a character and simultaneously read it to see if the proper bits were recorded. Under normal conditions the read and check circuitry reads the characters off the tape, checks their parity and notes that they are not dummy characters, temporarily stores them in the check register, and then discards them.

4.11 *Operation under Trouble Conditions*

The AMA circuits have a number of features for detecting, diagnosing and reporting to the system any troubles encountered. In some instances the AMA circuit removes power from itself and places itself in an alarm state, activating the proper scanner drivers. In other cases, the AMA cir-

cuit merely reports faults to the system, such as wrong tape speed or parity failure on read, and relies on the system to put it into maintenance states for trouble diagnosis or to switch to the use of the standby AMA circuit. In both cases, trouble detection and switching to the alternate unit take place in less than one second, corresponding to the maximum loss of one block of AMA data. It is expected that most of the troubles will be detected in very much less time and, in fact, before any particular block of AMA data is to be recorded.

In case of tape breakage the AMA circuit automatically removes power from the transport motors, stops the write sequence, and places the circuit into the power alarm state. The same action is taken if the unit runs out of tape. Normally, the end of the tape is detected by photocells through a clear section of the tape at the end of the reel and the proper scan point is activated for the system to turn off power.

If the parity check and generating circuit finds that the 21-bit word received has even parity, the sequence control orders the AMA circuit to record five check dummy characters (0111101). The parity failure is reported to the system on the next EN0 pulse over the PANB. The system may elect to send that 21-bit word again on the next EN2 pulse for another recording.

The AMA recording activity is synchronized with the system 5-msec clock. If the circuit does not receive the 21-bit input word within the first millisecond after the recording cycle starts, it proceeds to record the five check dummy characters. This again is reported to the system as a 21-bit parity failure.

It was mentioned earlier that the read and check register circuitry is used to keep a running check on the write circuitry. If the check circuit finds an error, indicated by even parity of the 7-bit character or by detection of the dummy character, or if no characters at all are read from the tape, it locks that character in the check register and reports this fact to the system on the next EN0 pulse via the PANB. The system may then elect to gate the stored character out on EN1 over the PANB for inspection. The following EN2 pulse will clear the check register whether or not the character was gated to the system, and the check circuit will begin again checking characters read from the tape. The system determines whether the error rate as reported by the check circuit is great enough to put the particular AMA circuit into maintenance states for trouble diagnosis or to switch to the use of the other AMA circuit.

4.12 *Operation in Maintenance States*

For the purposes of trouble diagnosis in an AMA circuit, two maintenance states are provided. The tape transport motors do not run in

either maintenance state. In maintenance state 1 the translator outputs are fed directly to the check register. Thus the write amplifiers, tape, and read amplifiers are bypassed. The sequence control produces the clock pulses which gate the 7-bit characters from the translator at a 10-msec rate instead of the normal 1-msec rate. Data are received over the PADB at the rate of one 21-bit word every 50 msec instead of the normal 5 msec. Every character is gated back to CC. In maintenance state 2 the EN1 signal is used to gate to CC various internal conditions of the sequence control.

V. SUMMARY

The No. 1 ESS master control center provides:

1. a convenient means for both local and remote control of the system via teletypewriters,
2. a simplified arrangement for displaying and controlling the system operational status as well as for reporting system troubles to the maintenance personnel,
3. a facility to be used for manually testing lines, trunks, and service circuits, and
4. compact and inexpensive storage of subscriber charging information in a form readily usable in an electronic data processing system.

VI. ACKNOWLEDGMENTS

The authors wish to acknowledge their indebtedness to the many members of the electronic switching systems laboratory and center who have participated in the planning and design of the No. 1 ESS master control center. In particular, we would like to mention R. E. Hart, D. F. Illari and B. J. McMahon for the formulation of many of the system requirements; K. K. Kennedy, C. E. Krause and G. F. Swetnam for their contribution to the design of the AMA circuitry; J. E. Smathers and D. E. Smith for their contribution to the design of the TTY transmit-receive circuit; G. E. Jacoby for his contribution to the design of the line and trunk test circuit; and M. R. Gotthardt for his contribution to the design of the alarm, control and display circuit.

REFERENCES

1. Harr, J. A., Taylor, F. F., and Ulrich, W., Organization of No. 1 ESS Central Processor, B.S.T.J., this issue, p. 1845.
2. Harr, J. A., Hoover, Mrs. E. S., and Smith, R. B., Organization of the No. 1 ESS Stored Program, B.S.T.J., this issue, p. 1923.

3. Downing, R. W., Nowak, J. S., and Tuomenoksa, L. S., No. 1 ESS Maintenance Plan, B.S.T.J., this issue, p. 1961.
4. Connell, J. B., Hussey, L. W., and Ketchledge, R. W., No. 1 ESS Bus System, B.S.T.J., this issue, p. 2021.
5. Freimanis, L., Guercio, A. M., and May, H. F., No. 1 ESS Scanner, Signal Distributor, and Central Pulse Distributor, B.S.T.J., this issue, p. 2255.
6. Biddulph, R., Budlong, A. H., Casterline, R. C., Funk, D. L., and Goeller, L. F., Line, Trunk, Junctor, and Service Circuits for No. 1 ESS, B.T.S.J., this issue, p. 2321.
7. Carbaugh, D. H., Drew, G. G., Ghiron, H., and Hoover, Mrs. E. S., No. 1 ESS Call Processing, B.S.T.J., this issue, p. 2483.

Line, Trunk, Junctor, and Service Circuits for No. 1 ESS

By R. BIDDULPH, A. H. BUDLONG, R. C. CASTERLINE, D. L. FUNK and L. F. GOELLER, Jr.

(Manuscript received January 10, 1964)

In the No. 1 electronic switching system, individual circuits are needed on a per-line, per-trunk, and per-call basis to provide an interface between the outside world and the centralized call processing equipment. These circuits, including digit transmitters and receivers, are characterized by simplicity and compactness, and are program controlled. This article discusses their electrical and mechanical design along with transmission properties and maintenance procedures.

I. INTRODUCTION

Centralization of memory and control, long a dominant trend in telephone switching, has come close to the ultimate in the No. 1 electronic switching system (ESS). Nevertheless, individual circuits are still required on a per-call and even per-line basis to match the widely variable outside world to the standardized "inside world" of the central processor.[1] These individual circuits, the line, junctor, trunk and service circuits, provide the subject matter for this paper.

It is obvious that the central processor must work at very high speeds if it is to take over all memory and control functions. Even though actual signals from customers and other telephone offices come in relatively slowly, the central processor must operate rapidly to divide its time among the many signals flowing simultaneously over thousands of lines and trunks. What may not be so obvious, however, is the way in which this centralization affects the circuits in contact with lines and trunks during the processing of calls. In No. 1 ESS, these circuits have been reduced to very simple configurations; each circuit performs only a few functions under program control,[2] and different circuits are connected as needed via the switching network.[3] Both flexibility and economy result.

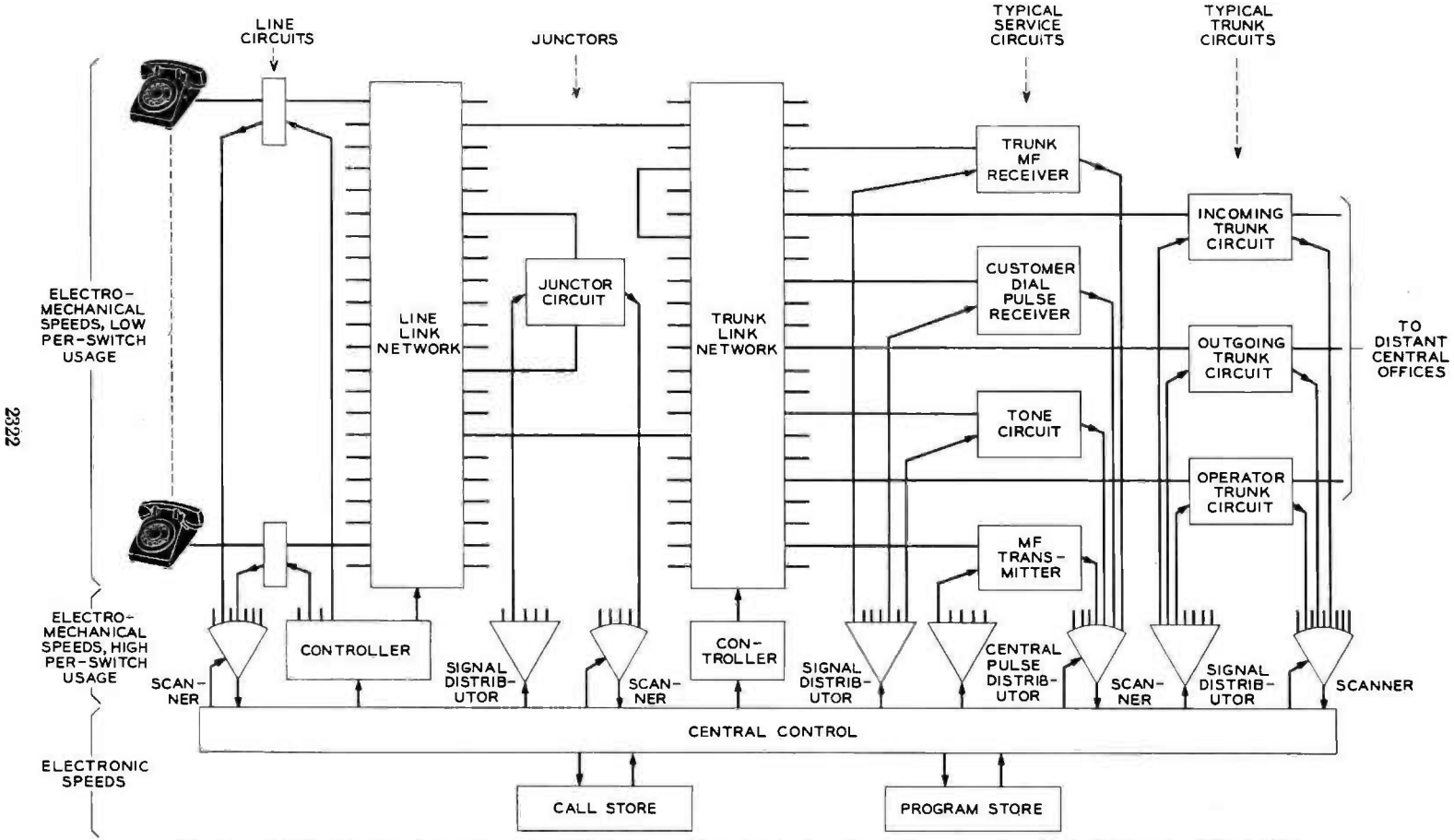


Fig. 1 — Block diagram showing relationship between line, trunk, junctor and service circuits and the rest of No. 1 ESS.

II. CIRCUITS DIRECTLY ASSOCIATED WITH CUSTOMER LINES AND TRUNKS

Fig. 1 shows the relationships between the various circuits associated directly with customer lines and trunks and the rest of No. 1 ESS. The status of customer lines and trunks is detected in the scanners by current-sensitive devices called "ferrod sensors" and transmitted to the central processor. The latter, consulting its memory and stored program, operates appropriate switching devices via signal distributors or central pulse distributors, depending on whether slow or fast action is required.[4] Interconnections are made via the networks, and the network controllers (rather than signal distributors) are used to operate switches in the line circuits.

It is important to note that ferrod sensors, although they behave in many ways like supervisory relays, perform no function other than current detection; they have no contacts and produce no circuit actions except indirectly via the central processor.

The scanners, signal distributors, central pulse distributors and network controllers act as input and output devices for the central processor. They require, however, additional circuitry to meet the variable conditions found on customer lines and trunks. Thus line, junctor, trunk and service circuits have been developed.

The line circuits shown in Fig. 2 are the simplest of these; they carry out the traditional functions of line and cutoff relays, supervising each customer line for originations and removing the sensing element to prevent its shunting the talking path. Carbon protector blocks, also shown as part of the line circuits, limit lightning surge voltages. In a way, there is no such entity as a line circuit in ESS, since the ferrod sensor is part of the line scanner, the cutoff relay is part of the network, and the carbon blocks are part of the protector frame. Nevertheless, it is convenient to refer to line circuits for purposes of exposition.

Next in complexity comes the junctor circuit, shown in Fig. 3. This circuit is used only during conversations between customers served by the same No. 1 ESS. Calling and called customers are connected to a junctor circuit via paths through line link network only (the trunk link network is not required, as can be seen from Fig. 1). Each junctor circuit contains two relays, labeled A and B. These are magnetic latching relays, pulsed operated or released by the signal distributor on command from the central processor.

Trunk circuits contain only a few more components than junctor circuits, but because more is required of them, their design is considerably more complex. They will be discussed in detail in the several sections

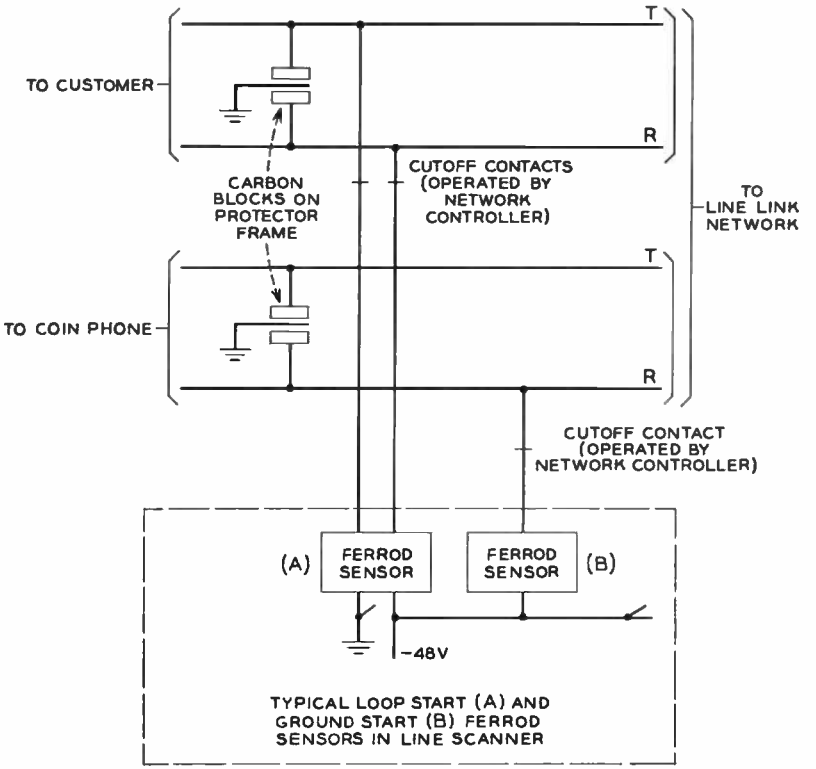


Fig. 2 — Typical loop start and ground start line circuits for sensing call originations and subsequently removing the sensor shunt.

which follow. At this point, however, it is necessary to emphasize the difference between a trunk and a trunk circuit. A "trunk" is a communication channel between two switching systems. It starts at the outgoing terminals of the switching network in the originating office and ends at the incoming terminals of the switching network of the terminating office.* As shown in Fig. 4, a trunk includes the transmission facility terminated in two "trunk circuits," one at each end. Traditionally, trunk circuits convert supervisory information (telephone on or off hook) from the distant central office into a form suitable for local use, and, conversely, convert local supervision to a form which can be transmitted in the opposite direction. Often, in present systems, current for the

* For transmission purposes, a trunk is measured between the outgoing network terminals of the originating office and the outgoing network terminals of the terminating office.

transmitter in local subsets is supplied from trunk circuits, digit pulsing features are included, and a variety of other functions are performed.

In No. 1 ESS, trunk circuits retain very few of these functions. Indeed, as can be seen in Table I, coin control, ringing, tone application and the like are *not* provided in trunk circuits but come from "service circuits" to which lines and trunks can be connected as required via the line and trunk link networks. This use of the networks is possible only because of the speed with which the networks and the central processor can work together to find, remember and set up paths.

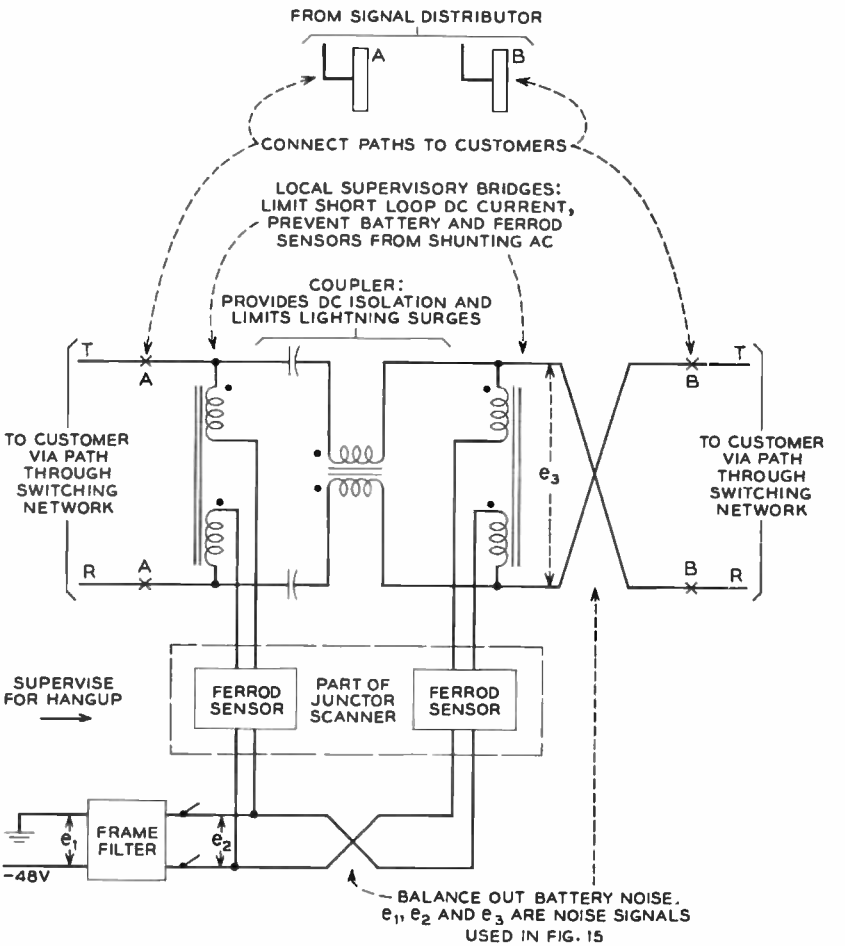


Fig. 3 — Junctor circuit used during conversations between customers served by the same ESS.

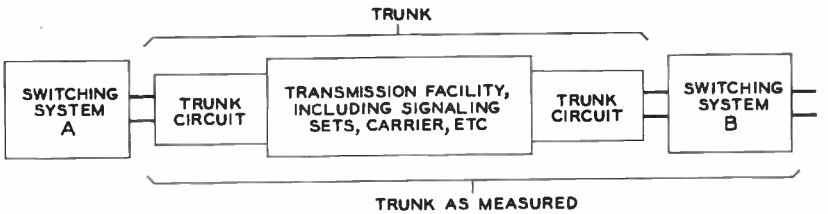


Fig. 4 — Relationship of trunk circuits to a trunk.

The advantages of this approach are many: no distinction between coin and noncoin trunks need be made, since coin control is not a function of the trunk circuit. Different types of ringing can be applied to different lines on a class basis. This permits sensitive trip relays with greater range to be used wherever possible, while less sensitive trip relays are retained for the remaining lines. New types of ringing can be added as desired, since, unlike earlier systems, no changes need be made on a per-trunk basis. Digit transmitters and receivers for signaling distant offices can be arranged in single groups by pulsing type, giving the usual

TABLE I—GENERAL TRUNK CIRCUIT FUNCTIONS AND NO. 1 ESS CIRCUITS WHICH PERFORM THESE FUNCTIONS

Function	Performed in No. 1 ESS by
Supervision	Scanners
Battery feed, dc isolation Transmission to local lines Transmission, trunk-to-trunk Path continuity check through line and trunk link networks Lightning surge protection for line and trunk link networks Make and break current through line and trunk link networks	Talking circuits, including trunk, junctor, and conference circuits
Signaling Ringing Returning tones Coin control Foreign potential detection Certain other tests	Service circuits including digit transmitters and receivers and ringing, tone, coin control, and test circuits
Timing Sequencing Memory (including digit storage) Charging	System control, including central control, call processor, call store, and program store

TABLE II—LINE, TRUNK AND SERVICE CIRCUITS USED IN PROCESSING TELEPHONE CALLS IN NO. 1 ESS

Function	Intraoffice Call	Outgoing Call	Incoming Call
Detect origination	scanner via line ckt.	scanner via line ckt.	scanner via incoming trunk ckt.
Foreign potential test, party test, return dial tone	customer dial pulse receiver	customer dial pulse receiver	—
Obtain digits of called party	customer dial pulse receiver	customer dial pulse receiver	dial pulse, revertive, or multifrequency receiver as required
Outpulse digits of called party	—	dial pulse, revertive, panel call indicator or multifrequency transmitter as required	—
Return busy, overflow, or audible ringing tone to calling customer	tone circuit	circuits in distant office	tone circuit
Send ringing signal to called customer, detect answer and trip ringing	ringing circuit	circuits in distant office	ringing circuit
Provide talking current, transmission circuit	junctor circuit	outgoing trunk circuit	incoming trunk circuit
Supervise for hangup	scanner via junctor circuit	scanner via outgoing trunk ckt.	scanner via incoming trunk ckt.

advantages of one large group over several smaller groups. Further, the type of pulsing on any trunk can be altered by a simple program change, because any transmitter or receiver can be used with any trunk, depending only on a translation option.

Table II shows how various line, junctor, trunk and service circuits are used to process typical telephone calls in No. 1 ESS. Several circuits and several network connections may be in simultaneous use on a single call.[3]

III. THE DESIGN OF TRUNK AND SERVICE CIRCUITS

3.1 *Switching Design*

Each trunk and service circuit in No. 1 ESS carries out some of the functions listed in Table I. However, within any one circuit, association of one function with a particular switching device is uneconomical. Instead, a given function is related to one particular state of a group of switches. The Karnaugh map[5] is a design tool well suited for such situations; it also aids in circuit explanation.

An example of the technique is afforded by a somewhat simplified version of the customer dial pulse receiver (CDPR) circuit as shown in Fig. 5. The CDPR is a service circuit used to correct distortion on pulses generated by customer dials and to repeat the improved dial pulses to the scanner. It performs the additional functions of testing for foreign potentials crossed to outside plant conductors, party identification on two-party lines, return and removal of dial tone, check of continuity through the switching network, and cut-through or the making and breaking of current flowing in each connection.

Matching circuit functions to circuit states must be done in such a way that (a) a minimum number of states is required and (b) the transitions from state to state are made as simply as possible. The solution for the CDPR is shown in Fig. 5 along with the circuit. Three magnetic latching relays, A, B and C, are operated by the signal distributor to provide eight states.

For minimum delay as well as minimum relay wear, the operation or release of just one relay takes the circuit from one useful state to the next. Two arrows, one solid and one dotted, show the two principal state sequences. For individual lines, the A relay is operated to put the circuit in the power cross test state. If no crosses to a foreign potential are detected, the B relay is operated to connect dial tone and the pulse-correcting circuitry. A is released to remove dial tone while leaving the pulse-correcting circuitry connected, and B is released to make the circuit idle when all digits have been received from the customer. Only four signal distributor operations suffice.

It must be remembered, however, that the scanner extracts information from the circuit at a high enough sampling rate to detect all dial pulses and that the central processor counts and stores digits, times for interdigital periods, translates route information, etc. The simplicity of the CDPR is possible only because of the versatility of the central processor.

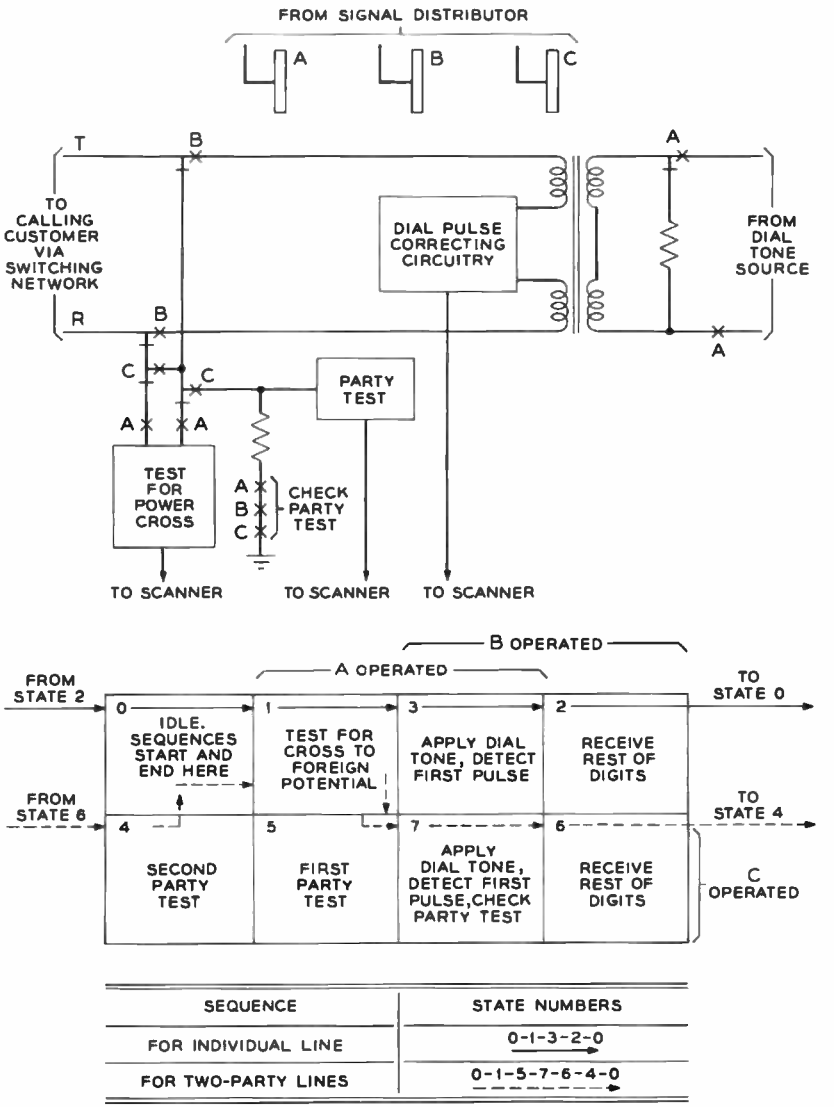


Fig. 5 — Simplified circuit and Karnaugh map for customer dial pulse receiver, illustrating association of functions with states rather than individual relays.

The sequence for two-party lines (dotted arrow) starts with the power cross test state as before, then, on operation of the C relay, enters the first of two party-test states. Dial tone is returned after the party test is completed, and while dial tone is being transmitted the party-test circuit is checked for proper operation. Dial tone is removed after the customer starts dialing. Then, after all digits are received, the second party test is performed. Finally, the circuit is returned to its idle state. Only six signal distributor operations are required.

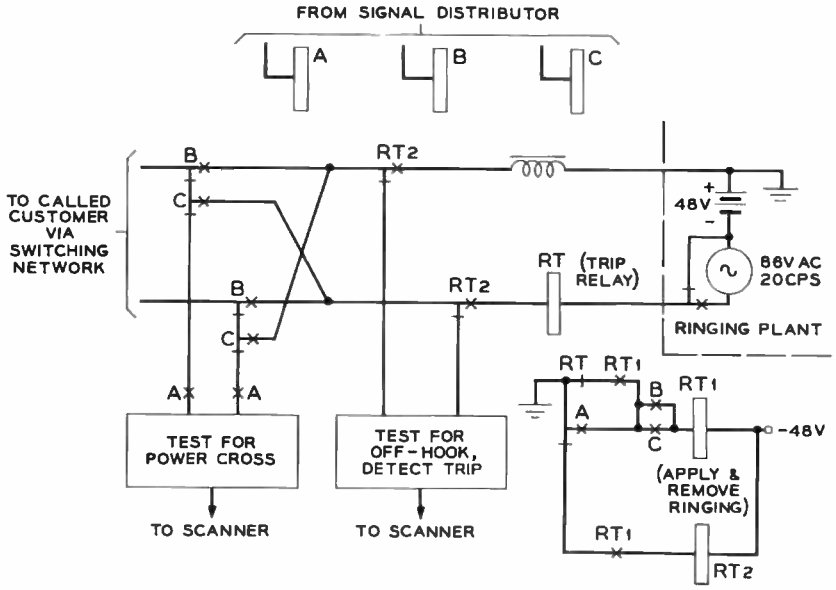
As a convenient method of keeping track of circuit states, each relay operated by the signal distributor is given a "weighting" number: A = 1, B = 2, and C = 4.* In any given state the weighting numbers of each operated relay are added together to produce the state number. Thus the two sequences in the customer dial pulse receiver described above would be 0-1-3-2-0 and 0-1-5-7-6-4-0 respectively. It must be emphasized that only one relay can be operated or released by a signal distributor at any one time. Thus sequences such as 0-3 are impossible. This constraint can be seen easily if one "walks through the map" as in Fig. 5 from one adjacent square to another, keeping in mind that opposite edges of the map are considered adjacent (as in the transition 2-0, where the B relay is released).

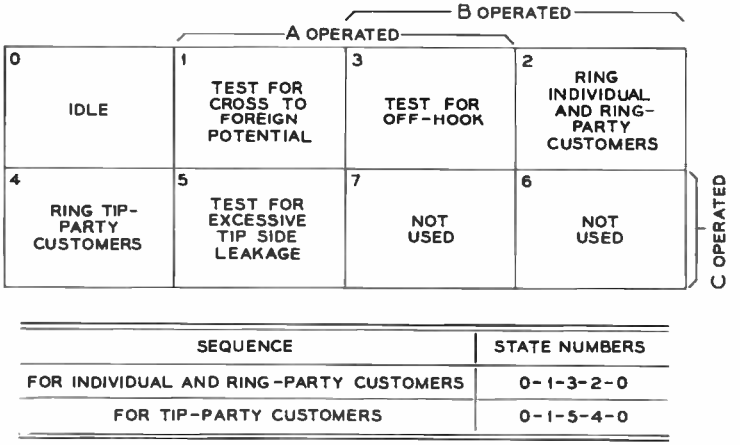
Figs. 6, 7 and 8 show three additional service circuits with their associated maps and typical sequences. Ringing, coin control potentials, and tones can thus be applied to any customer line in the office independent of the circuits used for conversation.

3.2 *Transmission Design*

Fig. 3 shows the transmission configuration used for intraoffice calls and also indicates the functions of some of the components. When a call is destined for a distant office, the somewhat more complex transmission circuit of Fig. 9 is employed. This latter circuit uses different coil resistances to control trunk supervisory currents, provides improved longitudinal balance, and also permits an impedance transformation when necessary. A nonconventional feature is use of an inductor-resistor network which substantially reduces the variation of impedance of customer lines with frequency as seen by the interoffice transmission facility. Although this network should be considered part of the customer loop to which the trunk is attached in any given conversation, the improved return loss it provides is useful only in transmission paths containing gain. Thus it is placed in trunk circuits for economic reasons.

* When more than three relays are required, octal notation is used: D = 10, E = 20, F = 40, G = 100, H = 200, etc.

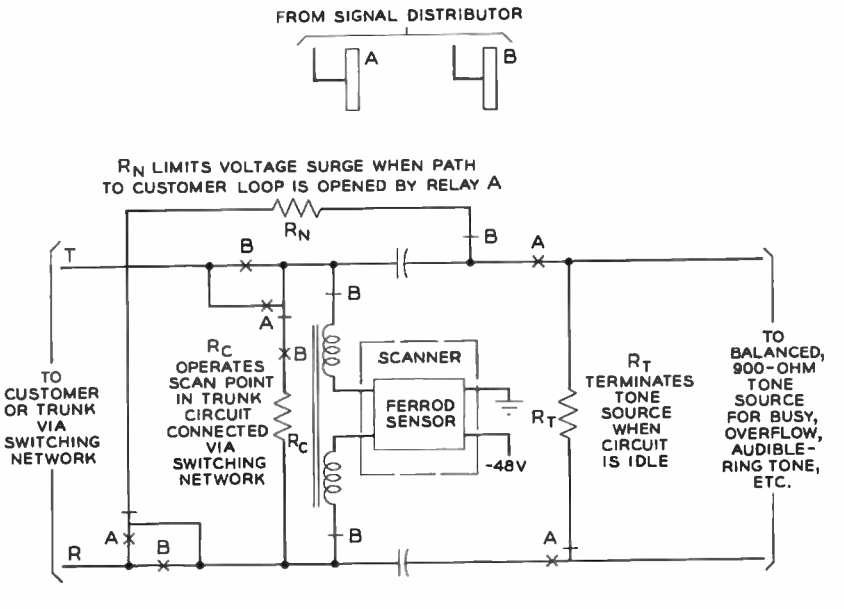




SEQUENCE	STATE NUMBERS
FOR INDIVIDUAL AND RING-PARTY CUSTOMERS	0-1-3-2-0
FOR TIP-PARTY CUSTOMERS	0-1-5-4-0

Fig. 6 — Simplified circuit for applying ringing to certain classes of lines.

Fig. 10 shows the effects of this simple network on mean loop impedance, and Fig. 11 illustrates the increased mean return loss resulting from its use. Fig. 12, based on a comprehensive survey, shows how this improved echo and singing return loss is distributed over present Bell System loops.[6] This increase in customer loop return loss can be utilized in various ways; in particular, some increase in gain can be permitted on



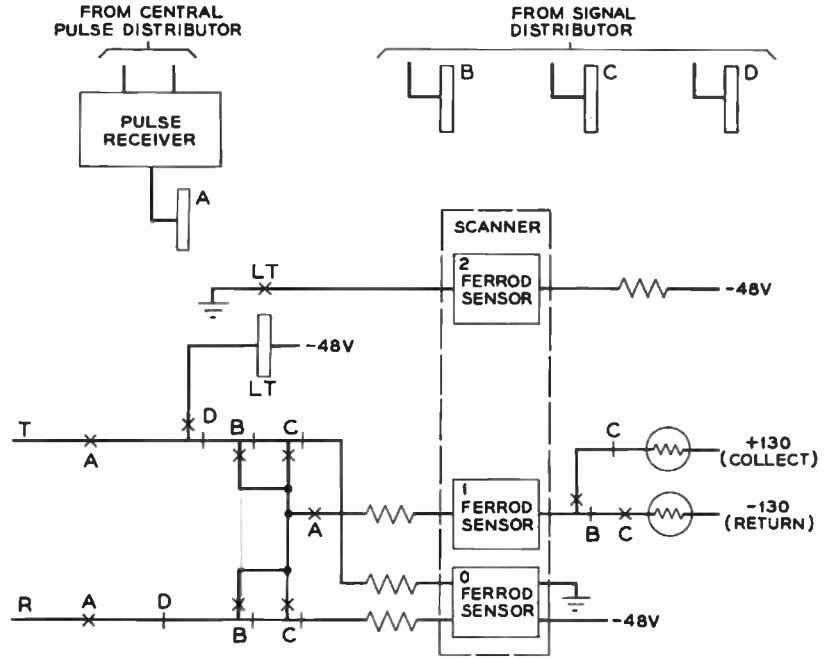
	A OPERATED	B OPERATED
0 IDLE	1 APPLY TONE TO ESS CUSTOMER. SUPERVISE AT ASSOCIATED SCAN POINT	3 APPLY TONE TO TRUNK. LOCAL SUPERVISORY BRIDGE DISCONNECTED
		2 USE R_C TO CHECK CONTINUITY THROUGH NETWORK TO TRUNK CIRCUIT

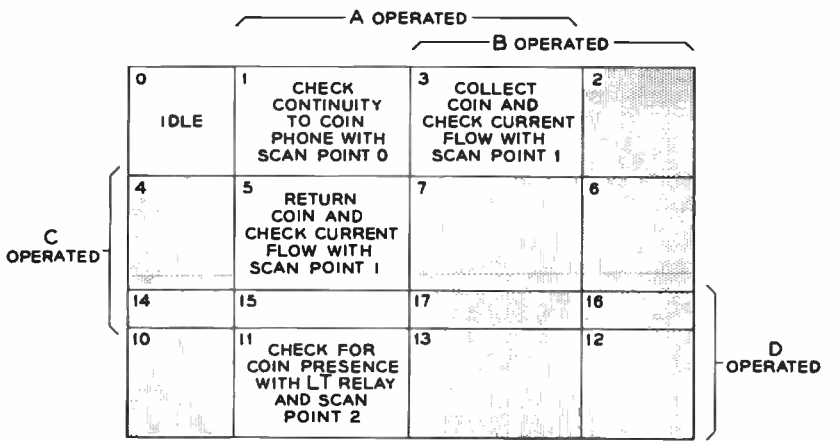
SEQUENCE	STATE NUMBERS
FOR LOCAL CUSTOMERS	0-1-0
FOR TRUNKS	0-2-3-2-0

Fig. 7 — Tone circuit.

toll connecting trunks. Although not shown on Fig. 9, idle circuit terminations are provided to assure repeater stability.

In a tandem connection, an incoming trunk must be joined to an outgoing trunk. Under such circumstances, customer loop compensation is omitted. Further, the local supervisory bridges in both the incoming and outgoing trunk circuits are switched out (as discussed in Section 3.3) to produce the equivalent of exactly one transmission circuit. The insertion loss of this tandem circuit varies less than 0.4 db between 200





SEQUENCE	STATE NUMBERS
COIN COLLECT	0-1-3-1-0
COIN RETURN	0-1-5-1-0
COIN TEST	0-1-11-1-0

Fig. 8 — Simplified coin control circuit; because of very short holding time, central pulse distributor is used to speed up operation.

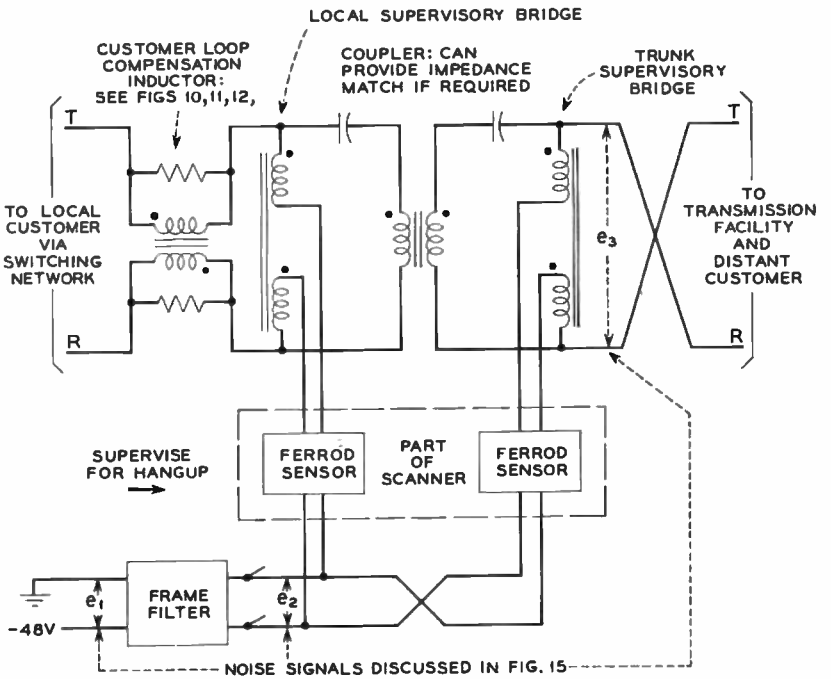


Fig. 9 — Transmission elements of trunk circuits: as shown, configuration corresponds to an incoming trunk circuit.

and 5000 cps, with a 1000-cps flat loss of 0.3 db. The structural echo return loss (not including switching network conductors)* is 36 db measured between 900 ohm + 2.14 mf terminations.

Fig. 13 illustrates the longitudinal balance characteristics of trunk and junctor circuits. Balance of each conductor relative to ground is necessary to reduce longitudinally induced noise from power lines and earth potentials, and to reduce battery noise and crosstalk coupled by common ground impedances. Individual components are designed to provide a longitudinal balance for the entire trunk circuit of at least 55 db when measured as shown. The measuring circuit simulates representative field situations. Both inductive and resistive components of the supervisory bridge inductors relative to their midpoints must be carefully controlled, and a balanced configuration is required in both transformer and blocking capacitors.

* Structural return loss of a trunk circuit is defined as the return loss of a trunk circuit terminated in a reference network measured against an identical reference network.

Because of the compact equipment arrangements in trunk and junctor circuits (to be discussed in Section V), great care is exercised to reduce crosstalk coupling. Special efforts have been made to reduce impulse noise in one circuit produced by relay operations in the second circuit on the same equipment unit. Audio-frequency crosstalk has also received attention; Fig. 14 shows the attenuation of voice-frequency crosstalk between a pair of such circuits.

Battery noise and common battery impedance present another disturbance to transmission. Fig. 15 shows how the frame battery filter reduces such noise; also shown is the effect of the well-known transposition which causes any remaining noise introduced into the line side to cancel that introduced into the trunk side.

In addition to the transmission properties of trunk and junctor cir-

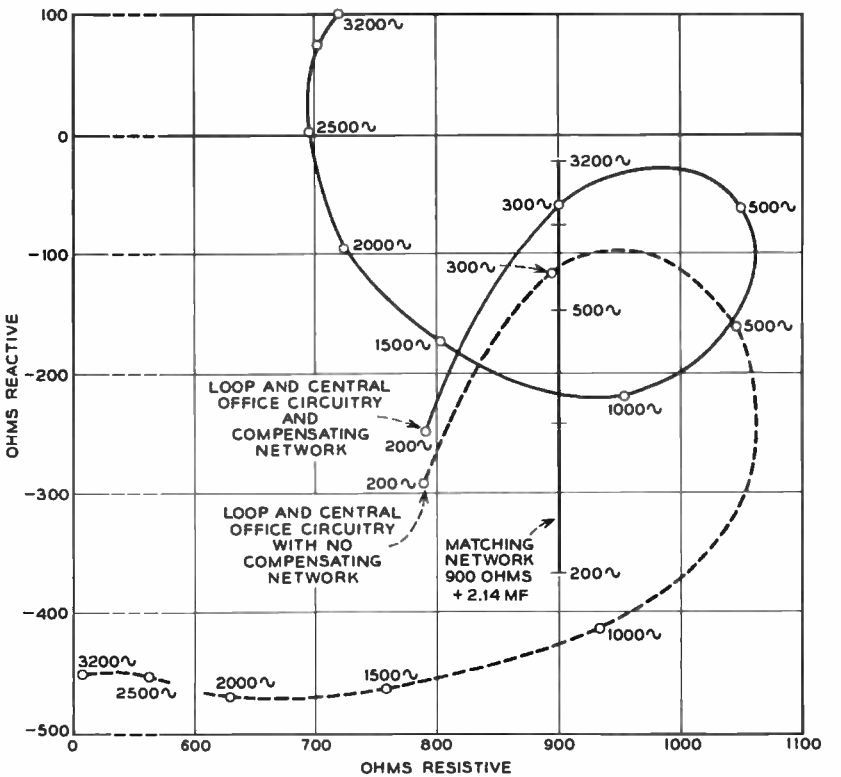


Fig. 10 — Effect of compensating network on mean input impedance of Bell System customer loops.

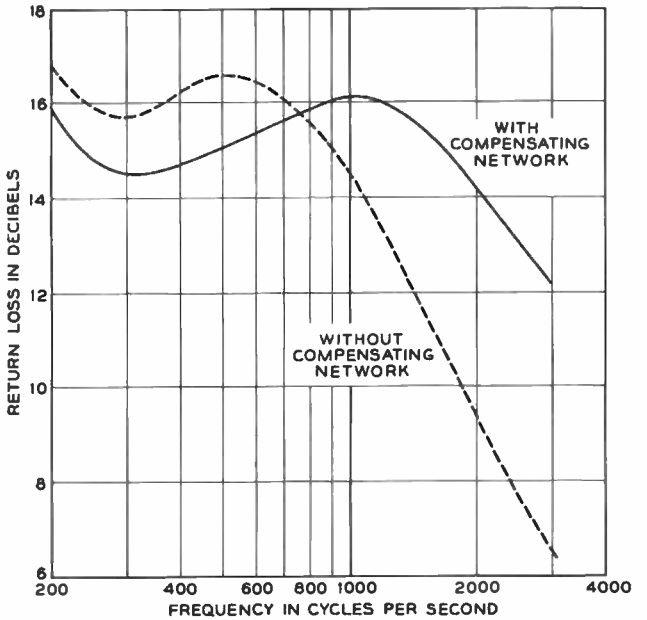


Fig. 11 — Effect of compensating network on mean customer loop return loss matched against 900 ohms + 2.14 mfd.

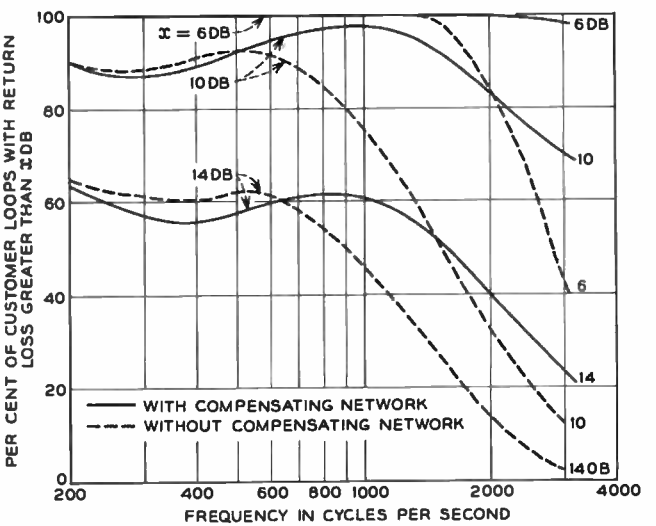


Fig. 12 — Effect of compensating network on return loss distribution of customer loops.

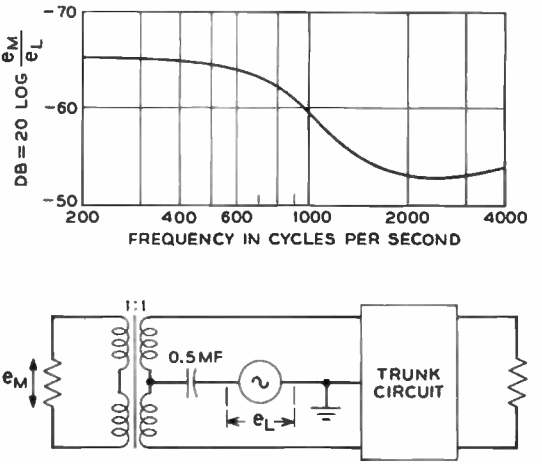


Fig. 13 — Longitudinal balance of trunk transmission circuit. Balance in db = 20 log $|e_{metallic}/e_{longitudinal}|$ when measured as shown.

cuits, a number of other transmission features of importance are included in No. 1 ESS. The tone generators which supply signals to the tone circuit in Fig. 7 are balanced to ground and present a very high return loss when measured against 900 ohms + 2.14 mf. The level of these tones is closely controlled. Similar balance and return loss properties will be found in other service circuits, including the customer dial pulse receiver and interoffice transmitters and receivers.

A transistorized conference circuit is available to permit up to four persons to hold a joint conversation. Each input to this circuit has a separate appearance on the switching network so that full access to all lines and trunks is available. Modified versions of the conference circuit are used with certain operator-controlled calls and coin-zone dialing applications to permit the operator to split and hold the parties and talk to either or both without impairing transmission.

3.3 *Switching and Transmission in Trunk Circuits*

When the switching principles described in connection with the service circuits are combined with the basic transmission package, a small but versatile group of trunk circuits results. Three are chosen for discussion in Figs. 16, 17 and 18. Just as service circuits are made up of basic building blocks such as pulse correction circuits, detectors, etc, which are connected singly and in groups as needed, so the transmission package

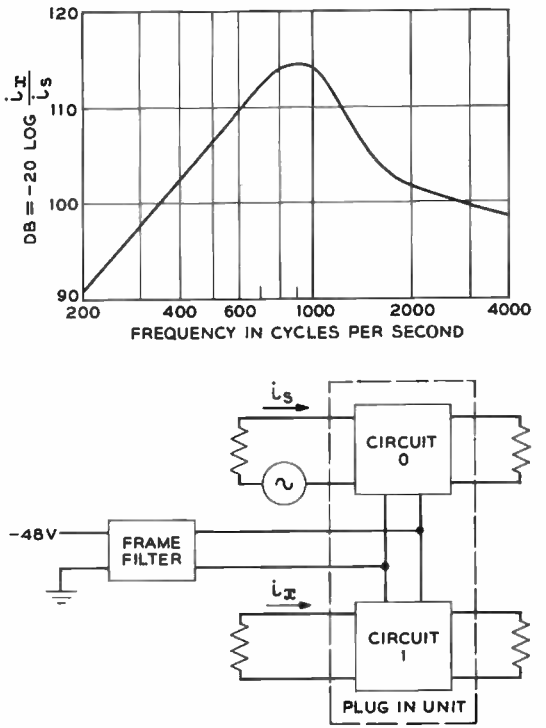


Fig. 14 — Attenuation of crosstalk signal i_x in ckt. 1 over input signal i_s in ckt. 0 when coupling is via common frame filter and power wiring as well as adjacency of components on mounting unit.

can be thought of as supervisory bridges, coupler, loop compensation network, idle circuit termination, and the like. In addition to arranging these components in various ways for local and tandem calls, relays in trunk circuits can bypass them all and connect a pair of conductors with no series or shunt components to the switching network. This permits direct connection to digit transmitters and receivers as required; dial or revertive pulses as well as multifrequency and panel call indicator signals can bypass the transmission elements in the trunk circuits.

Bypass and tandem requirements dominate in establishing the switching design of most trunk circuits. The general idea is to have only two supervisory bridges (one for each trunk) and one coupler in any tandem connection. To achieve this, three rules are applied:

(1) Operator trunk circuits always retain the coupler. In all trunk

circuits the coupler contains a transformer, which is usually a 1:1 device in the 900-ohm ESS office. However, switchboards usually use 600-ohm circuits and, if a switchboard is located in the same building as the ESS, one trunk circuit is shared by both. Because of this frequent need for impedance transformation in operator trunk circuits, the coupler always remains.

(2) Incoming (nonoperator) trunks must frequently be connected to operators for intercept service, etc. Since operator trunk circuits always contain couplers, incoming trunk circuits never retain couplers on a tandem connection.

(3) Outgoing (nonoperator) trunks may be seized by either operator or nonoperator incoming trunks. Since operator trunk circuits always contain the coupler and incoming circuits never contain it, two separate tandem states are provided in outgoing trunk circuits, one with and one without the coupler.

Comparison of the Karnaugh maps of the incoming and outgoing trunk circuits shows that the same number of states is used in each circuit. This points up another reason for rules (2) and (3) above. The incoming trunk circuit must transmit a reverse battery signal to the calling office when the called customer answers, so that the calling office can start charging. This reversal can be made only by a signal distributor operation. Thus the incoming trunk circuit needs two talking states, one for charge and one for free, in both the tandem and local conditions (four states in all). On the other hand, the outgoing trunk circuit detects the battery reversal from the distant office by means of a diode-polarized ferrod sensor in the scanner which transmits the information to the

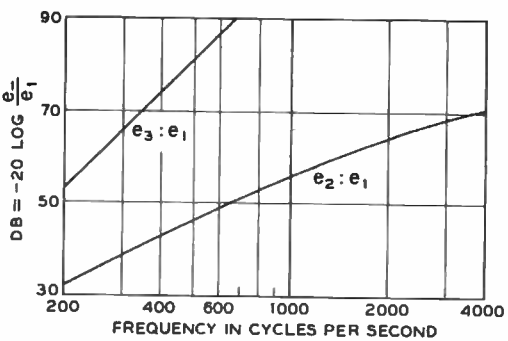
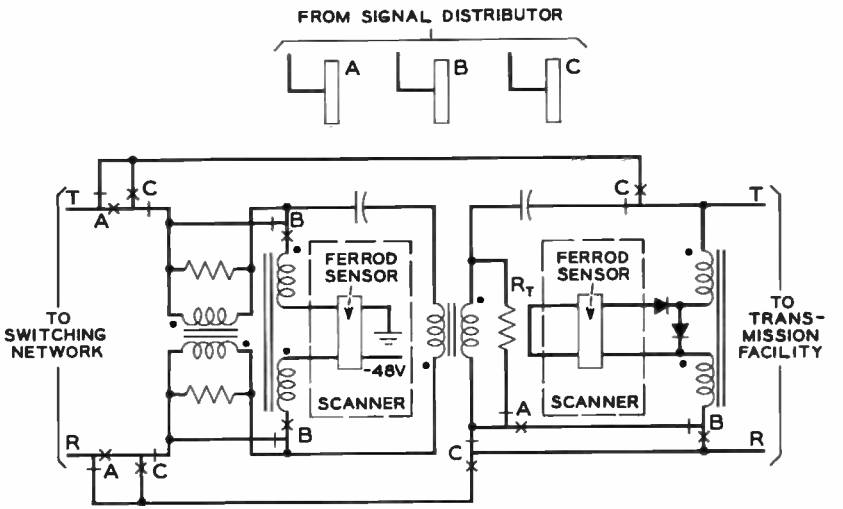
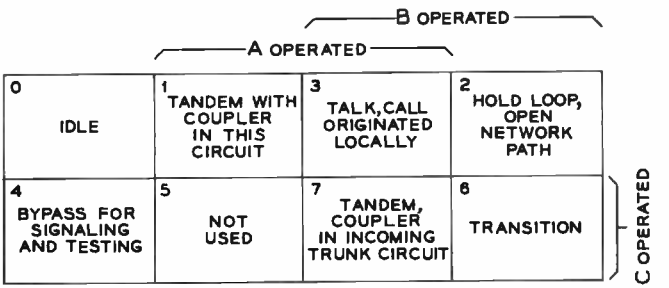


Fig. 15 — Attenuation of noise by frame filter (e_2:e_1) and combined attenuation due to filter and noise canceling transposition (e_3:e_1). See Figs. 3 and 9 for circuits and definitions of e_1, e_2, and e_3.





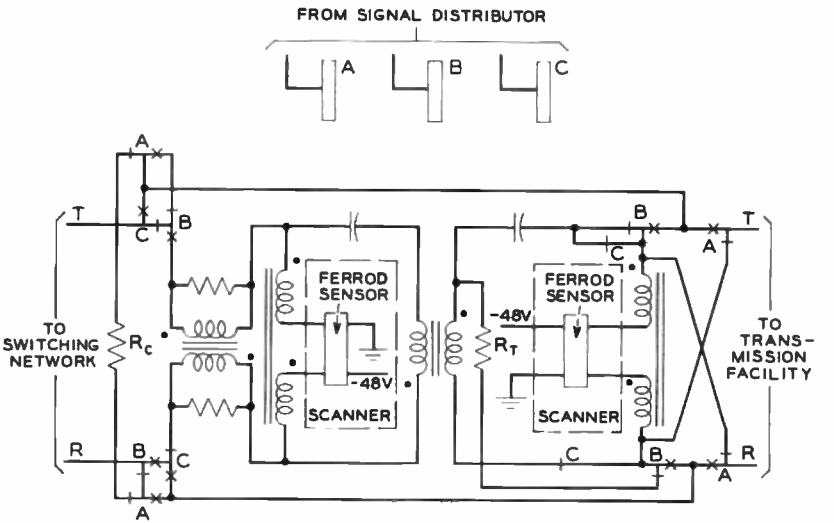
SEQUENCE	STATE NUMBERS
LOCAL ORIGINATED CALL	0-4-6-2-3-2-0
TANDEM-(COUPLER IN THIS CIRCUIT)	0-4-6-2-3-1-0
TANDEM-(COUPLER IN INCOMING TRUNK CIRCUIT)	0-4-6-2-3-7-3-2-0

Fig. 16 — Outgoing trunk circuit.

central processor. Thus no differentiation between charge and free talking states need be made by signal distributor operations. This factor makes it easier to provide the two tandem states in the outgoing trunk circuit than in the incoming trunk circuit. With this background discussion, the operation of the circuits in Figs. 16, 17, and 18 should be self-explanatory.

IV. DIGIT TRANSMITTERS AND RECEIVERS

Digit transmitters and receivers, unlike the registers and senders of crossbar and panel offices, are relatively simple circuits. In No. 1 ESS, all digit registration and all control of office operation are carried out by the central processor. Thus, the only major function left to be performed by transmitters and receivers is translation from the "language" of trunks and lines to the "language" of the central processor.



B OPERATED

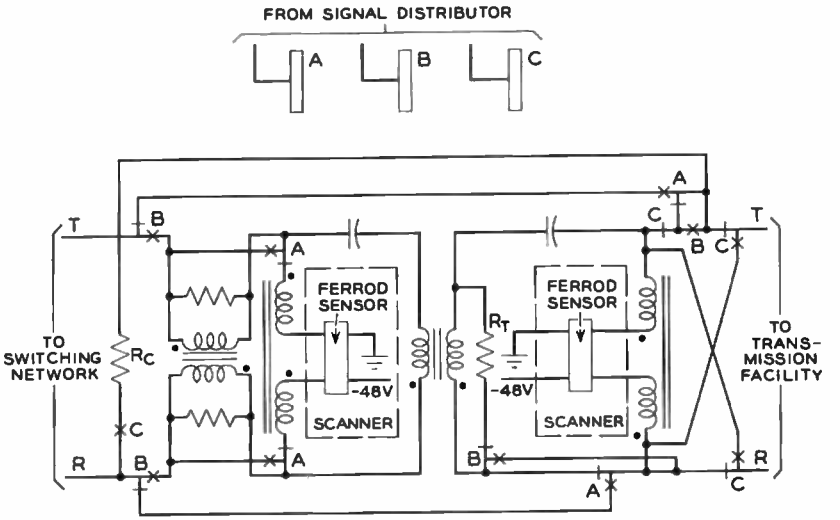
A OPERATED

0 MONITOR FOR ORIGINATION, HOLD	1 BYPASS FOR SIGNALING	3 TALK LOCAL CHARGE	2 TALK LOCAL FREE
4 NETWORK CONTINUITY CHECK USING R_C	5 NOT USED	7 TANDEM CHARGE	6 TANDEM FREE

C OPERATED

SEQUENCE	STATE NUMBERS
LOCAL TERMINATED CALL	0-1-0-2-0-2-3-2-0
TANDEM CALL	0-1-0-2-6-7-6-2-0

Fig. 17 — Incoming trunk circuit.



B OPERATED

A OPERATED

0 IDLE, MONITOR FOR ORIGINATION	1 BYPASS FOR SIGNALING	3 TANDEM (OFF-HOOK)	2 TALK LOCAL (OFF-HOOK)
4 SEIZE AND HOLD	5 NETWORK CONTINUITY CHECK USING R_C	7 TANDEM (ON-HOOK)	8 TALK LOCAL (ON-HOOK)

C OPERATED

SEQUENCE	STATE NUMBERS
LOCAL ORIGINATED CALL	0-4-6-2-0
LOCAL TERMINATED CALL	0-1-0-2-3-2-0-2-6-2-0
INCOMING TANDEM CALL	0-1-0-4-5-7-3-2-0
OUTGOING TANDEM CALL	0-4-5-7-3-1-0

Fig. 18 — Operator trunk circuit.

No. 1 ESS, with its network capable of transmitting dc signals at relatively large voltage and current levels (compared with speech signals), permits direct connections between trunks and specialized circuits designed for various types of pulsing. Trunk circuit bypass states remove all series and shunt impedances just as the cutoff contacts in the

line circuits do. By concentrating pulsing operations in a relatively small number of specialized circuits, control operations are greatly simplified. For instance, only a few scan points need to be scanned at a high enough rate to detect dial pulses, and only a small number of central pulse distributor outputs are required for the fast and frequent operation of outpulsing devices.

For customer lines, No. 1 ESS provides customer dial pulse receivers as described in Section III. TOUCH–TONE signaling[7] can be added by associating a TOUCH-TONE detector with a customer dial pulse receiver. A combination of the two circuits is called a TOUCH-TONE receiver, and such a combination can receive both conventional and TOUCH-TONE pulsing. The TOUCH-TONE detector is completely transistorized and converts ac tones generated in customer subsets to dc signals required by ferrod sensors in the scanner.

For trunk signaling, transmitters and receivers for dial pulsing, revertive pulsing and multifrequency pulsing are available. In addition, a panel call indicator transmitter has been designed. The multifrequency and dial pulse receivers and the revertive transmitter all detect pulses of the particular type from the transmission facility and convert such pulses to scanner signals. The multifrequency receiver is the most elaborate of the three, and, like the TOUCH-TONE detector, is transistorized.

The multifrequency, dial pulse and panel call indicator transmitters and the revertive receiver all use devices operated by the central pulse distributor to gate out suitable pulses. The multifrequency transmitter contains two transistor oscillators and a mixer to generate the required signal, and six devices operated by the central pulse distributor to cause the oscillators to produce the proper frequencies for each digit. The circuits which send dc pulses need fewer central pulse distributor outputs for the generation of loop opens and closures or high- and low-current conditions. However, they have much longer holding times and require separate circuits to convert their dc signals to voice-frequency tones for pulsing over carrier transmission facilities.

V. EQUIPMENT CONSIDERATIONS

Trunk, junctor and service circuits are constructed from a rather small universe of devices. Just three different codes of magnetic latching relays are used in signal distributor state-switching, and fewer than ten transmission components are needed for the configurations depicted in Figs. 3 and 9. Only in the more elaborate service circuits (which are supplied in relatively small quantities) does the number of different component

types reach any appreciable size. Here several codes of transistorized circuit packs are required along with several additional types of relays.

Wire spring magnetic latching relays, although slow by electronic standards, are the practical economic choice when switching operations need not be made very quickly or often. Their ability to switch a number of leads independent of each other and the activating coil is particularly advantageous. They require no holding power and are pulsed operated and released by signal distributors; the coils of all three codes have identical electrical properties.

Junctor circuits and a great many trunk and service circuits contain only latching relays and transmission components. A new series of inductors and transformers was developed for No. 1 ESS, and small mylar-foil capacitors were adopted. Uniformity in component size — all relays and transmission devices are four inches tall or less — makes possible very compact equipment arrangements.

With central processor control, circuits as well as components exhibit uniformity. As can be seen from Figs. 3, 7, 16, 17, and 18, the central processor sees all junctor circuits and many trunk and service circuits in terms of two scan points and three signal distributor points or less. Such standardization led to the development of a family of plug-in junctor and trunk units, using the simple angle-type sheet metal chassis shown in Fig. 19, which mates with the same connector used by printed wire boards.[8] Such units are surface wired with wire-wrap connections. In a typical office, about three-fourths of all trunk and service circuits are plug-in and mount on universal trunk frames as shown in Fig. 20. Junctor frames are similar in appearance. The center bay contains the scanner and signal distributor; the bays on the left and right contain positions for plug-in units. Wiring within each bay is highly standardized as a result of fixed plug-in connector terminal assignments and is completed at the factory.

Sixty-four equipment units (or up to 128 circuits) plug into a standard 26-inch-wide frame. This density is made possible by the use of seven inches of space in front of the frame rather than the four inches conventional mounting would have allowed. The actual mounting area on a frame is only eleven square inches per circuit.

In addition to saving space, a high density of circuits makes possible economical scanner and signal distributor design. Per-point cost drops with increasing size in both circuits, and high circuit densities also produce short lead lengths.

Digit transmitters, receivers and other trunk and service circuits which do not fit this standardized pattern are mounted in miscellaneous

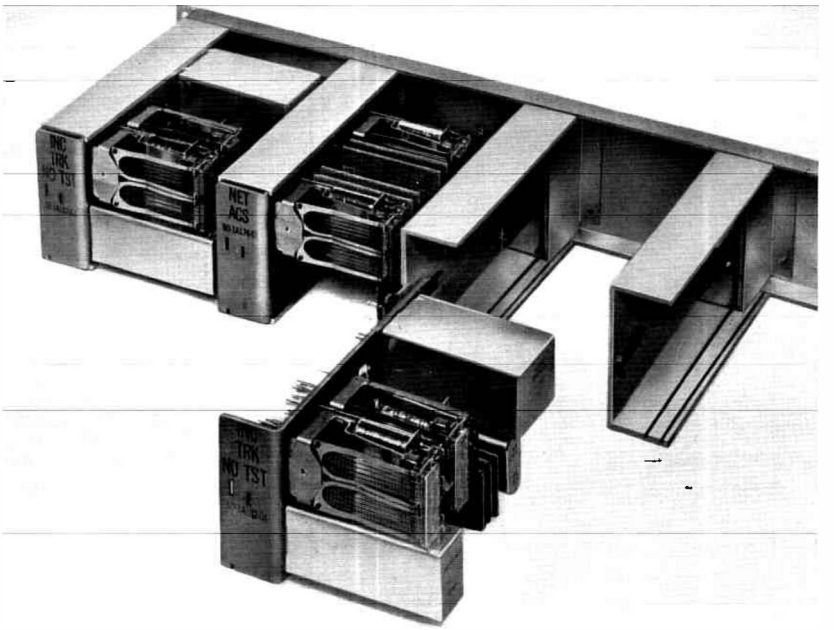


Fig. 19 — Plug-in trunk unit containing two circuits.

trunk frames. These units, typified by the pair of MF receivers shown in Fig. 21, often consist of combinations of semiconductor circuit packs, network and relays. Each unit has one or more terminal strips to simplify frame wiring (on the rear) and installer wiring (on the front).

In contrast to the plug-in units, this fourth of the trunk and service circuits requires detailed engineering to meet specific office requirements. The majority of these designs are single circuit units, although customer dial pulse receivers, TOUCH-TONE detectors, and MF receivers come two to a unit, and certain small auxiliary circuits come as many as 24 to a unit.

VI. MAINTENANCE

6.1 *Introduction*

Trunk and service circuit maintenance is based on three primary objectives: The system must (a) remove any faulty unit from customer use as quickly as possible, (b) pinpoint detected troubles to a very small

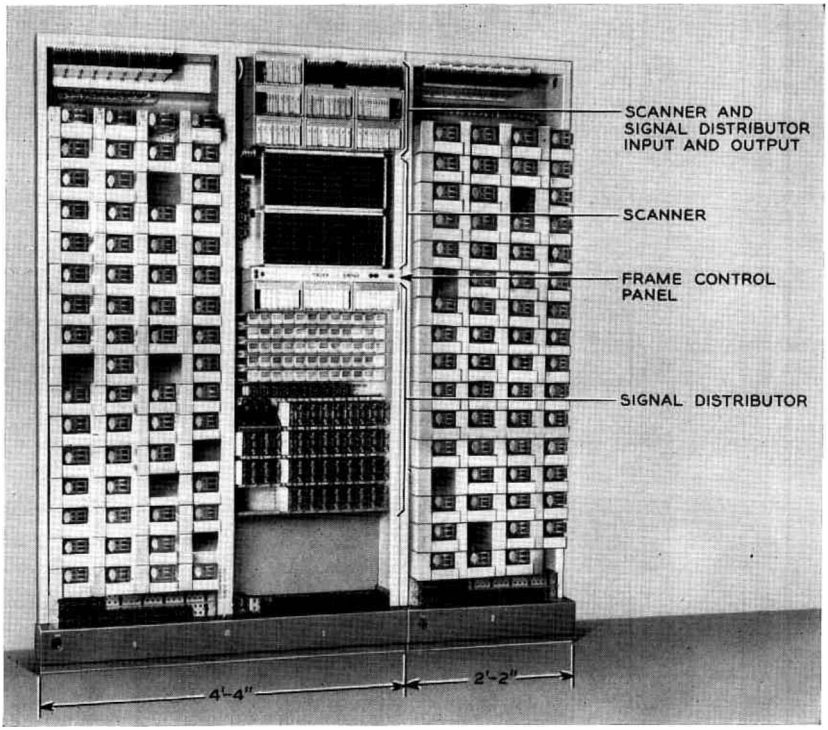


Fig. 20 — Universal trunk frame.

amount of hardware, and (c) keep the maintenance force informed using a minimum amount of teletypewriter output. To accomplish these objectives, symptoms are gathered by observing circuit and system abnormalities, analyzing the abnormalities, and drawing conclusions regarding any trouble detected. Maximum use is made of temporary memory, stored program and the switching network.

The four main phases of trunk and service circuit maintenance, consisting of trouble detection, diagnostic testing, trouble reporting, and circuit disposition, are depicted in the generalized flow chart of Fig. 22.

6.2 *Trouble Detection*

During customer calls, the system is always on the lookout for abnormalities. Holding times of digit receivers and transmitters and other service circuits are measured in the central processor to detect various stuck conditions. Up-checks and down-checks are built into every signal

distributor operation. Certain circuits contain special check states for rapid testing of large portions of the circuit. The customer dial pulse receiver and ringing circuits provide tests for certain outside plant troubles on each originating and terminating call. The transfer of supervision from one circuit to another checks network continuity and proper functioning of circuit relays.

Many call abnormalities are indications of trouble conditions which could be in one of several circuits. In these cases, various decisions are made to isolate the faulty circuit. If, for example, upon connection of a line to a dial pulse receiver an off-hook condition is not observed, the fault may lie with the receiver, the network connection, the line ferrod, or the loop. By substituting a second receiver or network path or checking the line ferrod, faults in these areas of circuitry can be deduced.

Most customer call abnormalities are indications of circuit faults; a few are the result of traffic overload conditions. Some screening is done to separate these two conditions and prevent unnecessary testing. Such screening would be used upon failure to receive start-pulsing signals from a terminating office after the originating office transmits seizure signals. Counts are kept in temporary memory for each transmitter indicating the number of times this has occurred. If the counts are evenly distributed over all transmitters, a traffic overload condition is assumed to exist. A diagnostic test on a transmitter is carried out only when its failure count is significantly higher than the failure counts of other circuits.

Outgoing trunks are treated in a similar manner. However, per-trunk memory is not available for this function and time sharing of temporary

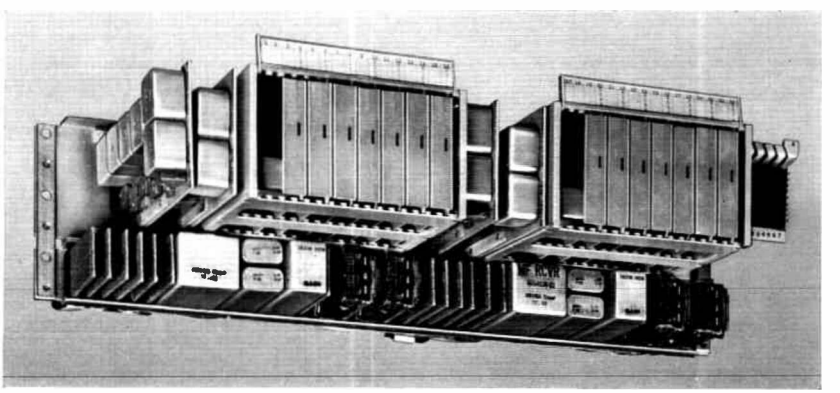


Fig. 21 — Equipment unit containing two MF receivers.

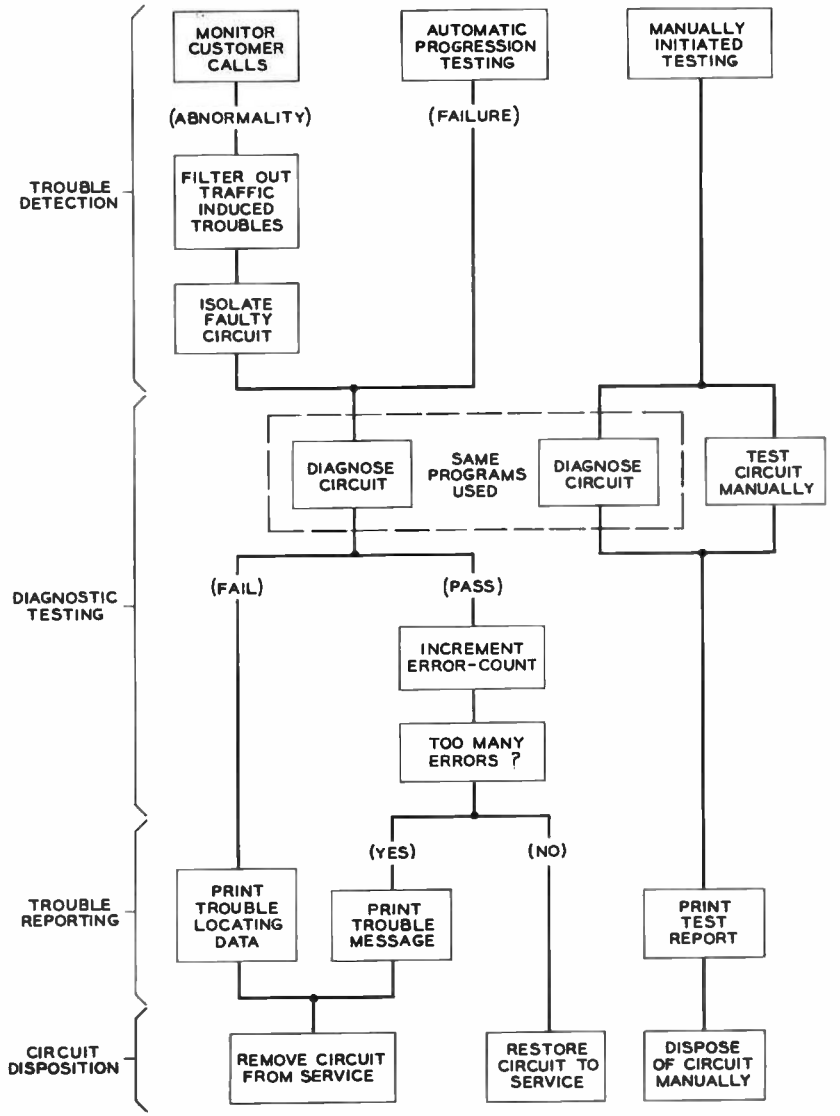


Fig. 22 — Generalized flow chart for trunk and service circuit maintenance.

memory must be used. A few memory words are temporarily assigned to one trunk group or a part of one trunk group to store failure rate information. This information is analyzed in a manner similar to transmitter failure information, preventing undesirable and meaningless testing of outgoing trunks.

While information is being gathered on one trunk group, other groups may be encountering similar difficulties. These indications are counted in a single counter for all the remaining trunk groups in the office. A sufficiently high count will cause detailed observations to be stopped on one trunk group and started on some other trunk group. Thus a small amount of memory can lead to the identification of a faulty trunk with a minimum amount of teletypewriter output for the maintenance man.

6.3 *Diagnosis*

In general, a number of small test segments is needed to test any trunk or service circuit. Each program test segment requires a circuit or combination of circuits to carry out normal functions. Trunk, service, and test circuits, as required, are connected together through the switching network. Correct circuit operation generates a particular set of data, which the central processor stores in temporary memory. If a circuit should malfunction during the test segment, a different set of data, which can be thought of as a "trouble symptom," would be generated. Rather than stopping upon receipt of the first trouble symptom, all test segments are carried out. Much more can be learned from a complete set of symptoms than from just the first one encountered.

When all test segments are complete, a large quantity of data has been accumulated. Diagnosis is completed by comparing these data with a block of "all tests pass" data stored in the permanent memory. Failure to match means a fault exists in the circuit under test, and the test data stored in the temporary memory are printed out for the maintenance man. He can isolate the faulty components through use of a dictionary which was prepared by simulating faults in the circuit under test and recording the trouble symptom.[9]

Many tests are available for customer lines, trunks and service circuits following the basic pattern outlined above.

Outgoing trunks to distant offices can be checked automatically for supervision and digit pulsing as well as tone returned from the distant office. This is done by placing program-initiated and controlled test calls to test lines in the distant office and detecting the response signals from them.

Incoming trunks can be tested for permanent signals, called customer supervision and ability of trunk circuit relays to change state; such tests are made without test calls being initiated by the distant office. Outgoing operator trunks can be tested with the aid of suitable recorded announcements which inform the operator that a test is being made, that the system expects her to take a prescribed set of actions, and that the system will interrogate the trunk circuit responses.

Tests of the type described above are carried out in response to call processing failures. The test programs are also used for automatic trunk progression testing; a control program is added to indicate to the diagnostic programs which circuits are to be tested and in what order. When a trouble is indicated on the first pass through the diagnostic program, the control program treats this as a trouble detected; the diagnostic program test is then repeated. Failure of the second pass causes removal of the faulty circuit from service and a trouble print-out to the maintenance man.

Safeguards are included to prevent the automatic removal of too many circuits from service. Whenever two successive automatic tests of the same type fail, any common test circuitry is checked before normal testing is resumed. Also, the system cannot automatically remove from service more than a fixed percentage of the circuits in any group.

6.4 *Error Analysis*

When a trouble is detected by the failure of a check during call processing or progression testing but a subsequent diagnostic test yields an "all tests pass" result, the system records an "error" for the circuits under test. Errors can be caused by transient noise pulses, unstable circuits, marginal conditions not simulated during diagnostic tests, dirty relay contacts, and the like. Maintenance error counts are kept on a limited number of trunks at any given time. These running counts are updated and compared to removal-from-service criteria. If this comparison shows the error rate to be too high, the trunk is removed from service. When a removal from service occurs, the maintenance man is notified and given the trunk circuit identity. If any trunk does not accumulate enough errors to warrant removal from service, its space in the error counting facility is given to a new trunk where an abnormally high error rate is suspected. Periodically, the maintenance man is given a summary error count to indicate the level of performance of these circuits.

6.5 *Manual Test Control*

The trunk and line test panel (T&LTP) in the master control center,[10] shown in Fig. 23, makes possible another battery of tests. Customer lines, trunks and service circuits can be connected to the master control center by using TOUCH-TONE callers on a master test line. Lines having permanent signals can be connected by a program handling a temporary memory queue which contains the identity of lines awaiting attention. Manual keys control the access circuitry; through additional keys, the line, trunk or service circuit in question is connected to voltmeter or transmission test circuits or jacks for portable test equipment. The TOUCH-TONE caller can also be used to control the setting of relay states in trunk or service circuits; this permits manual trouble-shooting.

The TOUCH-TONE caller can be used to specify a trunk or service circuit on which the maintenance man wishes to run a diagnostic program once or repeatedly. Repeated testing is stopped at the end of a complete diagnostic program test if any failure occurs. A teletypewriter message gives the number of times the diagnostic program was completed before failure and the usual diagnostic information on the failed test. If a circuit to be tested is busy, a camp-on feature notifies the maintenance man when it is available.

6.6 *Summary*

Extensive checking during normal customer calls leads to rapid trouble detection. Diagnostic test programs remove faulty units from customer service. Gathering meaningful data during diagnosis pinpoints troubles to a small amount of equipment. Screening before and after diagnostic tests are performed, coupled with the gathering of meaningful diagnostic data, helps keep the maintenance force informed with a minimum of teletypewriter output. Diagnostic programs are available for each type of trunk or service circuit; these programs can be used in many ways both by the machine and the maintenance man. With these aids, the maintenance man should have little difficulty in isolating troubles. Replacement of defective circuits is facilitated by the extensive use of plug-in units.

VII. CONCLUSIONS

The line, trunk, junctor and service circuits in No. 1 ESS have been designed to take full advantage of stored program techniques.

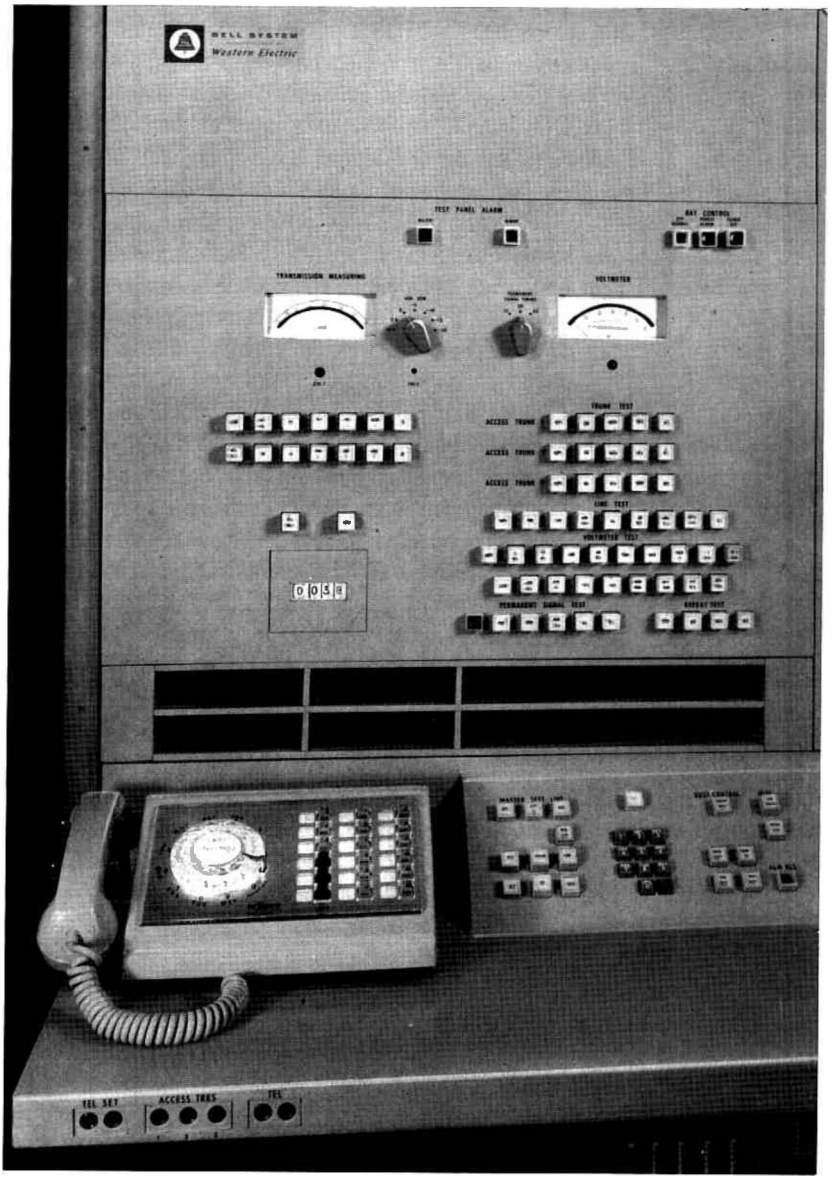


Fig. 23 — Trunk and line test panel in the master control center.

Because the hardware has been reduced to a minimum, the great majority of changes which may eventually be required can be made entirely within the program itself. The isolation of functions in service circuits permits the addition of new features or the deletion of old without changing anything except the service circuits involved and the translation information in memory. Programmed maintenance techniques make possible equally simple changes in testing. With this flexibility, No. 1 ESS should meet successfully the challenges of the future.

VIII. ACKNOWLEDGMENTS

Many persons have contributed to the work described in this paper. The authors are particularly indebted to A. E. Joel and J. M. Nervik for their help during the initial planning stage and A. Feiner and R. E. Staehler for their helpful suggestions and advice. K. H. Muller made many contributions, particularly in mechanical design. Mention should also be made of R. N. Battista, F. Elenbaas and C. G. Morrison for their assistance in connection with voice-frequency signaling.

REFERENCES

1. Harr, J. A., Taylor, F. F., and Ulrich, W., Organization of No. 1 ESS Central Processor, B.S.T.J., this issue, p. 1845.
2. Harr, J. A., Hoover, Mrs. E. S., and Smith, R. B., Organization of the No. 1 ESS Stored Program, B.S.T.J., this issue, p. 1923.
3. Carbaugh, D. H., Drew, G. G., Ghiron, H., and Hoover, Mrs. E. S., No. 1 ESS Call Processing, B.S.T.J., this issue, p. 2483.
4. Freimanis, L. Guercio, A. M., and May, H. F., No. 1 ESS Scanner, Signal Distributor, and Central Pulse Distributor, B.S.T.J., this issue, p. 2255.
5. Karnaugh, M., The Map Method for Synthesis of Combinational Logic Circuits, Trans. A.I.E.E., Part I, Comm. and Electronics, Nov., 1953, pp. 593–599.
6. Hinderliter, R. G., Transmission Characteristics of Subscriber Loop Plant, IEEE Trans. Comm. and Electronics, No. 68, Sept., 1963, pp. 464–470.
7. Battista, R. N., Morrison, C. G., and Nash, D. H., Signaling System and Receiver for TOUCH-TONE Calling, IEEE Trans. Comm. and Electronics, No. 65, March, 1963, pp. 9–17.
8. Chevalier, J. G., and Eisenhart, R. K., No. 1 ESS Circuit Packs and Connectors, B.S.T.J., this issue, p. 2241.
9. Downing, R. W., Nowak, J. S., and Tuomenoksa, L. S., No. 1 ESS Maintenance Plan, B.S.T.J., this issue, p. 1961.
10. Dougherty, H. J., Raag, H., Ridinger, P. G., and Stockert, A. A., No. 1 ESS Master Control Center, B.S.T.J., this issue, p. 2283.

No. 1 ESS Apparatus and Equipment

By J. C. FERGUSON, W. E. GRUTZNER, D. C. KOEHLER, R. S. SKINNER, M. T. SKUBIAK and D. H. WETHERELL

(Manuscript received January 15, 1964)

No. 1 ESS provides a greater variety of services than any prior complex switching system, in central offices of greater capacity, but with more highly standardized equipment of much smaller volume. Much of the credit for this achievement is due to the use of fast electronic circuits under control of a generic program. Apparatus and equipment development was aimed at accenting the benefits in this system, disciplining options to concentrate demands on a few standard frame building blocks, minimizing the varieties and codes of apparatus to take advantage of economies inherent in large volume production, and combining these with flexibility and versatility in a dependable, maintainable, compatible system that will provide all services wanted now and in the future.

I. INTRODUCTION

The goal throughout the No. 1 electronic switching system (ESS) development[1] was to achieve the highly standardized modular design which will be most economical to engineer, manufacture, install, operate, maintain, and administer. As a result, No. 1 ESS uses a limited variety of frames as building blocks and relies on a generic stored program to provide most office-to-office variations. The system represents a giant step forward in combining versatility and flexibility with standardization in switching systems.

This advance in the switching art comes at a most appropriate time. Many of the earliest dial offices are nearing retirement age, and most of them are in buildings too full of equipment to accommodate replacing units of crossbar equipment. No. 1 ESS will avoid the cost of establishing new wire centers in new buildings in these cases. The installation of a first No. 1 ESS which is to serve as a replacement for existing equipment can be made in a toe-hold of as little as 2000 square feet of floor space. This can ordinarily be made available. This first installation will replace two

or three times its volume of panel or step-by-step equipment and clear space for large future additions. In this way, existing buildings may ultimately house several times as much switching equipment as they were originally designed for.

There is possibility of a further large dividend. In the past, the Western Electric Company has found it generally uneconomical to manufacture large central office equipments in anticipation of demand. There were too many equipment variables as well as too many new features which continually require design changes. As a result, large central offices have been built to customer order. Now, with the elimination of most equipment variables in No. 1 ESS, and with the facility to add new features by program change alone, line assembly in advance of orders promises large savings in production and investment costs. Consolidation of demands will increase lot sizes in the shop and permit a more uniform flow of production. The present interval between placing an order for a switching system and cutting it into service will be shortened. The unproductive investment period will be correspondingly reduced. Savings will be further enhanced by the concentration of demands for those frames used in common by local, tandem, and toll switching systems.

The outstanding attributes of No. 1 ESS equipment include:

(a) provision in the ultimate for the widest variety of services ever offered in one switching system: all of those now available in the several local, toll, and tandem switching systems for large Bell System installations as well as many new services

(b) flexibility for economical growth and the provision of new features not practical in existing systems, plus the ability to work with all systems

(c) use of just one pair of central processors[2] to serve offices varying in size up to 65,000 lines. This permits the replacement of several existing offices with one No. 1 ESS office

(d) highly automated and sophisticated accounting, maintenance,[3] traffic, and traffic-measuring facilities

(e) relatively few small standardized frames of equipment (with most of the usual variables eliminated)

(f) highly standardized floor plans arranged on a modular basis

(g) small building volume for switching equipment ($\frac{1}{3}$ of previous areas, $\frac{1}{4}$ of previous volumes)

(h) wide battery voltage tolerances.

The more important factors in the equipment design are:

(a) Semiconductor devices and new temporary and semipermanent memories in electronic circuits appreciably increase the operating speed of the system.

(b) Programs in memory replace most of the wired logic required in earlier systems; this simplifies circuits and avoids innumerable apparatus and wiring options.

(c) Communication between equipments by high-speed digital pulses over a relatively few pairs further simplifies the equipment and cabling.

(d) Ferreed switches and ferrod sensors and the simplification of their controls and communicating links compress networks.

(e) Shared trunk controls simplify trunks and service circuits.

(f) New techniques are exploited to reduce operating effort as well as to eliminate much apparatus and wiring in equipment for AMA, maintenance, traffic-measuring, and auxiliary services.

(g) New concepts in protector and distributing frames take advantage of electronic memories and other No. 1 ESS innovations to minimize cross-connection changes, jumper lengths, and frame volumes.

(h) The design of all equipment, framework, wiring, and cabling eliminates options wherever possible.

II. NEW EQUIPMENT MODULES ARE STANDARDIZED

Electromechanical switching systems have come along one after another in orderly evolution. New equipment practices have been introduced one or two at a time, and field experience following each innovation has proved its value. No. 1 ESS (see Fig. 1) is so different in so many ways that it represents a switching revolution. This posed a challenge to develop new forms of apparatus and equipment modules that exploit to the full the capabilities of these new system concepts.

Questions which were considered in the combining of components into frames of one consistent design were: (a) How should semiconductor circuits be packaged in a uniform manner with the fewest variables for their many uses throughout the system? (b) How should ferrite sheet and twistor memories be designed for the stores? (c) How should ferreed switches and ferrod sensors be designed to become network crosspoints and scan points? (d) How should simplified trunks and common trunk control equipment be designed? (e) How should AMA recorders and teletypewriters fit the new maintenance and traffic record concepts? (f) How should power, alarm, and auxiliary services be provided?

The design of a frame could not be frozen until all combinations of units which were to be mounted on it were known. Other factors also had to be kept in mind: frames would have to be interconnected in standardized floor patterns in office buildings; the basic package designs would greatly influence frame associations and interframe cabling. In the foreground of all design decisions was the goal of standardization without optional equipment variables.

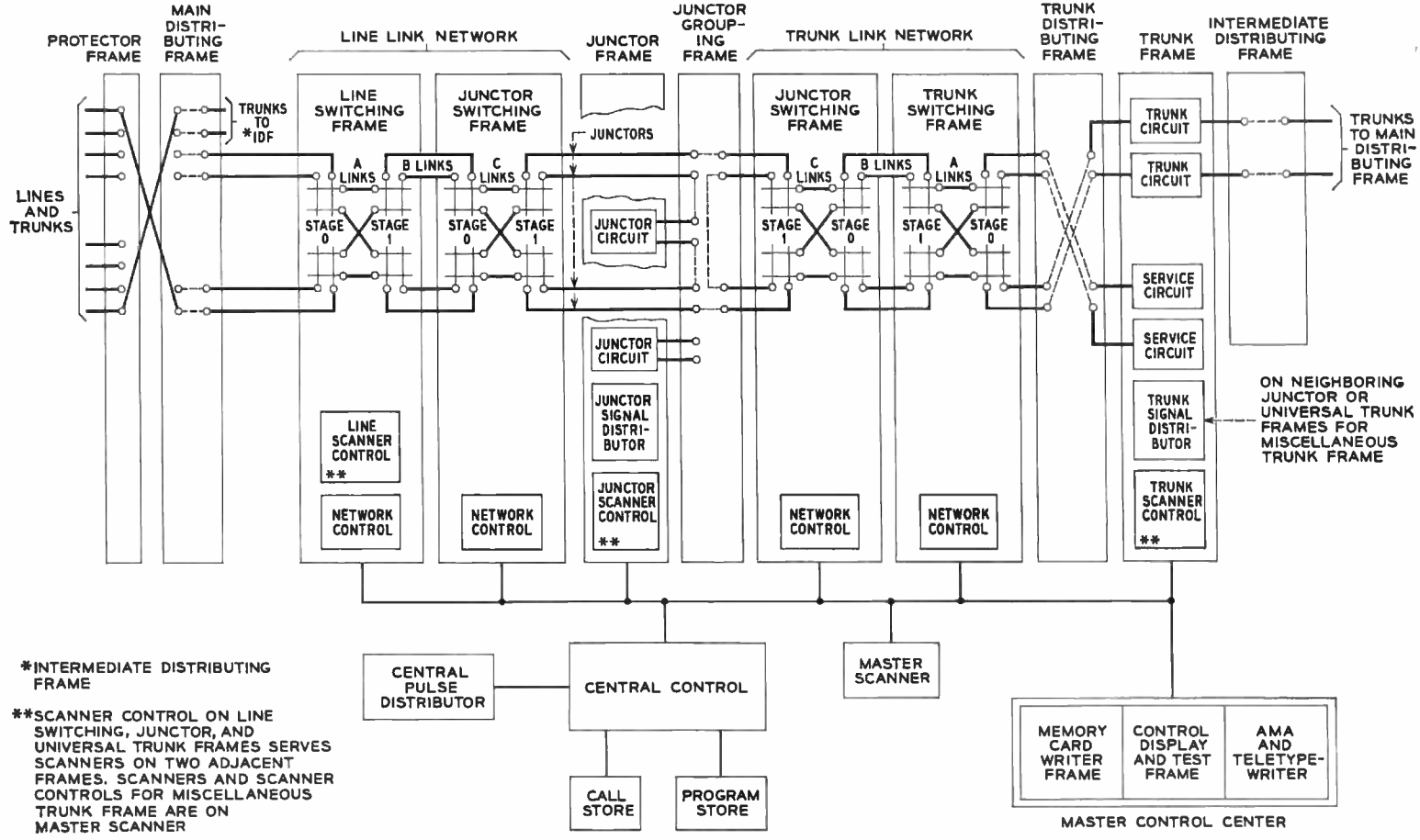


Fig. 1 — Equipment schematic.

2.1 *Basic Packages*

Some of the basic packages developed are: (a) semiconductor circuit packs, (b) ferreed switches, (c) ferrod sensors, (d) trunk and junctor plug-in units, (e) twistor memory, and (f) ferrite sheet memory.

In each of these designs, two basic decisions had to be arrived at simultaneously. First, all possible circuit configurations for functions everywhere had to be considered and reduced to the fewest possible. Second, the innumerable ways for packaging had to be studied and the basic design selected. Then the variables in each case had to be adapted to the standard package with the very minimum of codes. This brought production economies as well as maintenance economies in both equipped and spare packages.

2.2 *Frames*

Concurrently with the basic package developments, frame designs were explored. The frame structure had to be as nearly universal as possible to accommodate all varieties of equipment combinations in the best manner for shop and field. Frames for earlier large dial systems were of a variety of constructions but generally were single-sided and 11 feet, 6 inches high. Many equipment and building standards had been developed over the years around these designs. For the Morris, Illinois, ESS trial installation,[4] double-sided cabinets 7 feet high with doors front and rear were decided upon to minimize lead lengths and to facilitate air conditioning. Requirements for this new system had to be carefully studied to arrive at the ideal frame. Should it be single-sided or double-sided? Should it follow the 11-foot, 6-inch standard, or were there compelling reasons for a new height? Should it be an enclosed cabinet?

2.2.1 *Double-Sided versus Single-Sided Frames*

A study of double-sided versus single-sided frames proved that the very minor floor space advantage in favor of double-sided frames was offset by production and maintenance advantages favoring the single-sided frame. Air conditioning of individual frames, which dictated enclosed double-sided cabinets for Morris, was no longer a requirement. In addition, frame covers were eliminated to reduce cost and to ease system maintenance.

2.2.2 *Frame Studies*

A series of studies determined that the single-sided frame should be 7 feet high with modular widths of 1 foot, 1 inch. Reasons for adopting this design fell into three classes.

(*i*) *Equipment Arrangements:*

(a) It fits the circuit functions into more orderly, symmetrical equipment modules.

(b) It provides short, direct pulse leads for intra- and interframe communication.

(c) It gives floor loads compatible with existing buildings.

(d) It permits economical use of office volume with attractive equipment and building designs.

(*ii*) *Maintenance:*

(a) Access from the floor is preferable to that from rolling ladders.

(b) Accident hazards are less with no rolling ladders.

(*iii*) *Building Costs:*

(a) Considering existing buildings with high ceilings, a possible small floor space loss is offset by savings from the omission of auxiliary framing and rolling ladders.

(b) Considering buildings with low ceilings, there are important additional savings in new building costs and in the ability to install 7-foot frames in commercial buildings.

2.2.3 *The Frame Adopted*

The frame finally adopted is shown in Fig. 2. Sheet metal uprights of 1-inch by 5-inch cross section are centered on the base to provide an $8\frac{1}{2}$-inch depth for apparatus on the front and a $3\frac{1}{2}$-inch depth for wiring on the rear. Frames having five widths on 1-foot, 1-inch modules provide for all equipment.

III. FRAME EQUIPMENTS ARE STANDARDIZED

No. 1 ESS employs new types of apparatus for most of its functions: ferreed switches for network switching; ferrod sensors for scanning; magnetic latching wire spring relays in trunks; twistor and ferrite sheet memories; semiconductor devices in plug-in circuit packs for almost all logic and controls; and others as described in detail in Section V. These components, as needed, are arranged on the 24 frames listed in Table I in ways which make each frame, as nearly as practicable, a completely functional building block free of options. No. 5 crossbar at the same stage of development had approximately 3 times as many frame equipments, 10 times as many trunk equipments, and 100 times as many coded variables for apparatus and wiring options.

The equipment arrangements developed for the different frames follow a standard pattern. Terminal strips, bus transformers, and other apparatus

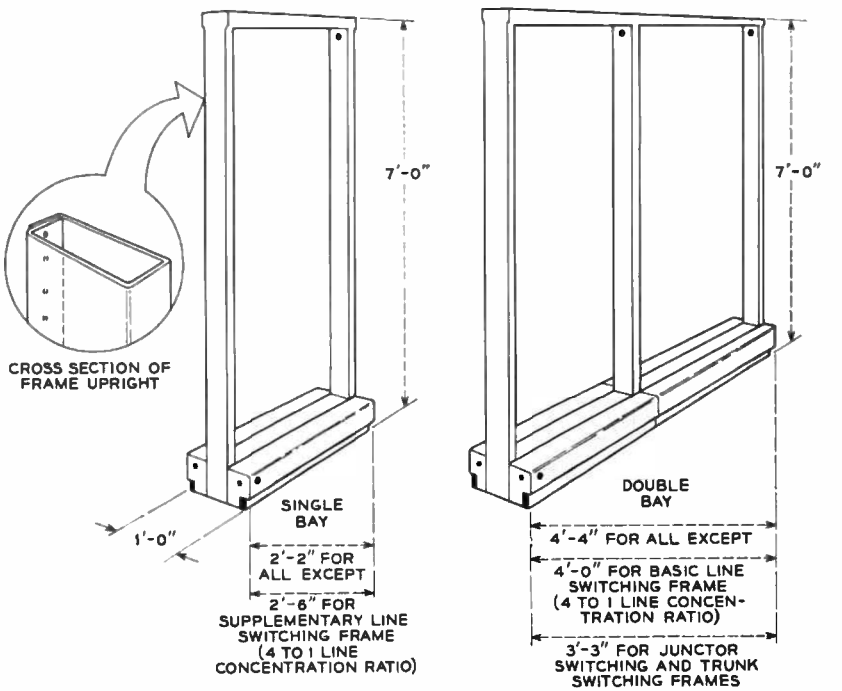


Fig. 2 — Frame.

requiring little maintenance are at the top of each frame, so that practically all maintenance work can be done without the use of stepladders. Frame filters, fuses, and power cutoff relays, as required, are located at the bottom of each frame. Appliance outlets are in the front and rear of each frame base. Every frame has a control panel with telephone and spare jacks and pin jacks for −48 volts, +24 volts, ground, and high-resistance ground. Most frames, in addition, require alarm, off-normal and out-of-service lamps, power cutoff keys, and some keys for special purposes; these are also mounted on the frame control panel located at a convenient height above the floor.

3.1 *The Control Panel*

The typical control panel shown in Fig. 3 carries a group of push-button keys, indicator lamps, and test jacks in a perforated steel housing. The housing acts as a shield to protect adjacent electronic equipment, while the perforations permit vertical passage of air if needed to cool

TABLE I — FRAMES, ABBREVIATIONS, AND LENGTHS

Frame	Abbreviation	Length		Remarks
		Feet	Inches	
Central control	CC	8	8	2 per office
Program store	PS	10	10	
Call store	CS	2	2	
Master scanner	MS	2	2	
Central pulse distributor	CPD	2	2	Operate in pairs
Master control center	MCC			
memory card writer	MCW	2	2	1 per office
control display & test	CDT	4	4	
AMA-teletypewriter	AMA-TTY	4	4	
Line switching	LS			
2 to 1 line concentration (home or mate)		4	4	
4 to 1 line concentration				4 to 8 per LLN
basic (home)		4	0	
supplementary (mate)		2	6	
Trunk switching	TS	3	3	4 to 8 per TLN
Junctor switching	JS	3	3	4 per fully equipped LLN and TLN
Junctor	J			
home or mate — basic		4	4	
home or mate — supplementary		2	2	
Universal trunk	UT			
home or mate — basic		4	4	
home or mate — supplementary		2	2	
Miscellaneous trunk	MT	2	2	
Miscellaneous	M	2	2	
Recorded announcement	RA	2	2	
Power distributing	PD	2	2	
Miscellaneous power	MP	2	2	
Ringing and tone	RT			
½-amp capacity		4	4	
6-amp capacity		6	6	
Protector	PROT	6	6	
Main distributing (or IDF 8′ high)	MDF	6	6	Modules of these lengths are ordered as required
Intermediate distributing	IDF			
Trunk distributing (or IDF 7′ high)	TDF	4	4	
Junctor grouping	JGF	2	2	Ordered in pairs

that equipment. The keys are used primarily to disconnect or restore power to various sections of the frame for maintenance purposes, although test functions are also sometimes provided. The keys are mechanically interlocked to guarantee that if one duplicated frame or bus control section has power off, it must be restored to "power on" before the mate control section is turned off. Red lamps indicate trouble conditions, including "power off" in any section, while white lamps indicate an "off normal" condition and whether either frame control is "out-of-

service" for any reason. The telephone jacks at the left end of the panel permit convenient telephone connections to maintenance or test personnel at other locations in the office, and the pin-type jacks provide voltage sources for frame maintenance or for portable test sets.

3.2 *Filter and Fuse Panels*

Each frame is equipped with a filter panel designed to restrict the rate of current change on the frame supply feeders. This filter limits the noise transmitted to other frames via the centralized power distributing frame. These filter panels are located in the frame base immediately below the frame fuse panels. Thus the power feeders, terminating on the filter panel after entering via the frame upright, have a minimum exposure to stray noise sources. Power cutoff relay panels, when required, are located in this same area of the frame.

3.3 *The Circuit Packs*

The plug-in circuit packs engage connectors on an apparatus mounting which occupies 4 inches of vertical space. The 36A apparatus mounting is used for the great majority of circuit packs. Three adjacent mountings fit across a 2-foot, 2-inch bay, and each mounts as many as 16 circuit packs on 0.4-inch centers, or correspondingly fewer of those packs that require 0.8-inch or 1.2-inch centers. The 38A apparatus mounting mounts a single circuit pack and is used in locations requiring too few circuit packs to justify the larger mounting.

3.4 *Relay Equipment*

Many trunks have their relays and other components for one or two circuits on small plug-in units which mount interchangeably in supports and sockets on a trunk frame which is universally wired for all of them.

Fig. 3 — Typical frame control panel.

3.5 *Ferreed Switches and Ferrod Scanners*

The ferreed switch affords much greater flexibility than the crossbar switch in possible arrangements of crosspoints and controls. Advantage has been taken of this flexibility to provide the best switch array for each stage of line and trunk switching in the networks. Line, trunk, junctor, and master scanners all have 1024 (64 × 16) point matrices equipped with ferrod sensors having the proper operating characteristics.

3.6 *Terminal Strips and Bus Transformers*

A universal design of terminal strip and transformer using molded wire techniques is used for all interframe wiring which connects at the top of any frame (see Fig. 4). The conductors of bus systems, which address the frames, are terminated on ferrite core pickoff transformers. These transformers are molded directly into the molded ladder terminal arrays of the terminal strips. Several codes of these bus transformers, terminal strips, and combination units have been made available.

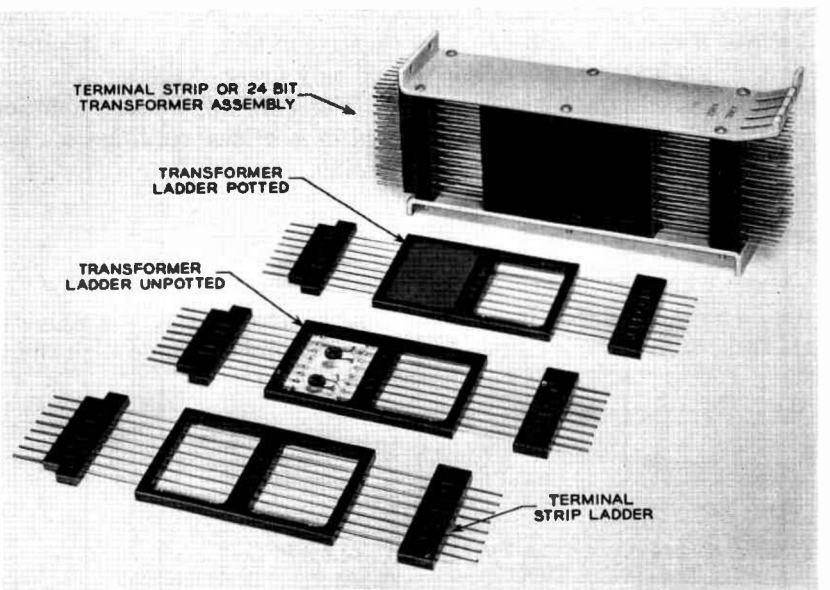


Fig. 4 — Terminal strip and transformer for interframe bus system.

3.7 *Frame Wiring*

In general, all shop wiring is to terminals on the rear, and installer wiring is to terminals on the front of frames; and, in general, both use 26-gauge conductors. Shop wiring includes surface wiring, loose wiring, and local cable of conventional kinds, as well as some new wiring techniques. Where high densities of circuit packs and wired interconnections exist, as in central control, special wiring procedures prescribe specific routings for surface wiring, loose wiring, and local cable.

3.8 *Designations*

Designations on equipment give frame name, frame and bay numbers, and specific functions in accordance with nomenclature and abbreviations standardized for the system.

IV. FRAME ARRANGEMENTS IN AN OFFICE ARE STANDARDIZED

Engineering, installing, operating, and maintenance costs are reduced when frame arrangements in an office are standardized. Important considerations here involve: (a) arrangements of frames on floors, (b) office and frame capacities, (c) cable rack and office lighting, (d) end guards for end of frame line-ups, and (e) interframe cabling and wiring.

Much attention has been given to each of these items as well as to appearance, with results as shown in the figures. Missing from these illustrations, however, is an important feature — the attractive color scheme. Cable rack enclosures, end guards, and frame bases are a dark shade of blue, which makes an attractive contrasting border around the light blue-gray of the frames within each line-up.

4.1 *Floor Plans*

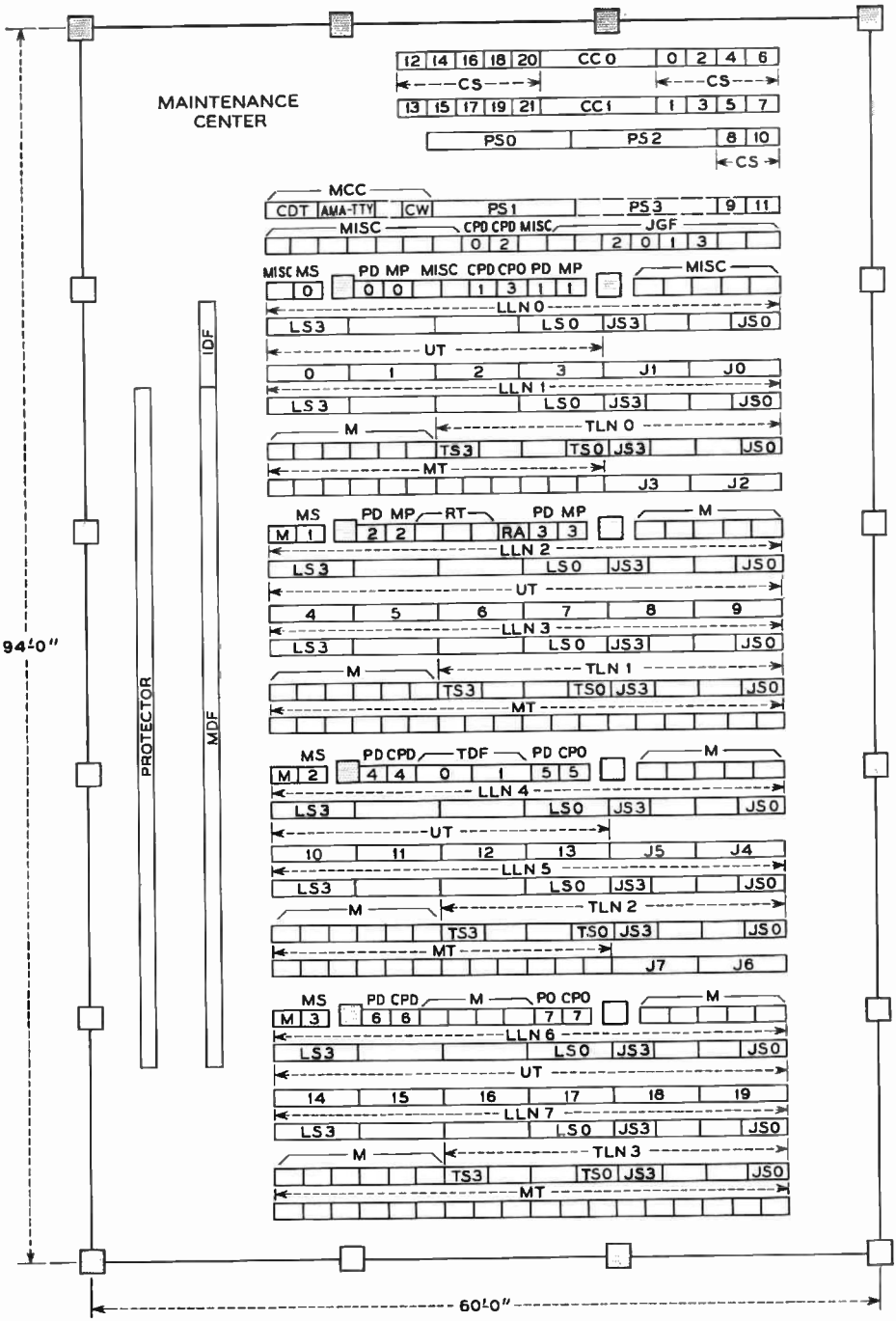
A universal floor plan pattern has been developed which:

(a) Grows naturally from the smallest to the largest installation. With a few minor variations, it makes efficient use of floor space at every size and for all traffic concentrations.

(b) Permits an office to start with one line concentration ratio and convert later to another if traffic density changes make it desirable.

(c) Provides short, direct cable runs that minimize cable costs and electrical interference.

(d) Locates the master control center, central controls, and stores



1. SHOWN ARE 8 LLN's FOR 32,768 LINES WITH 4 TO 1 LINE CONCENTRATION RATIO AND 4 TLN's FOR 4,096 TRUNKS WITH 1 TO 1 TRUNK CONCENTRATION RATIO
2. JGF IS CENTRALLY LOCATED FOR ULTIMATE GROWTH TO DOUBLE THE OFFICE SIZE WHETHER GROWTH IS TO REAR, TO ONE SIDE OR, TO ANOTHER FLOOR
3. SEE TABLE I FOR KEY TO ABBREVIATIONS

Fig. 5 — Typical floor plan.

together in preferred locations in one area. Space for about half of the stores is in the initial building, and the remainder is in an addition.

(e) Employs standard building bays and provides for building growth to the rear, to one side, or to a floor above.

(f) Aligns the protector frame and the main distributing frame (MDF) with associated network frames for orderly growth together, in a way that automatically shortens cables and MDF jumpers.

(g) Permits the same pattern to be followed in all new buildings and adapted to existing buildings.

The pattern, as applied to a typical office, is shown in Fig. 5.

The new MDF is parallel to the long building wall, with all frame line-ups perpendicular to it. Central control, store, and maintenance frames are in preferred locations in one area. Network, trunk, and other frames are in building bay modules which grow with the MDF.

Buildings will require a minimum ceiling height of 10 feet under beams. Floor loads are the standard 150 pounds per square foot: 100 pounds for equipment and 50 pounds for cable rack, interframe cable, and maintenance personnel.

4.2 *Office and Frame Capacities*

The more important capacities are shown in Table II.

4.3 *Cable Rack and Office Lighting*

The cable rack, which conceals and shields all interframe cabling, is frame-supported over each line of frames and across aisles (see Fig. 6). A cable rack stanchion (3½ inches in diameter) is used to support cable rack where frames are omitted for spans of 10 feet or so. The cross-aisle racks are placed at each end of a line-up and at intermediate points as needed. This system of frame and cross-aisle cable racks not only provides for routing and segregating of cables but rigidly interconnects the frames and line-ups, giving great stability to the overall structure.

Also, since the frames are low and the aisles are largely free from overhead racks, excellent illumination is obtained either from ceiling lights or from frame-supported lighting, both of which are standardized.

4.4 *End Guards*

End guards are used at main aisle ends of frame line-ups. Each has a swinging door to give access to cables and equipment inside. Wherever one or more frames are omitted in a line-up, each exposed frame end is dressed with an end guard.

TABLE II — OFFICE CAPACITIES

Capacity	Number per Office
Total busy hour calls (100 seconds each), intra-office plus incoming plus outgoing	100,000 approx.
Directory numbers	
number groups	128
numbers per group	1,000
office codes	32
Networks	
Line link network with 2 to 1 line switching	
Conc. Ratio / Lines	
2 to 1 / 32,768	
2½ to 1 / 40,960	
3 to 1 / 49,152	16
3½ to 1 / 57,344	
Line link network with 4 to 1 line switching	
Conc. Ratio / Lines	
4 to 1	16
5 to 1	12⅘
6 to 1 / 65,536	10⅔
7 to 1	9⅐
8 to 1	8
Trunk link network	
Conc. Ratio / Trunks	
1 to 1 / 16,384	
1¼ to 1 / 20,480	
1½ to 1 / 24,576	16
1¾ to 1 / 28,672	
2 to 1 / 32,768	
Frames	
Central control	1 pair
Program store	6
Call store	37
Master control center	1
Master scanner	As required
Central pulse distributor	8 pairs
Power distribution	1 per 400-amp peak load for each of −48 volt and +24 volt
Ringing and tone (½-amp capacity on 2 bays or 6-amp capacity on 3 bays)	1
Junctor grouping frame modules	4 pairs

Equipment in the main aisle end guards (see Fig. 7) includes plates of bus terminating resistors.

On the outside of the end guards are the aisle alarm lamps, the aisle directory for designating the frames in each line-up, and light control switches. A spare fuse holder is provided on the door.

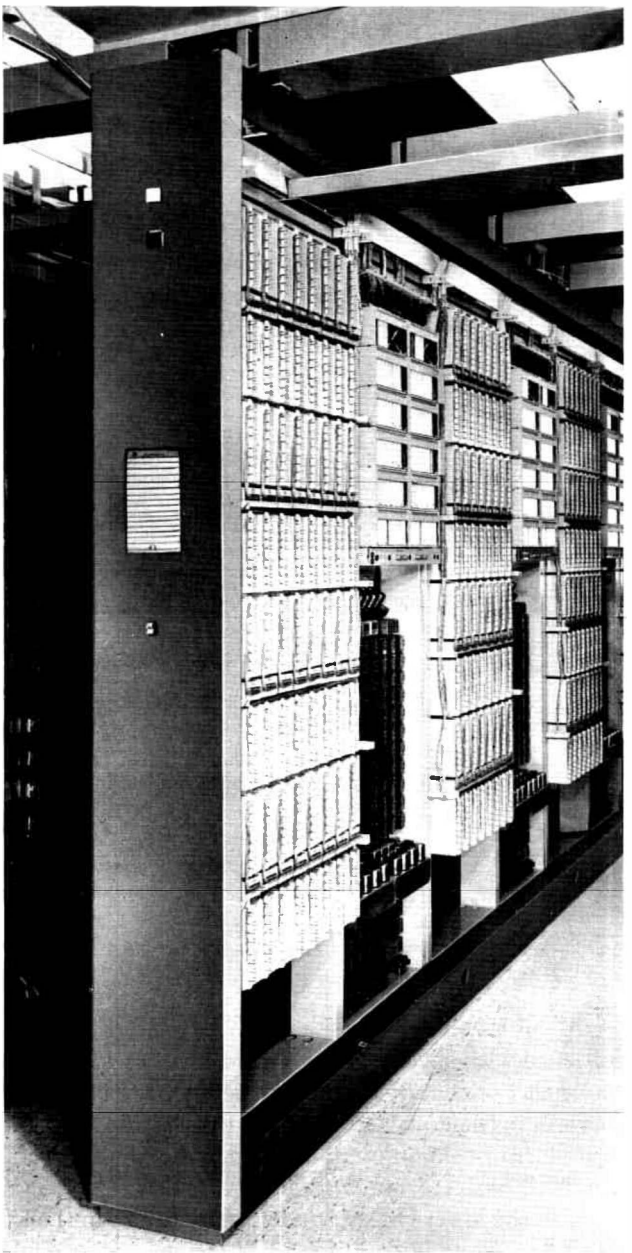


Fig. 6 — Typical maintenance aisle showing cable rack and lighting.

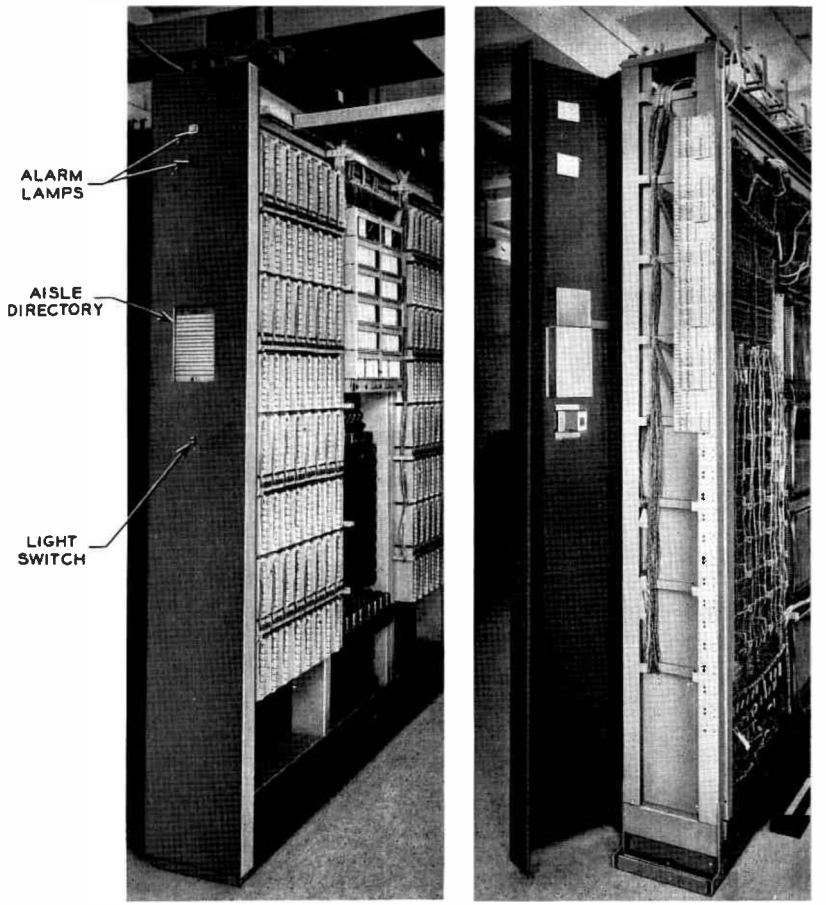


Fig. 7 — End guard and equipment.

4.5 *Interframe Cabling and Wiring*

Switchboard cables and dc and ac power cables for system loads are run in accordance with specific rules to minimize electrical interference. There are four classes of cables which are segregated in their own cable rack sections and broken out and connected in specified ways. The cable rack above each frame line-up has three separate channels on an upper level and a fourth centered below them. Address and answer bus cables are segregated in the lower channel and scanner cables in the front upper channel; both are terminated directly on transformers and terminal strips at the top of the frames. The dc power cables are segregated in the

rear upper channel and run down the frame uprights to filters and fuse panels. Tip and ring and relay control cables are in the larger central upper channel, and these run to frame or unit terminal strips or to ferreed switch terminals.

Power wiring for the appliance outlet on each frame is run in the shop. The installer then connects the wire ends at a frame junction by inserting them into the connectors in the appliance outlet on the adjacent frame.

Busways deliver power to the end frame in each line-up. Power wiring for lighting is run in busways to the fixtures over each maintenance aisle. The lamps have switches in the end guards of line-ups connected by low-voltage wiring to relays on the fixtures.

Special frame insulating practices are followed, and frames are connected to ground at only one point to avoid electrical interference from stray ground potentials and currents.

V. APPARATUS TYPES AND QUANTITIES

No. 1 ESS equipment consists to a large degree of new types of switching apparatus. To achieve system economies, a concentrated effort has been made to restrict the varieties and codes of apparatus used to an adequate but limited catalog of each apparatus type. This code concentration permits shop and field economies both in equipped frames and in spare parts.

5.1 *Number of Apparatus Codes and Their Quantities*

The number of codes (or types) of each of the more usual apparatus elements in No. 1 ESS is given in Table III, together with the quantity of each in a typical 10,000-line office.

Approximately 13 per cent of all circuit packs will be of one code and 65 per cent will be of 17 codes.

Since the typical office uses 5.5 transistors, 16 diodes, and 20 resistors per line, the unit costs for these components have a marked influence on the office cost.

5.2 *Semiconductor Devices*

Three transistor codes are used: one general-purpose type for amplifiers and switches, one power transistor type, and one pnpn switch.

Eight diode codes are used: three high-speed types to perform the bulk of the logic functions, three voltage regulator types, one level shifter, and one click reducer.

TABLE III — APPARATUS CODES AND QUANTITIES

Apparatus Elements	No. of Codes	Quantity in Typical 10,000-Line Office
Transistor	3	55,000
Diode	8	160,000
Resistor	23 types	200,000
Transformer or inductor	21 types	26,300
Capacitor	30 types	23,400
Ferreed switch (8 × 8 Type)	4	3,400
Ferreed switch (1 × 8 Type)	2	2,000
Ferrod sensor	4	8,000
Relay	78	14,000
Ferrite sheet memory	1	16
Twistor memory	1	32
Circuit pack	150	12,600

5.3 *Passive Components*

All passive components (resistors, transformers, inductors, and capacitors) for the system were selected from a master list for each class which was catalogued specifically for No. 1 ESS to limit varieties and to assure universal use of high-quality components having reasonable costs. Types were selected with the preferred construction for each desired range of electrical characteristics, and with particular attention to tolerance, physical size, life, aging stability, and failure mode (principal cause of failure at normal end of life).

5.4 *Circuit Packs*

In order to minimize the number of semiconductor circuit pack codes, it was necessary to develop several standard circuit pack arrangements of logic gates. Tests were made of a variety of wired gate packages to determine best circuit and semiconductor arrangements. Following this development work, pilot production of printed wire board designs was tested in brassboard circuits.

A family of packages evolved from this program. The semiconductors and many other small components are mounted on printed circuit packs which engage plug-in connectors in die-cast aluminum apparatus mountings. These mountings accommodate as many as 16 packs on 0.4-inch centers or correspondingly fewer on wider centers. The two packs removed from the mounting in Fig. 8 are typical for low-level logic circuits. They are made from $\frac{3}{32}$-inch thick phenol fiber of a fire-retardant grade and have components mounted on one side with printed wire interconnections on the other. The printed wire paths are gold plated at one end

of the board to form 28 connector terminals. The packs are $3\frac{3}{4}$ inches wide and $6\frac{7}{8}$ inches long, providing space for approximately 70 typical ESS components. The number of components per circuit pack varies from as few as 6 relatively large transformers to as many as 84 smaller components, including resistors, diodes, and transistors. This size of pack and number of terminals reflect experience with the Morris trial and represent a compromise among such factors as total contacts required for the system, lead lengths, cost, and number of codes. As shown in the figure, each circuit pack carries a colored label with a code number on the protective aluminum strip at the front. For proper selection, this color and number must match a similar label on the hinged strip at the top of the mounting.

Each apparatus mounting for circuit packs has this hinged designation strip across the top with designation cards, front and rear, to show for each circuit pack its position in the mounting, its apparatus code, and its color code. Circuit packs are physically, although not electrically, interchangeable.

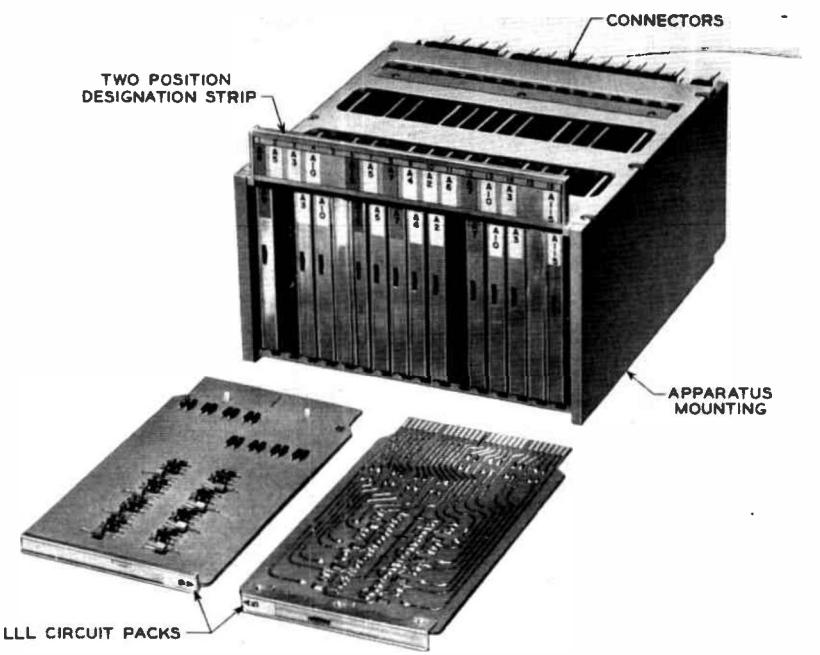


Fig. 8 — Circuit packs and apparatus mounting.

Each circuit pack bears one of three color codes: red, yellow or blue. No damage will be done if the wrong circuit pack is plugged into a connector designated with the same yellow or blue color code, although the circuit will not function properly if the codes do not match. The red packs must be plugged into connectors having the same code and color designation.

5.5 *Ferreed Switches*

The ferreed switch was designed to serve as the crosspoint element for No. 1 ESS networks. It is a two-wire, magnetically latched, pulse-operated device. It consists of two small sealed reed switches which are operated and released by controlling the magnetization of two adjacent rectangular Remendur plates. Remendur, an alloy of iron, cobalt and vanadium, is a magnetic material with a square hysteresis loop. Each plate is magnetically divided into two independent halves by a low carbon steel shunt plate, which also provides the mechanical structure for assembling the crosspoints into various arrays of 8 × 8 switches. When the two halves of each plate are magnetized series-aiding, the flux from both plates returns through the sealed reeds, causing contact closure. When the two plate halves are magnetized in series-opposition, the return flux through the reeds' gap is reduced to practically zero and the contacts open.

An individual two-wire crosspoint is shown in Fig. 9 and an 8 × 8 array of these crosspoints, known as a ferreed switch, is shown in Fig. 10. The phenolic forms on which the coils are wound are molded directly into the shunt plate. The windings are wound on these forms by automatic machines which provide a checkerboard pattern; that is, adjacent coils are wound in opposite directions to reduce magnetic interference. Since the contact gap in the sealed reeds must be accurately located relative to the shunt plate, both reeds of a crosspoint are carefully positioned in a molded contact assembly which, when inserted in the coil form, guarantees the desired tolerances. This contact assembly also provides for properly positioning the Remendur plates within the coil form. Terminals are provided on the front and rear of the switch so that all shop wiring can be on the rear of the frame and all installer wiring on the front. This allows the installer to work in the wider equipment aisles without interfering with the shop wiring.

Four varieties of these two-wire switches have been coded for the network. The first provides one 8 × 8 array, the second two 8 × 4 arrays, the third 16 × 4 of 8 array (which gives 16 lines access to 8 links but each line access to only 4 of these 8), and the fourth four 4 × 4 arrays.

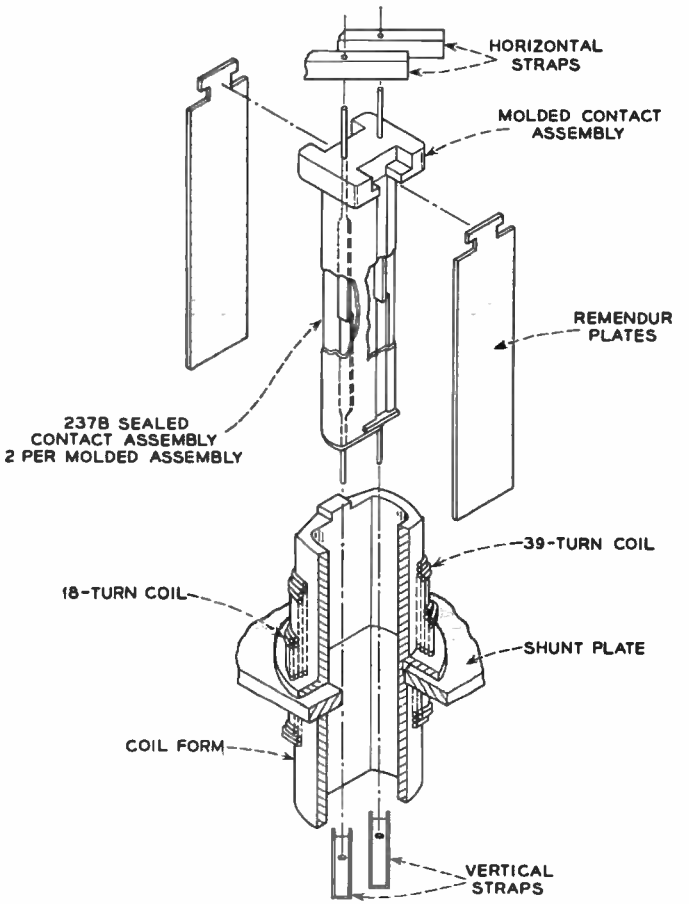


Fig. 9 — Two-wire ferreed assembly.

Another version of the ferreed, used for cutoff and test access, has two sealed reed switches mounted between two rods. One of these is permanently magnetized, and the other, a Remendur rod, has its polarity controlled by a winding, as shown in Fig. 11. Eight of these bipolar crosspoints are assembled into one coded switch, shown in Fig. 12. Two varieties of these switches are used The first has both ends of all crosspoints brought out to terminals, thus providing eight individual crosspoints in one package. The second has one side of all crosspoints strapped together to provide a 1×8 switch.

A summary of the characteristics of these ferreed switches is shown in Table IV.

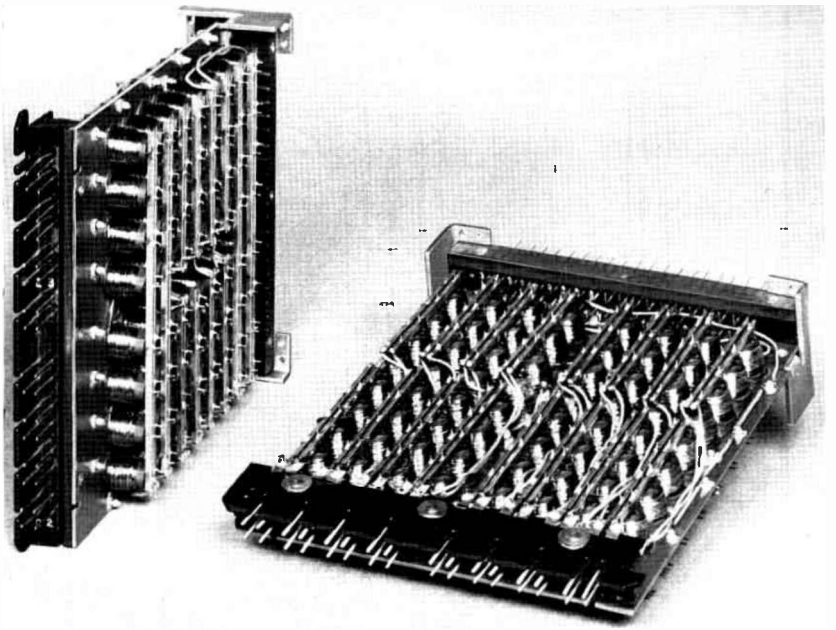


Fig. 10 — 8 × 8 ferreed switch.

5.6 *Ferrod Sensors*

The ferrod sensor, a current-sensing device, has been developed as the building block for all No. 1 ESS scanners. It consists of a ferrite stick located on the centerline of a pair of identically wound solenoidal coils. In the ferrite stick are two holes through which are threaded two single-turn loops of wire, one carrying the interrogate pulse and the other the readout pulse. Coupling between the two loops depends on the magnetic state of the material around the holes, which in turn depends on the amount of dc flowing in the solenoidal coils. Thus, with no current in the control winding (on-hook or open circuit) an interrogate pulse produces a large pulse in the readout loop, whereas presence of dc in the control winding suppresses this pulse.

To conserve frame mounting space, the ferrod sensor unit contains two ferrods, one behind the other. These units are mounted on equipment frames in egg-crate apparatus mountings (see Fig. 13), each of which accommodates 128 of these dual units or 256 ferrod sensors. The mounting serves not only as a physical support but also as an array of magnetic shields to prevent interference between adjacent sensors.

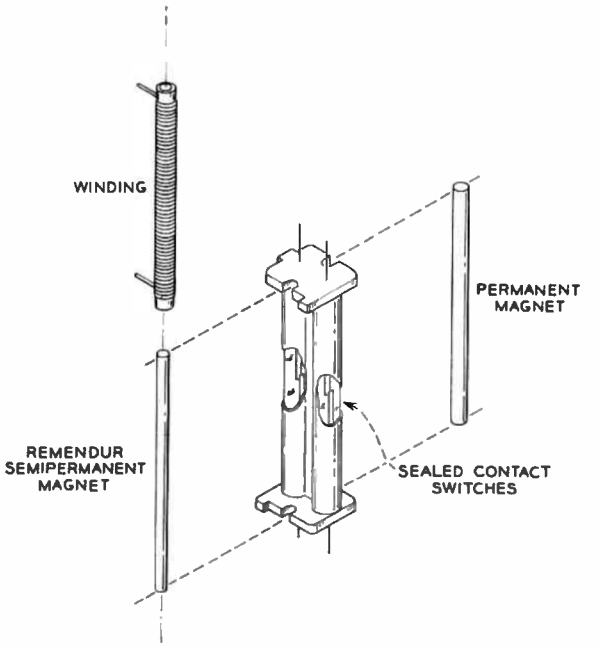


Fig. 11 — Bipolar ferreed assembly.

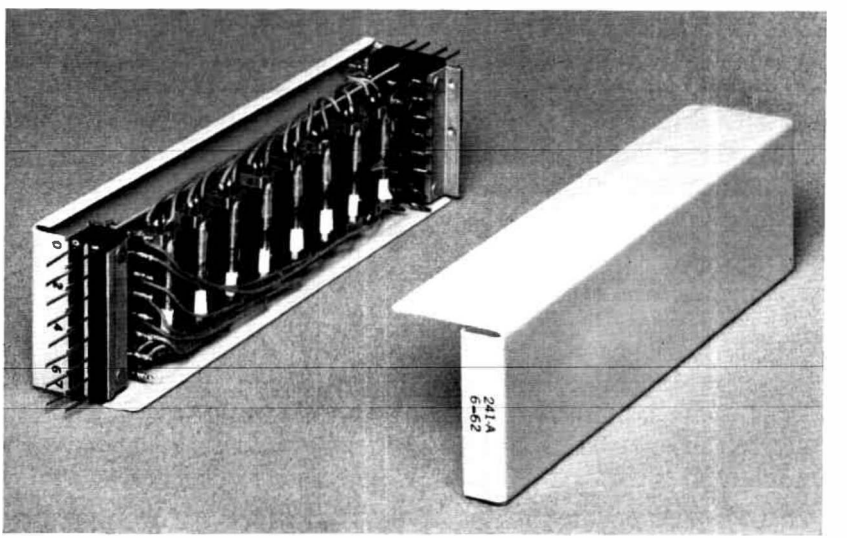


Fig. 12 — Bipolar 1 × 8 ferreed switch.

TABLE IV — SUMMARY OF FERREED SWITCH CHARACTERISTICS

	Over-All Dimensions			No. 1 ESS Network Circuit Requirements									
				Operate and Release Pulse			Characteristics						
	Height (inches)	Width (inches)	Length (inches)	Peak Amperes	25% width (microseconds)	Energy (joules)	Resistance (milliohms)	Operate chatter (milliseconds)	Breakdown (volts)	Dry operate (milliamperes)	Wet operate (milliamperes)	Expected life (operations)	Tolerable surge (amperes)
2-wire (8 × 8 type)	$6\frac{3}{4}$	$2\frac{1}{8}$	$9\frac{1}{4}$	9	200 to 500	0.2	<200	<3	>800	125	0	10^6	3.6
2-wire bipolar (1 × 8 type)	$1\frac{5}{8}$	$2\frac{1}{8}$	$9\frac{1}{4}$	6	200 to 500	0.01	<200	<3	>800	125	40	10^6	3.6

The ferrod sensor units take full advantage of the economies of molding. Both control windings are wound directly on a molded spool which supports the ferrite stick and, in the case of the more sensitive 1D and 1E sensors, the metallic magnetic return path. Two of these spool assemblies are located between two similar molded wire arrays which hold the spools in place and establish contact between the spool terminals and the molded wires, the ends of which serve as the apparatus terminals. In the 1B code, two contact protection networks are also supported between these molded wire arrays. These networks provide protection for the cutoff contacts, which are in series with the control windings and reduce interference in the connecting circuits as the cutoff contacts operate. Both of the sensors in the 1B unit are shop wired for loop start lines, but one of them can readily be converted for ground start operation by changing strapped connections on the front terminals of the unit.

5.7 *Ferrite Sheet Memory*

The 6A ferrite sheet memory is used in the call store for temporary, electrically-changeable information.[5] The memory module, shown in Fig. 14, contains 192 active ferrite sheets, plus 12 spare sheets, arranged in three columns and divided into four sections or submodules. An individual sheet, shown in the inset, is 1.04 inches square and 0.030 inch thick, with 256 holes of 0.025-inch diameter arranged in a 16 × 16 array. The material surrounding each hole acts as a small, two-state magnetic core to store one information bit. Thus the module has a capacity of

49,152 bits, which are organized by the wiring pattern into 2048 words of 24 bits each.

Four one-turn "windings" pass through each hole for X-address, Y-address, inhibit, and readout functions. The Y-address windings are formed by the conductors "printed" on the two sides of the sheets and through the holes by evaporation and plating techniques. Interconnections between adjacent sheets, and between sheets and the connectors at the sides, are by short wire jumpers soldered to the two large land areas on each sheet. The X-address, inhibit, and readout windings are made by threading fine 36-gauge wire through each column of holes. The readout windings are applied on a submodule basis so that each threaded length is the height of the submodule. Connections are made to the terminals printed on the sides of the epoxy-paper printed wire board at the top of each submodule. The X-address and inhibit windings are threaded through the entire height of the module, with X windings terminated on the connectors in the upper right corner and inhibit windings on terminals of the printed wire board at the bottom of the module.

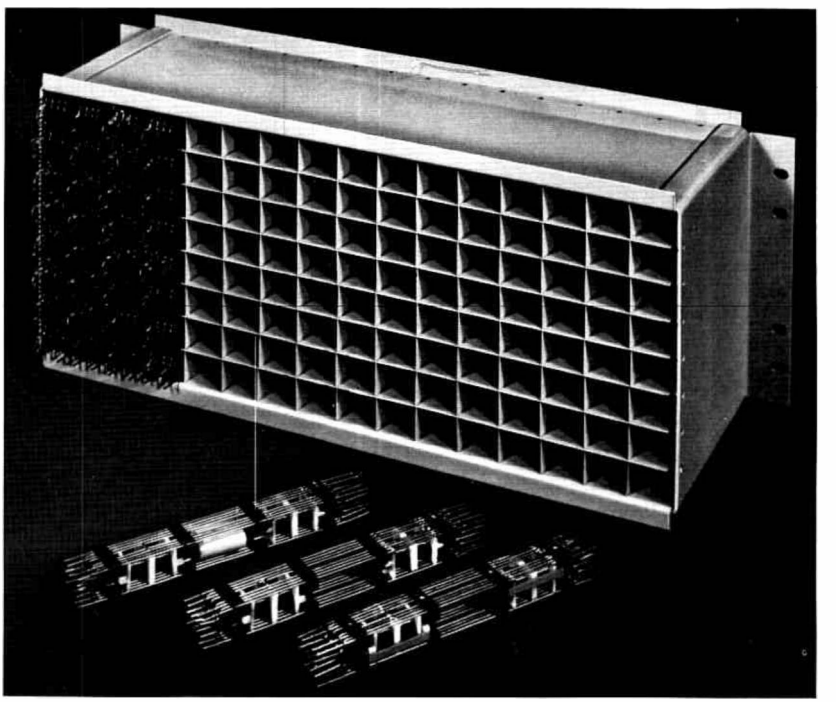


Fig. 13 — Ferrod sensors and mounting.

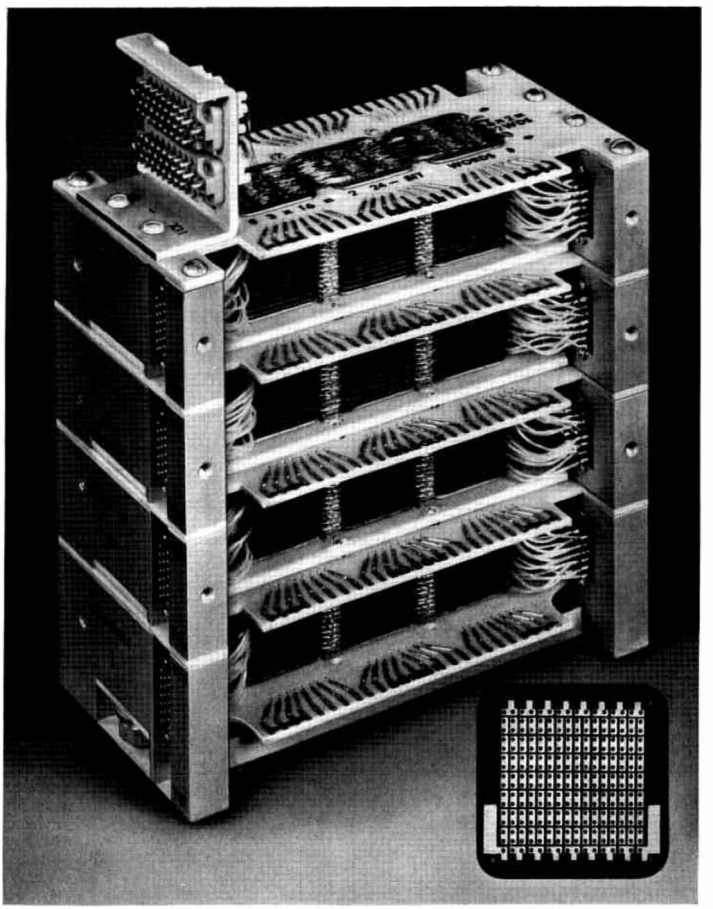


Fig. 14 — Ferrite sheet memory and sheet.

The use of pluggable connectors for all winding terminations facilitates testing and permits a compact assembly of the four modules needed for each call store by plugging the modules together with special, double-ended printed wire board connectors between. This minimizes noise pickup in the readout windings and eases field replacement.

5.8 *Twistor Memory*

The 1A twistor memory is used in the No. 1 ESS program store for bulk storage of the semipermanent program and translation information.[6,7] The memory module, shown in Fig. 15, is composed of 64 twistor

planes mounted vertically in an aluminum and steel framework. Each plane is made of stable glass-bonded mica, with solenoid tapes and twistor tapes cemented to each side. Each solenoid tape carries 64 strands of flattened wire, corresponding to 64 memory words, and each wire links a magnetic access core along the rear edge of the plane. The twistor tapes contain separate twistor wires, together with return paths, for simultaneous readout of each of the 44 bits per memory word. Two such tapes, designated A and B, are used for the entire module, and each is cemented to its side of each twistor plane. The two ends of tape A are terminated on the front side of the module, on the lower halves of the terminal fields, while tape B is terminated on the upper halves. The 64 × 64 field of access cores at the rear is linked by X- and Y-select windings and a bias winding, which are terminated on the rear side on terminal fields at both ends and along the top and bottom edges of the module.

Resilient flat springs between the twistor planes maintain intimate contact between 128 twistor memory cards and the twistor tapes. The memory card (see Fig. 16) is an aluminum sheet 0.016 inch thick with 64 rows of tiny permanent magnets, each representing a bit of information

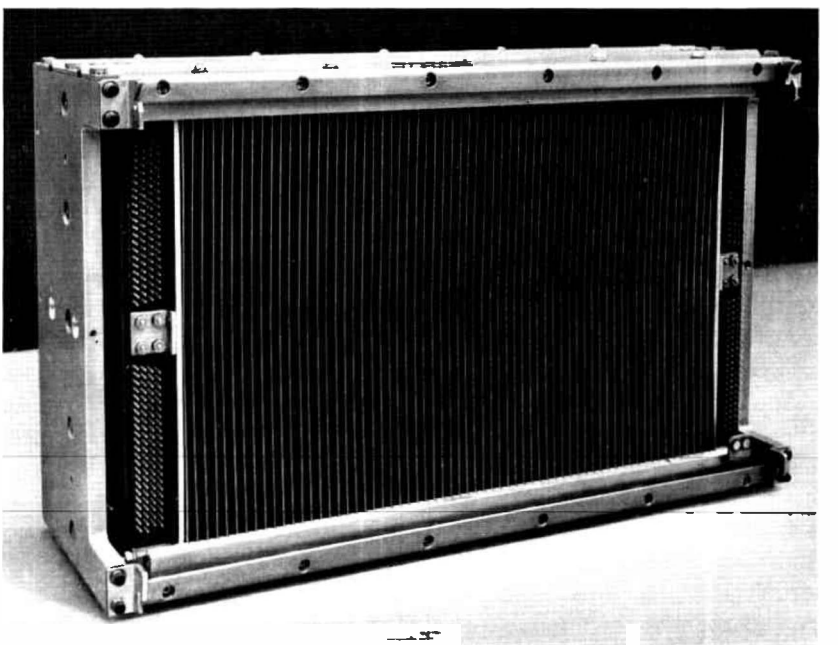


Fig. 15 — Twistor memory module.

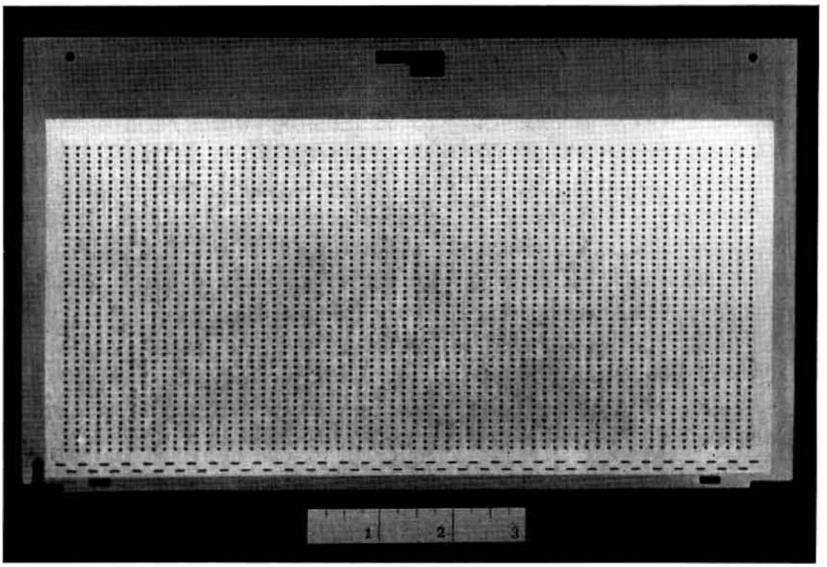


Fig. 16 — Memory card.

to be remembered. Thus the module has a capacity of 8192 words of 44 bits each, or 360,448 bits in all. A magnetized state is read as a binary "zero," while demagnetization corresponds to a "one." Two other rows of magnets along one edge are used for initializing, or controlling, the magnetic state of the twistor wires between word locations, and for sensing the word locations during the card writing process. The card is positioned vertically by a locating notch near one corner, which engages a preadjusted pin in the module. The depth of insertion is individually controlled for each card by two factory-adjusted screws. This maintains accurate registration for the bit magnets with respect to the intersections between the solenoid and twistor wires.

The memory cards are inserted into or withdrawn from the module simultaneously by the card loader. The keyhole-shaped opening near one edge of the card permits the loader to grip the card during insertion and withdrawal, while the rectangular punchings near the other edge are engaged by fingers of the card writing unit to draw the cards individually from the loader during the writing process. Other features to facilitate card loading and writing include: (a) special locating ears at the four front corners of the module for support and accurate positioning of the loader with respect to the twistor planes, (b) a connector in the lower right corner to supply 48-volt dc power to the loader, (c) tapered guiding

surfaces to facilitate entry of the cards into the module, (d) accurate dimensional requirements for the alignment and spacing of the twistor planes, and (e) special requirements for card flatness and the force needed to seat the cards in the memory module.

5.9 *Relays*

For operations of relays, such as those performing supervisory functions in the trunk circuits where electronic speeds are not essential, a family of magnetically latching wire spring relays (types AL and AM) was developed. Through the use of a new magnetic structure, each of these relays operates when driven with a −48-volt pulse and remains operated. The relay is released by a controlled +24-volt pulse (sufficient to release the relay while avoiding reverse polarization and resultant reoperation).

Some semiconductor circuits have loads requiring metallic paths. The signal distributor uses a mercury relay driven by a flip-flop to operate the multicontact relays forming the trunk relay selection trees. The program store uses a circuit pack containing eight dry reed relays operated by low-level logic (LLL) circuits. The line switching frame uses two types of circuit packs containing six dry reed relays operated by a four-bit register flip-flop. In both cases these relay contacts operate wire spring relays.

5.10 *Transformers and Terminal Strips*

The 0.5-microsecond interframe bus system[8] has semiconductor drivers in the central control, with receivers in as many as 60 peripheral frames distributed along a pair extending 450 feet. The two transformer single-turn input windings have a ground shield to minimize noise coupling.

The ferrite cores are supported in a dual molded wire comb array as shown (see Fig. 4) with a wire braid shield over the input windings. An array of twelve ladders is assembled to provide for a 24-bit bus. The depth of the transformer matches the size of the semiconductor driver and receiver circuit packs with which it is used. A similar terminal strip provides for those leads which do not pass through a transformer.

5.11 *Connectors*

5.11.1 *Circuit Pack Connector*

The circuit pack connector[9] (see Fig. 17) is essentially a two-piece unit. Twenty-eight 0.036-inch diameter phosphor bronze springs are molded

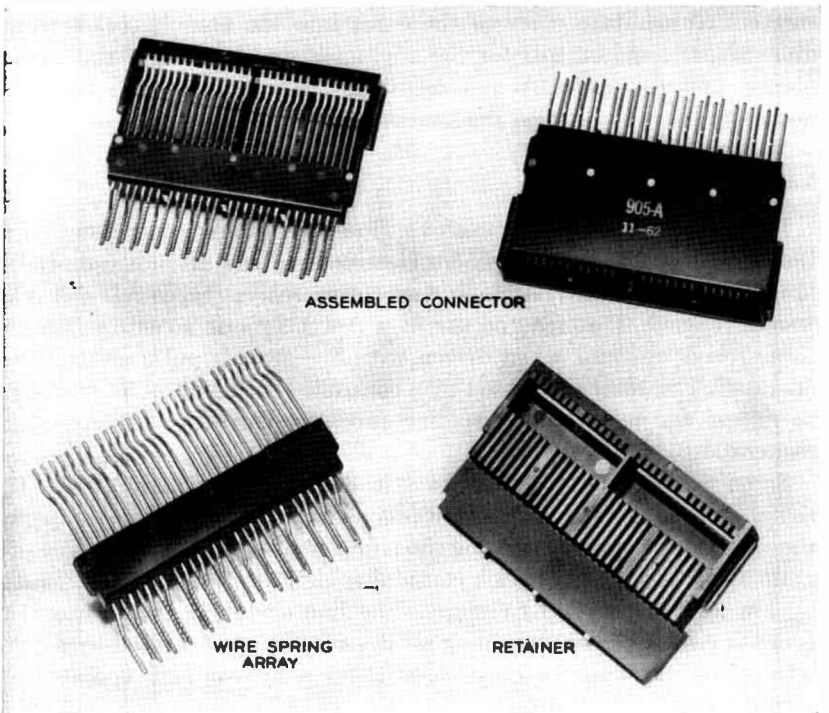


Fig. 17 — Circuit pack connector.

into a phenolic block in much the same manner as the wire spring relay. This pretensioned spring assembly is riveted to a phenolic molded retainer.

A gold overlay contact button is welded to the spring to assure a low-resistance contact. The other end of the spring is serrated for wire wrapping. The contact is designed for 500 insertions and withdrawals of circuit packs.

With the circuit pack in place, the minimum contact force will be 155 grams.

5.11.2 *Junctor Grouping Frame Connector*

The connector and mating plug of the junctor grouping frame are used to interconnect cables with 16 pairs. The contacts are formed from 0.036-inch diameter phosphor bronze wire. Two sets of sixteen contacts, each formed as spring members, are used in the connector, and two sets

of flattened contacts are used in the plug. Each contact spring has a nominal force of 250 grams on the mating plug contact. All contacts are gold plated to insure a minimum life of fifty insertions.

5.12 *Distributing Frame Connecting Blocks*

Two new connecting blocks, similar to that shown in Fig. 18, have been developed for No. 1 ESS distributing frames. Both employ terminals of a new design adapted from that in 66-type connecting blocks, which are finding wide use in station systems.[10] The "quick-connect" feature speeds up the making of cross connections, because all that is necessary is to hold a plastic-insulated wire in the slot opening of the terminal clip (with about $\frac{1}{8}$ inch extending beyond) and force it into place with a hand tool. Thus, without prior preparation, a thrust of the hand causes the terminal to cut through the insulation and com-

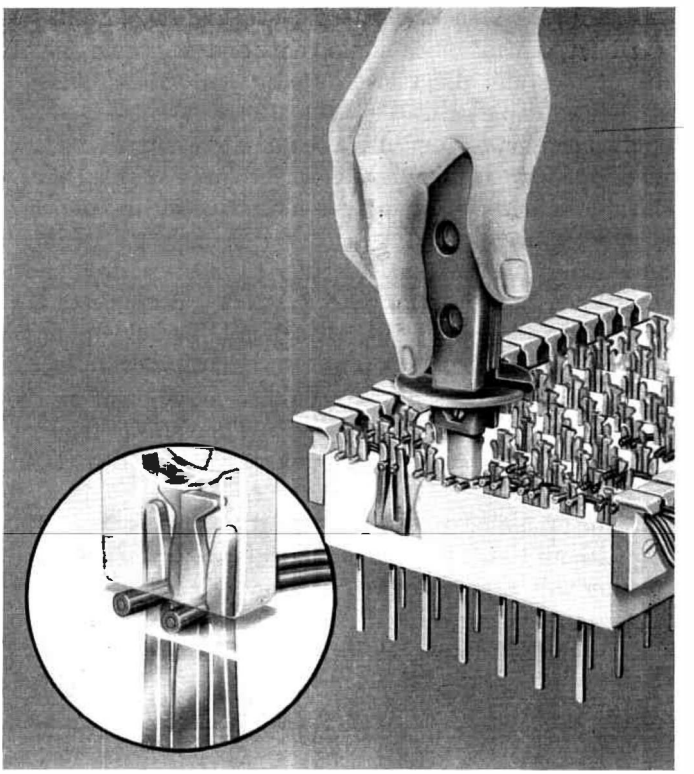


Fig. 18 — Distributing frame connecting blocks and insertion tool.

plete a good connection. A simple forked tool is used to remove a lead without disturbing wires on adjacent terminals.

Each connecting block has 64 pairs of terminals, but one has two slots per terminal to accommodate two jumper wires, while the other has only one slot. Both terminals accept two solderless wrapped connections on the rear. Both blocks are molded from white plastic. The terminals are made from 0.045 by 0.045 inch phosphor bronze.

5.13 *Protector Apparatus*

A new connector, coded 302, terminates 100 outside plant pairs on the protector frame. This new unit is responsible for a 3.3:1 reduction in frame length while providing all of the usual features for protection, isolation, and testing of lines.

The connector panel is a molded, flame-retardant, plastic unit 16.2 inches long and 4.3 inches wide with 100 jacks arranged in four rows of 25 each (see Fig. 19). Each jack has five pins: tip and ring (T&R) for outside plant, T&R for central office equipment, and a ground pin. Each individual protector unit plugs into a jack to provide standard 500-volt protection for one cable pair.

Gold-plated contacts arranged in two 50-pair groups at top and bottom give front access for attaching test-shoes for various cable tests.

A stub cable is factory-terminated on the rear of the connector, using either 24- or 22-gauge conductors.

An individual protector unit (see Fig. 19 inset) is provided for each cable pair. When fully inserted into the panel, it connects the central office equipment to outside plant through gold-plated contacts. A cable pair is disconnected by pulling a protector unit forward to a detent position, which disconnects the central office equipment without disconnecting protection on the outside pair. This feature facilitates office installation, cutover, cable testing, temporary service denials, and other services.

The 3A-type protector units have no heat coils, since No. 1 ESS is self-protecting without them. Certain circuits of other systems which are not self-protecting will require heat coils. They will use 4A-type units. No protection is required for cable systems that are not exposed to foreign potentials. Here, the dummy 5A type will be used.

VI. EQUIPMENT FRAME DESIGNS

6.1 *Network*

The No. 1 ESS network is a space-division network in which two-wire metallic connections are switched through eight stages of ferreed switches.

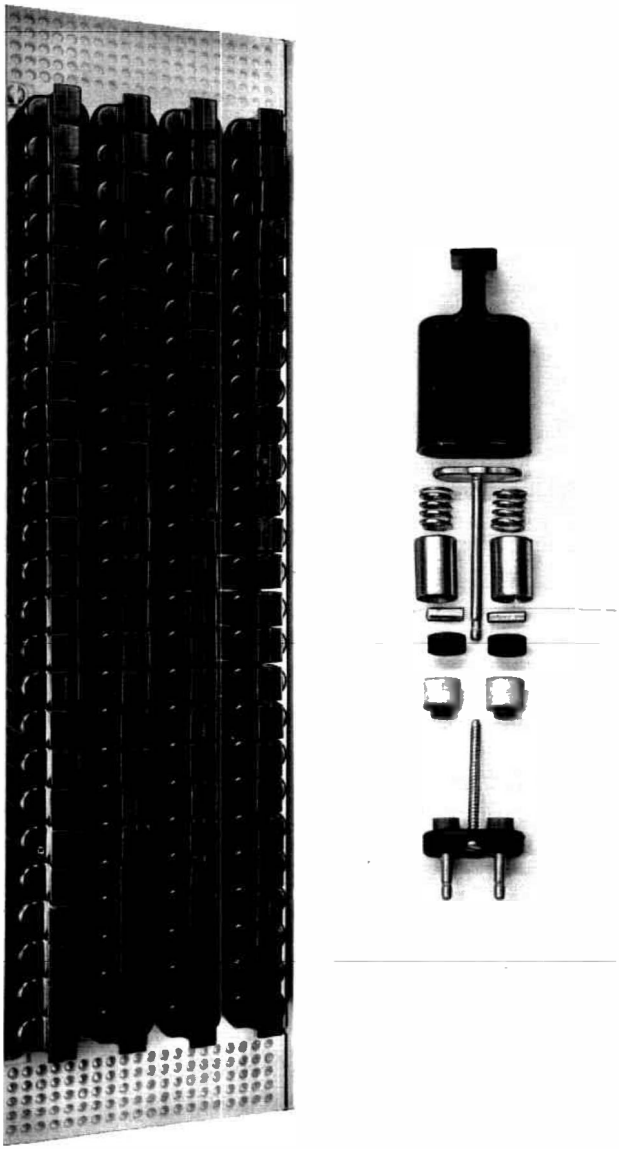


Fig. 19 — Protector and connector.

In addition to the normal interconnections of lines to lines, lines to trunks, and trunks to trunks, the network is used to interconnect lines and trunks to service circuits, such as: tone trunks, signal transmitters and receivers, coin supervisory circuits, ringing circuits, party test circuits, and maintenance circuits. To establish these interconnections, central control selects the desired network paths and, via the central pulse distributor and peripheral bus, addresses the network frames. The network in turn translates and executes these orders to establish the specified paths.

No. 1 ESS offices are arranged to work with a maximum of 16 line link and 16 trunk link networks. With 4:1 concentration, this provides network terminations for 65,536 lines and 16,384 trunks.

The line link network consists of line switching and junctor switching frames, while the trunk link network is comprised of trunk switching and junctor switching frames. Each of these frames provides for two stages of switching. The interconnection of these frames is shown in Fig. 1.

6.1.1 *Line Link Network*

There are two general types of line switching frames in this network; one provides 2:1 concentration and the other 4:1. Two frames are coded for each: one a home frame and the other its mate, which contains considerably less control equipment for line scanner and network.

In both the 2:1 and 4:1 frames, duplicated line scanner controllers are provided in the home frame. These two controllers alternate in serving a pair of home and mate frames, each one in turn controlling the scanner matrices on both frames for a fixed period of time under control of the program. If the active controller fails, the standby assumes control.

The duplicated network controllers in the 4:1 home frame serve a pair of home and mate frames. One controller normally serves the switches on the home frame and the other those on the mate frame, with either one taking over control of both frames if a failure occurs in the other. Duplicated network controllers are required on both home and mate frames, in the 2:1 frames due to the high calling rate there. Each controller normally sets up connections to only one half the switches on its frame, but either will control both halves of a frame under trouble conditions.

In addition to these control equipments, each home and mate 2:1-type line switching frame contains its half of a 1024-line ferrod sensor matrix for detecting service requests, and sixteen line concentrators with 32 lines each. Each concentrator is made up of two first-stage switches, each having four 4 × 4 crosspoint arrays, and two second-stage switches, each

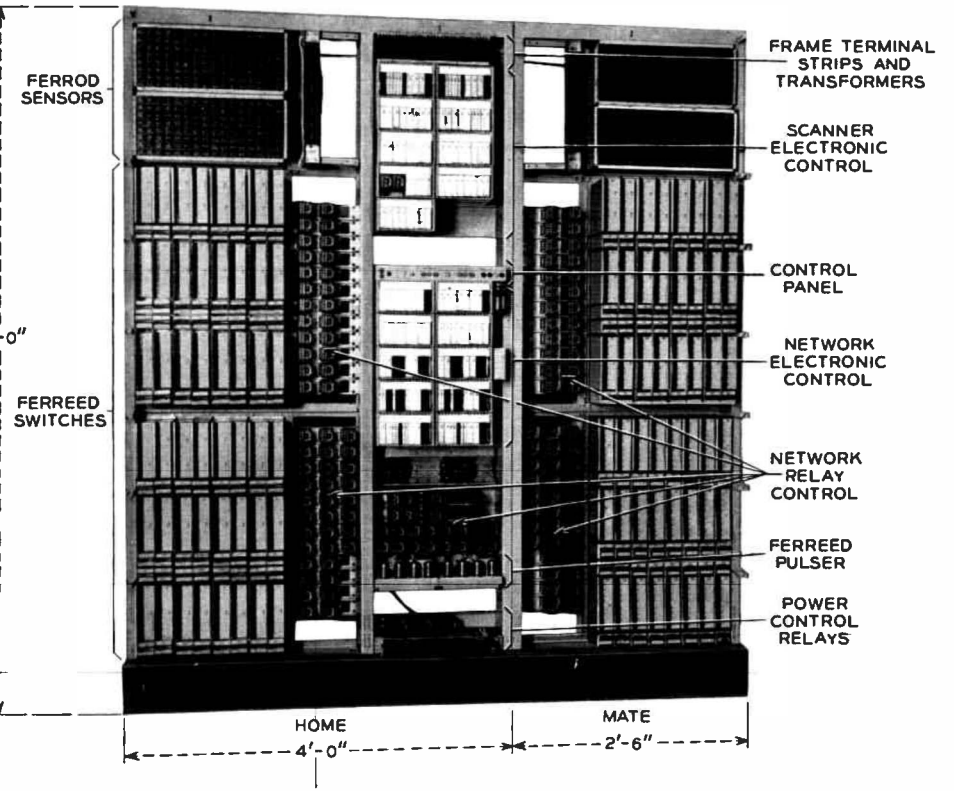


Fig. 20 — Line switching frames for 4-to-1 concentration ratio.

having two 8 × 4 arrays, which provide the 2:1 concentration. Each concentrator also has four bipolar ferreed switches, each with eight two-wire crosspoints for cutting off the ferrod sensors.

Each of the 4:1 home and mate frames contains its half of a 1024-line ferrod sensor matrix and eight line concentrators with 64 lines each, in addition to the above control equipments. Each concentrator has four first-stage 16 × 4 of 8 switches which provide 2:1 concentration and two second-stage switches, each with two 8 × 4 arrays, which provide an additional 2:1 concentration for a combined concentration of 4:1. Each concentrator also contains eight of the above bipolar switches. The four line switching frames are shown in Figs. 20 and 21.

The line junctor switching frame shown in Fig. 22 contains duplicated network control equipment and four grids, each with eight third-stage and eight fourth-stage 8 × 8 switches and eight 1 × 8 bipolar switches

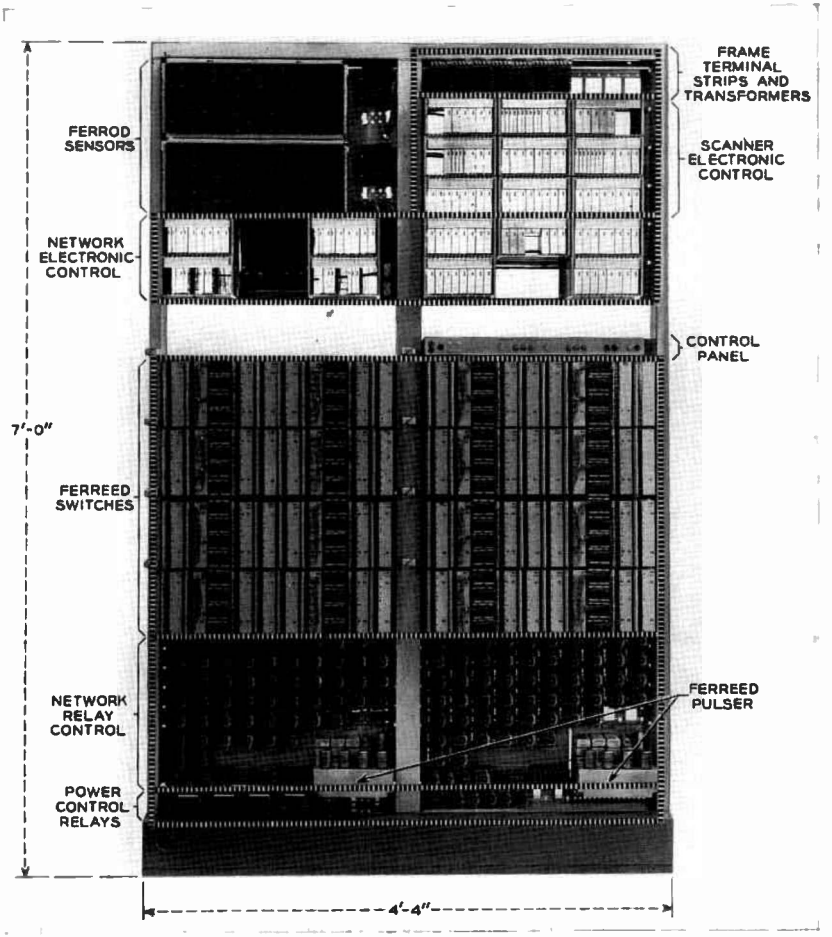


Fig. 21 — Line switching frames for 2-to-1 concentration ratio.

for test access into established connections. This frame has 256 "B" links on its third-stage and 256 junctors on its fourth-stage switches.

Each fully equipped line link network has four junctor switching frames and from four to sixteen line switching frames, counting home and mate frames as separate frames. This provides for concentrating lines ranging in number from 2048 to 8192 (in increments of 512 lines) on 1024 junctors. The 2:1 type of frame is used for concentration ratios of 2:1, 2½:1, 3:1 and 3½:1 and the 4:1 type for ratios of 4:1, 5:1, 6:1, 7:1, and 8:1.

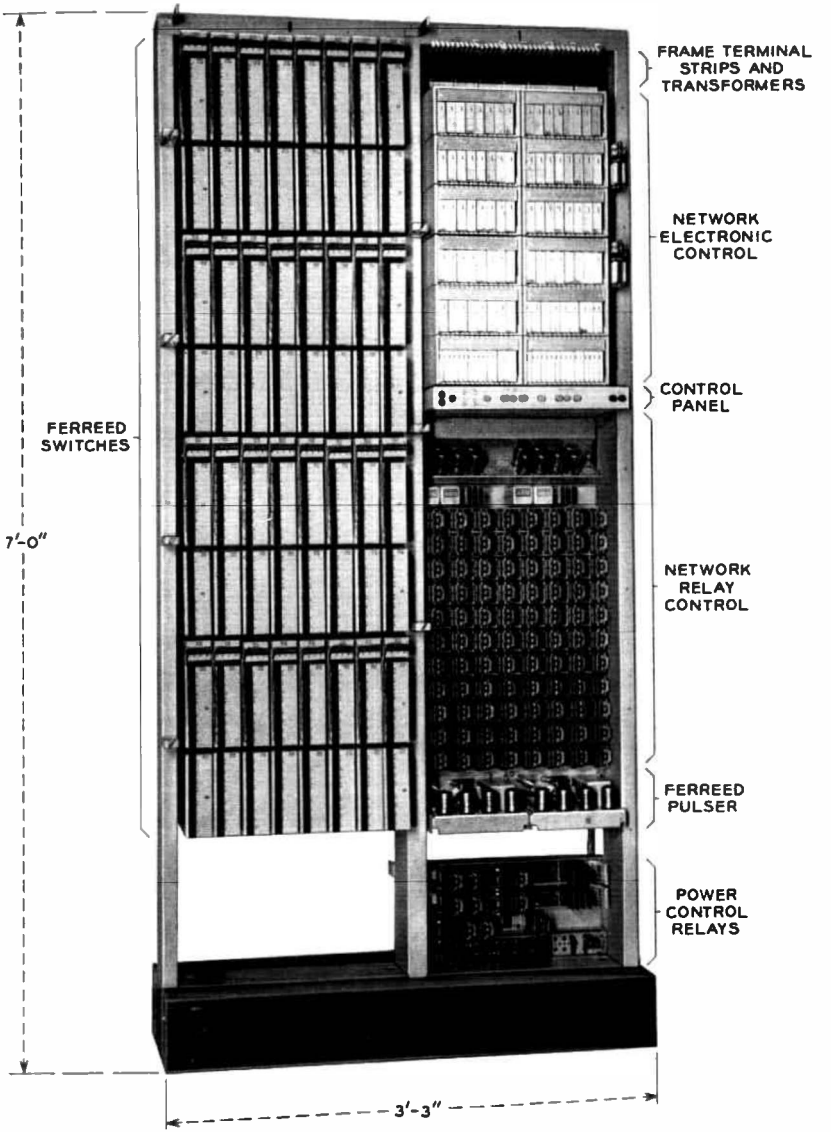


Fig. 22 — Line or trunk junctor switching frame.

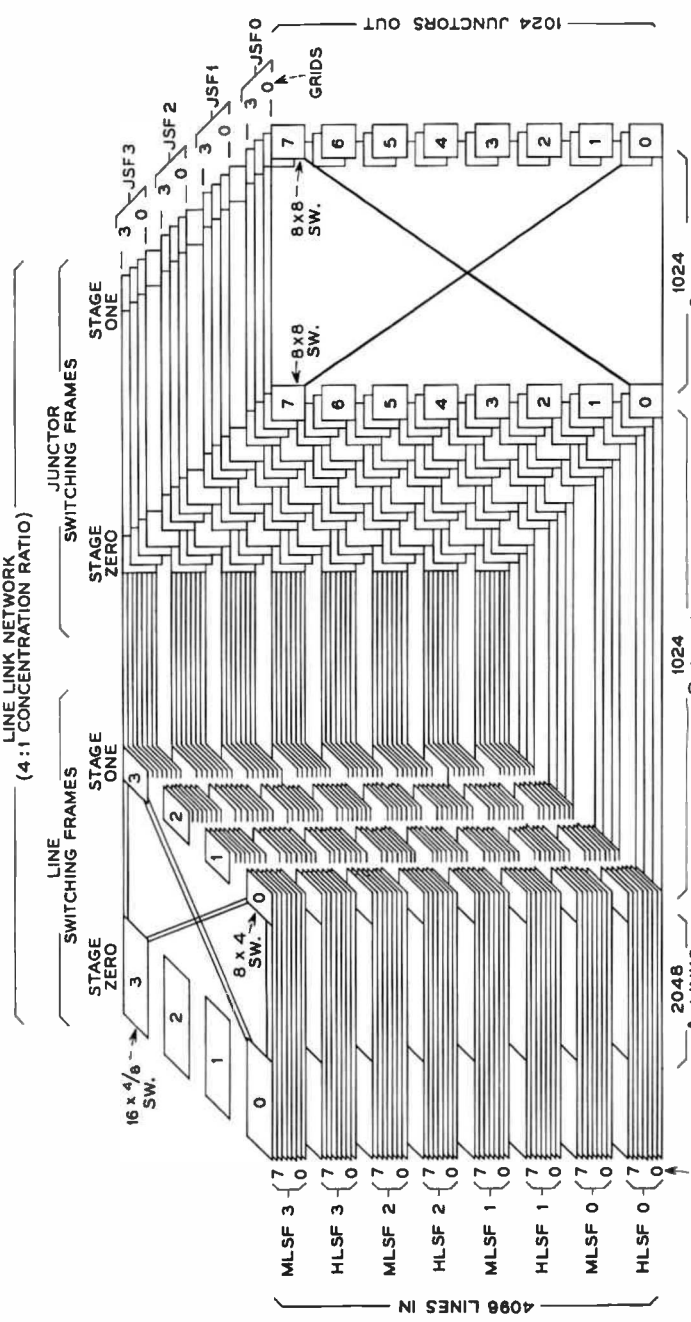


Fig. 23 — Link wiring pattern for 4-to-1 concentration ratio.

All lines on a line link network have access to all junctors via the A, B, and C links, which are spread between the first and second, second and third, and third and fourth stage switches, respectively. The wiring pattern for these links for the 4:1 concentration ratio is shown in Fig. 23.

Since all networks have a maximum of 1024 B links, regardless of their size, all networks other than the 2048-line 2:1 type and the 4096-line 4:1 type require the multipling of B links. Patterns have been established for these multiples to minimize blocking.

All network frames are shop-wired in the conventional manner, using unit surface wiring and frame local cables. However, since these frames are pulse operated, extreme care had to be exercised in the location of apparatus to minimize lead length and in the separation of leads into several local forms to minimize interference.

6.1.2 *Trunk Link Network*

The trunk link network has four junctor switching frames (the same as those in the line link network) and from four to eight trunk switching frames, which are the same as the junctor switching frames except for the omission of the bipolar ferreed switches. (See Fig. 24.)

The trunk switching frame has a capacity of 256 trunks on its first-stage and 256 B links on its second-stage switches.

Trunks ranging in number from 1024 to 2048 in 256 trunk increments are concentrated on 1024 junctors to give trunk concentrations of 1:1, $1\frac{1}{4}$:1, $1\frac{1}{2}$:1, $1\frac{3}{4}$:1, or 2:1. All trunks have access to all junctors, so B links must be multipled for all networks other than the 1024-trunk, 1:1 ratio size in accordance with patterns which minimize blocking.

6.2 *Trunk and Junctor Frames*

6.2.1 *Trunk Frames*

There are two types of trunk frames, the universal trunk frame (see Fig. 25) and the miscellaneous trunk frame. The universal trunk frame consists of a basic 4-foot, 4-inch double-bay frame and a supplementary 2-foot, 2-inch single-bay frame, each of which accommodates 64 plug-in trunk units. The miscellaneous trunk frame is a single-bay 2-foot, 2-inch frame used for mounting the conventional type of wired-in trunk circuits that come on 2-inch and 4-inch mounting plates.

The universal trunk frames, as their name implies, are universally wired so that any plug-in trunk unit may be plugged into any trunk position; each unit may have one circuit or two circuits. The supple-

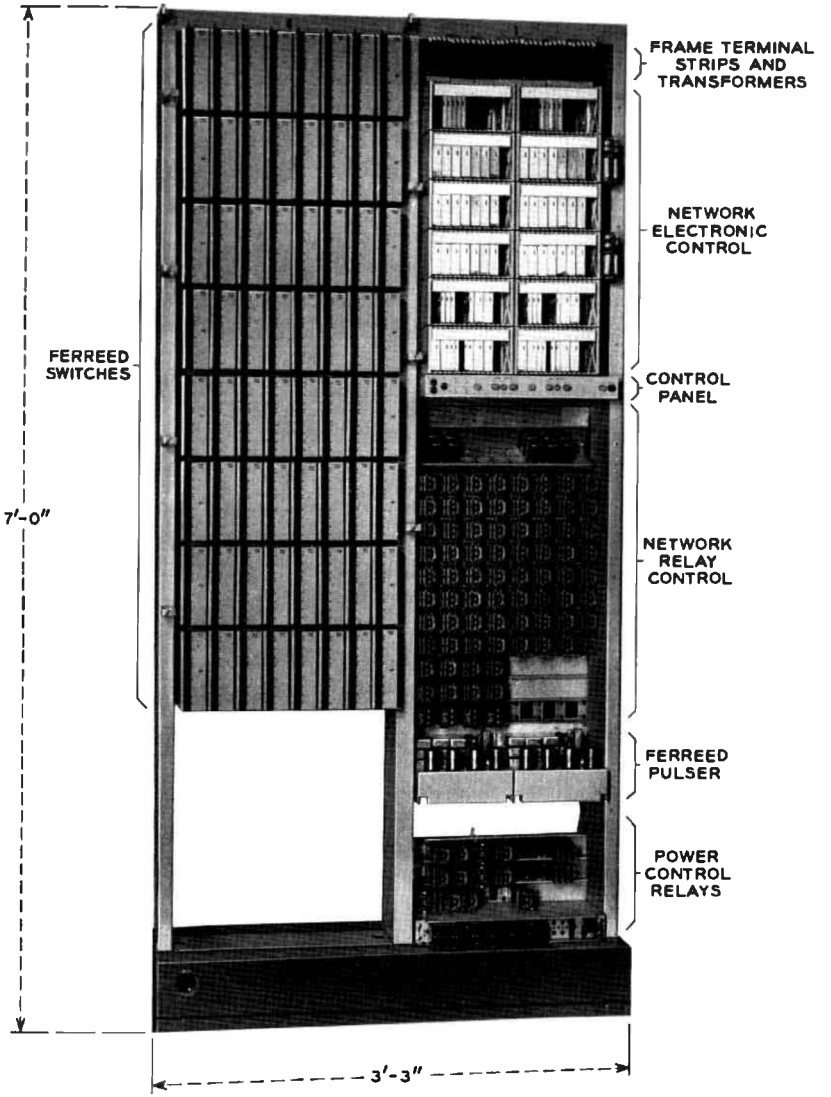


Fig. 24 — Trunk switching frame.

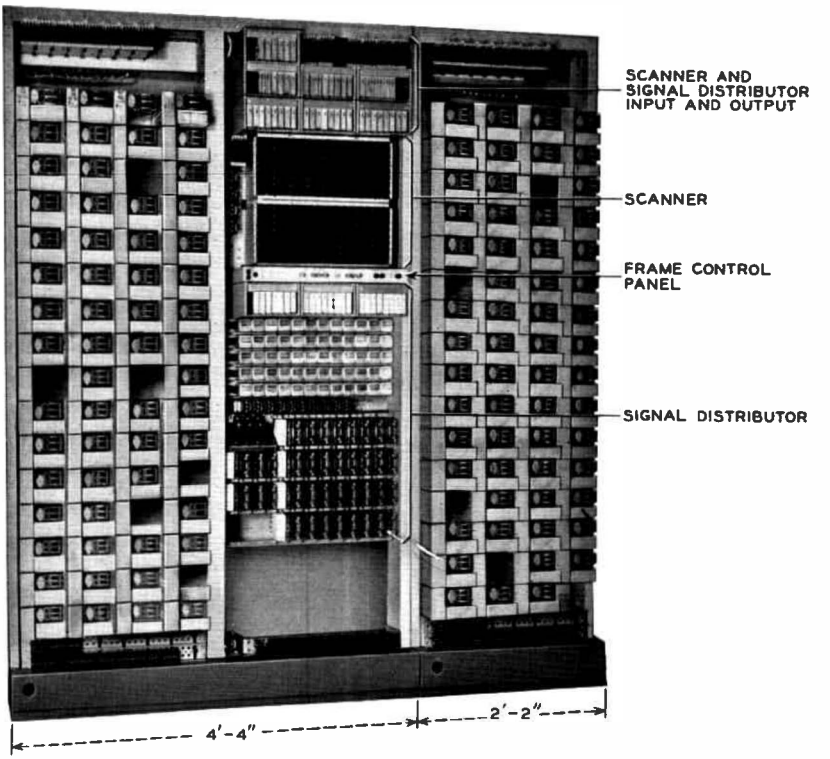


Fig. 25 — Universal trunk frame (or junctor frame).

mentary frame is located to the right of the basic frame so that the control bay, containing the signal distributor and scanner control, falls between the two trunk bays it serves. To save on control equipment, the universal trunk frames work on a home and mate frame basis. The home frame has a 1024-point scanner control unit which operates one half of a 1024-point scanner matrix on each of the home and mate frames. The scanner and signal distributor control equipments are duplicated for reliability.

The miscellaneous trunk frames contain such a variety of trunk and service circuits that it is uneconomical to provide them with universal scanners and signal distributors to satisfy all conditions. Instead, the scanning function for these trunks is performed by the master scanner, and the signal-distributing function is performed by the signal distributors on nearby universal trunk and junctor frames, each of which has 256

of its signal distributor points wired to frame terminal strips for this purpose.

6.2.2 *Junctor Frame*

The junctor frame is similar to the universal trunk frame except that it is wired for plug-in junctors instead of trunks.

6.2.3 *Trunk and Junctor Units*

6.2.3.1 *Plug-in Trunks and Junctors.* Since most trunk and junctor functions are performed by common control equipment such as central control scanners and signal distributors, the size and complexity of these trunk and junctor circuits is greatly reduced. Most high-runner incoming and outgoing trunks are simple circuits containing two or three relays.

This reduction permitted the development of the family of small plug-in trunk and junctor units. The simple angle-type sheet metal chassis shown in Fig. 26 mounts the transmission components, magnetic

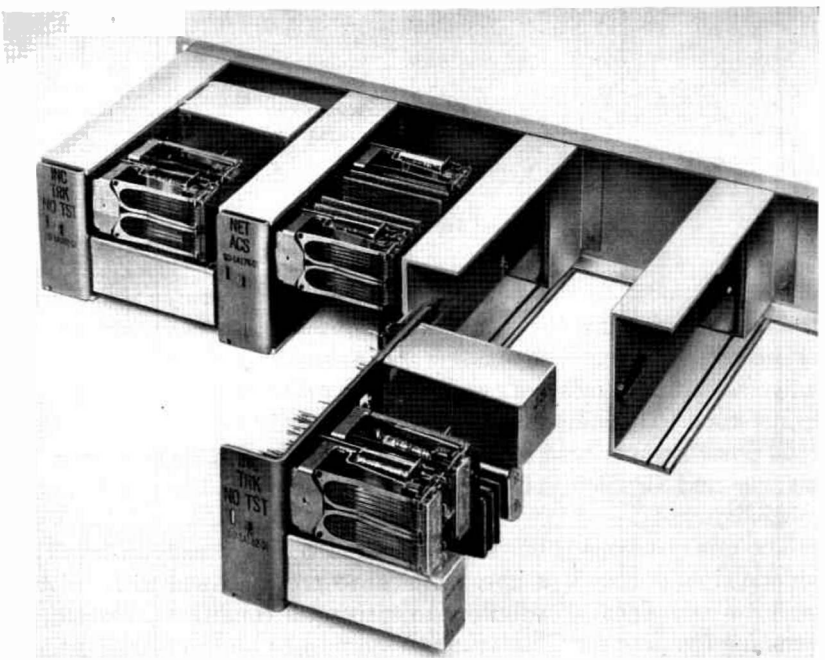


Fig. 26 — Plug-in trunk units.

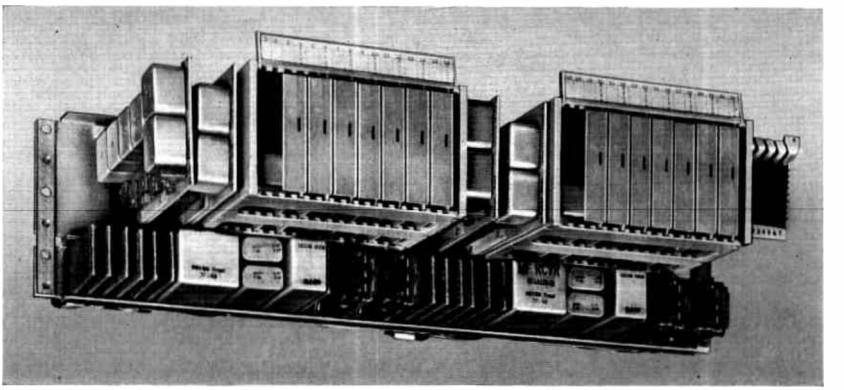


Fig. 27 — MF receiver unit.

latching relays, and a printed board connector. All units are surface wired using wire wrapped connections. Each plug-in chassis can be used for one or two (identical) trunk circuits, and all have a preassigned terminal pattern to insure compatibility with the universal frame wiring. In a typical office, all junctor circuits and 78 per cent of all trunk circuits will be of this type.

6.2.3.2 *Wired-in Trunk and Service Units.* The transmitters, receivers, trunks, and service circuits which do not fit the universal pattern have their combinations of semiconductor circuit packs, networks and relays wired in the conventional manner on mounting plates. Typical is the MF receiver shown in Fig. 27. These surface wired units will be located on the miscellaneous trunk frame and cabled via unit terminal strips to their associated master scanner, trunk signal distributor control points and distributing frame terminations.

6.3 *Central Control*

The central control,[11] shown in Fig. 28, comprises the logic portion of the system central processor. It contains approximately 2300 circuit packs, the majority of which are LLL (low-level logic). This basic logic circuit, a diode-transistor AND-NOT gate, is used to generate all logic functions and memory cells or flip-flops.

The advent of nanosecond logic circuits has necessitated much closer design cooperation between the circuit and equipment designers than was the case in relay switching systems. Wiring patterns and rules had to be developed to insure satisfactory switching speeds, circuit crosstalk

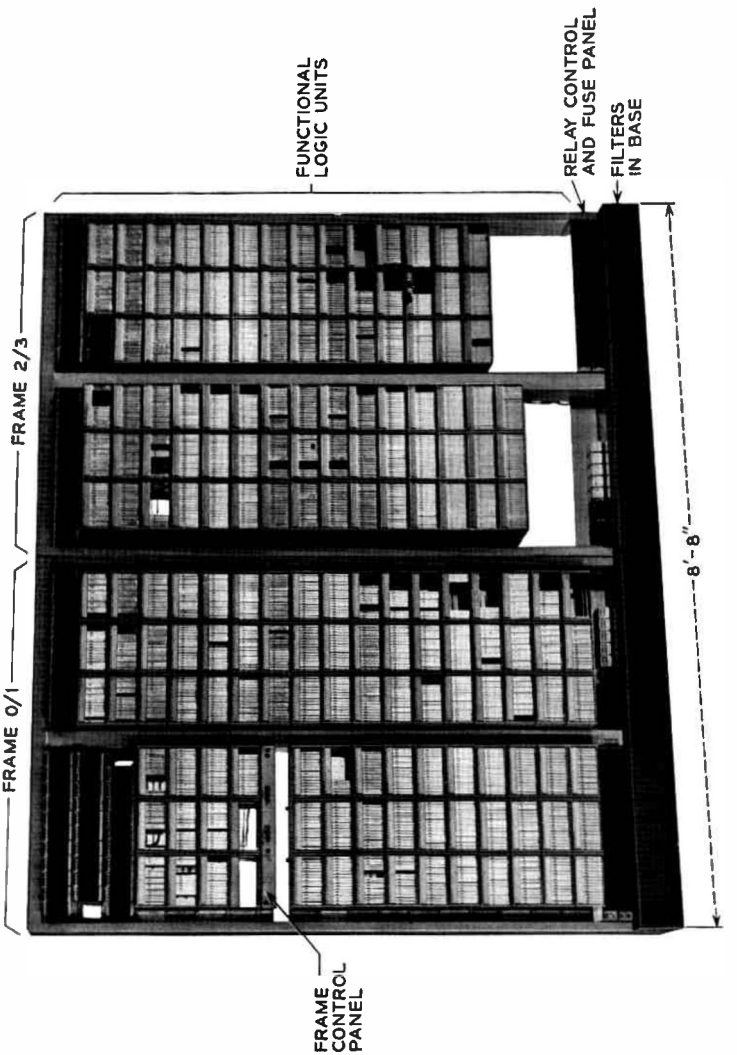


Fig. 28 — Central control.

protection, and a consistent manufactured product. Such requirements have dictated the dense packing of components shown.

The functions of the four bays of equipment can be subdivided under three major headings: input-output, data processing, and control and system maintenance facilities.

The input-output equipment comprises approximately 26 percent of the equipped volume of central control. The transformers, inductors, terminal strips, cable receivers, drivers, and logic are located near the top of each of the four bays: those for the program store and call store address and answer bus in bays 0 and 1, peripheral unit scanner answer and verify bus in bay 1, central control match bus in bay 2, and bus selection for peripheral unit addressing and central control matching in bay 3.

The data-processing function represents 47 percent of the central control volume and is distributed over all four bays. The 23-bit masked and unmasked bus systems used for data handling within the central control required an unconventional circuit pack organization. To provide uniform operation for all bits of a word and to meet timing requirements, no bus bit lead could exceed 6 feet in length. Apparatus for the various registers is distributed over several mounting plates. A particular register function is distributed over several mounting plates, each bit occupying only a few circuit packs on each plate. This permits the output gates of eight different registers associated with the same bus bit to be assigned to the same circuit pack, thus minimizing bus lead length and simplifying flip-flop control leads and maintenance diagnosis.

Aside from power filters, fuse panels, power control, and manual control panel, the remainder of the central control (approximately 20 per cent) is for maintenance facilities. The match bus and buffer register bus are organized in the same manner as the masked and unmasked bus systems. The emergency alarm and maintenance decoder are centralized functions providing a means for detecting, isolating, and performing maintenance checks of a malfunctioning central control. This equipment is located in bays 2 and 3 with portions extending into other functional units of the frame.

6.4 *Program Store*

The program store is the large, semipermanent memory for program and translation storage. It has a capacity of 131,072 words of 44 bits each, or 5.8×10^6 bits in all. The store, shown in Fig. 29, consists of three frames. Two double-bay frames contain the twistor memory modules, access circuits, and other related circuits, while the single-bay frame at

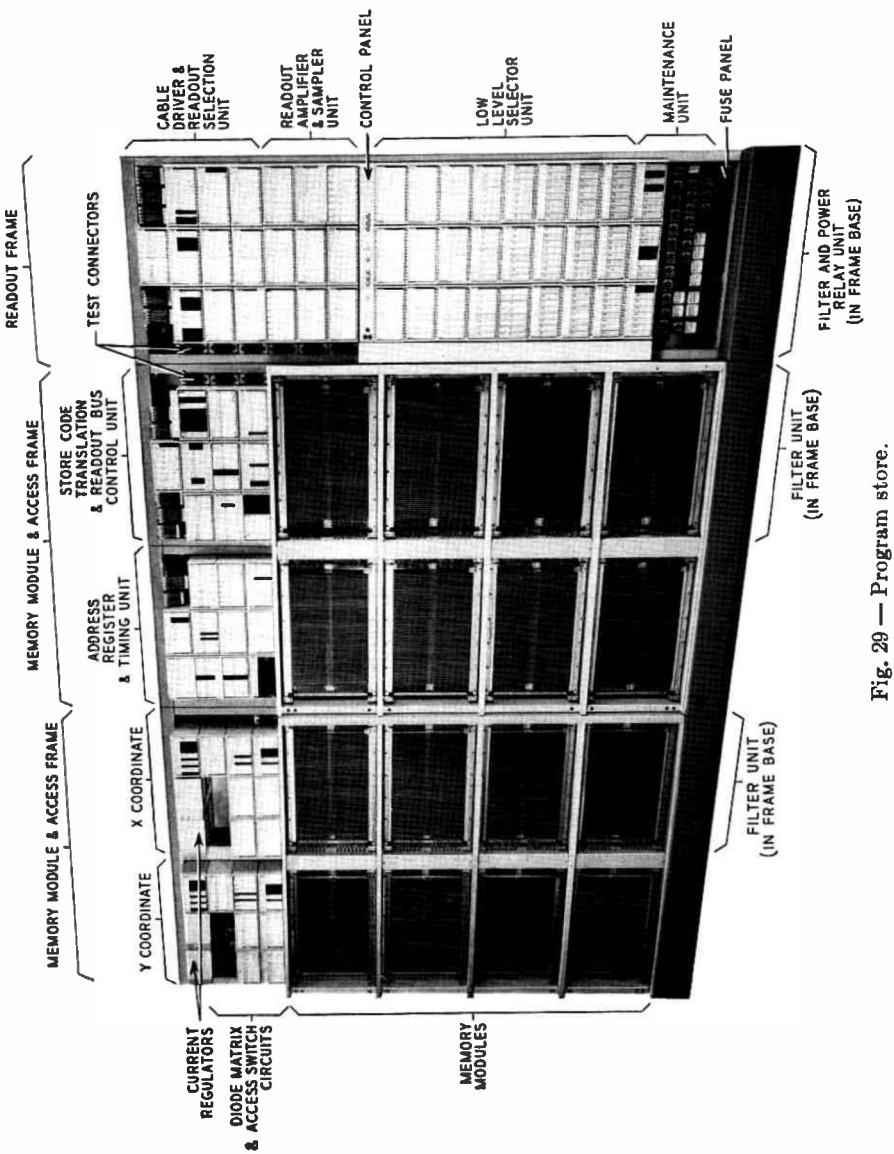


Fig. 29 — Program store.

the right contains circuits associated primarily with readout. Due to the high density of the memory modules, their frames are the heaviest in No. 1 ESS, with a weight of approximately 1900 pounds each.

The memory modules are arranged in a square 4 × 4 array to permit the 256 × 256 coordinate access wiring to be made on the rear with simple, short jumpers between modules.[6] Readout connections between modules, though not quite as simple, are also relatively short with this arrangement.

The input address is received by centrally located transformers at the top of the middle frame. Here it is efficiently channeled to register and access circuits to the left, and also to selectors in the readout circuits to the right, where selection is made as to which readout group and tape (A or B) should be read. Similarly, the timing unit is centrally located to synchronize operation of the access circuits to the left with strobe pulses for the readout circuits to the right. The diode matrix and access switch circuits are above the memory modules where they connect conveniently with the modules by means of a Y-access cable which runs vertically down the left side of the store and an X-access cable which runs horizontally just above the module array.

Readout connections are made on the front side of the store with cable running vertically in shielded ducts between the columns of modules and horizontally in ducts between the first and second and between the third and fourth rows of modules. These cables are further protected from noise pickups by use of close-twisted pairs and by limiting to two inches the unshielded length of leads which connect to the twistor tapes.

The X and Y readout multiples continue in shielded enclosures to terminals of the low-level selector circuit packs at the right. The selected outputs are then channeled to the samplers, amplifiers, and cable drivers above, and to the readout terminals at the top of the frame. This arrangement achieves shortest wiring runs and maximum separation between the sensitive readout circuits and the high-energy drive pulses of the access circuits.

Current regulators for both access and bias are located at the top of the first frame with the diode matrix and access switch circuits. This is important because tracking circuits on these units must maintain a fixed 2:1 relationship between currents for bias and for access drive. Also, this position at the top of the frame, with clearance above and below, provides for adequate heat dissipation.

Most circuit components for the program store are mounted on circuit packs, but the current regulators are packaged in a manner similar to the No. 1 ESS universal trunk circuits, as shown in Fig. 30. The unit contains

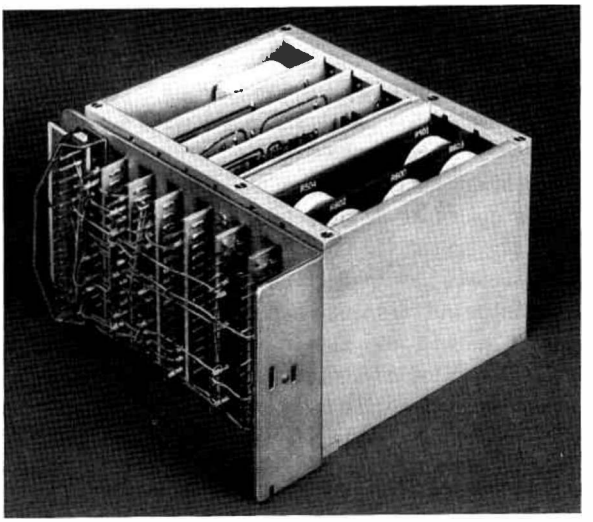


Fig. 30 — Current regulator for program store.

preadjusted potentiometers which compensate for variations in two reference resistors, zener voltage level, and the characteristics of the difference amplifier. Hence, replacement must be as a combination and not as separate packs.

Memory modules are mounted on the frame with three-point suspension to avoid distortion as a result of warping or twisting of the frame during shipping and installation.

Sliding covers are provided in front of the memory modules to protect memory cards from accidental damage. Each cover over a module containing program information is locked in place with a screw as a guard against accidentally disturbing the office program during translation changes.

6.5 *Call Store*

The call store, shown in Fig. 31, is the temporary, electrically changeable memory for telephone calls in process and for storing recent translation changes. It has a capacity of 8192 words of 24 bits each, or 196,608 bits in all.

Special emphasis was placed on short, direct connections between the ferrite sheet memory modules and their associated access and readout circuits.[5] The memory unit, containing four memory modules, is located behind the vertical panel below the center of the store, with Y-access circuits located immediately on the right and X-access circuits on the left.

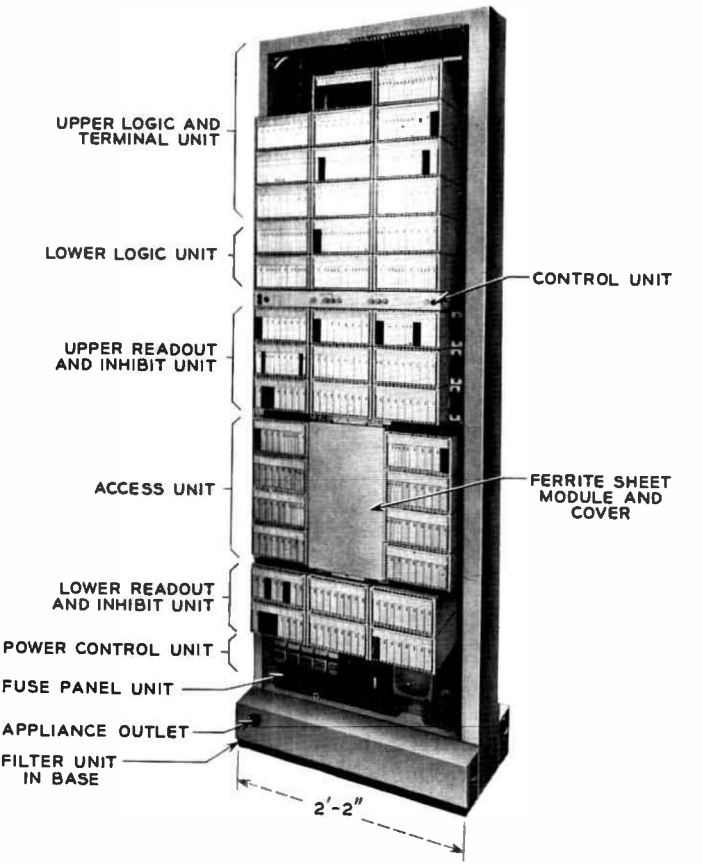


Fig. 31 — Call store frame.

Readout and inhibit circuits are divided, with some circuits above and some below the memory unit.

How direct the connections are is shown in a rear view, Fig. 32. At the right, each of four modules is connected with two cables only a few inches long to carry the X addresses from the diode matrix circuit packs to X connectors on the modules. Similar cables carry the Y addresses from circuit packs at the left to Y connectors on the various submodules of two memory modules, while other cables carry the Y addresses between modules in a slip pattern. To avoid congestion, yet facilitate handling, the access 18-conductor cables use stranded 28-gauge Teflon-insulated wire. A single shield over each cable minimizes interference with the sensitive readout circuits and also provides a common ground.

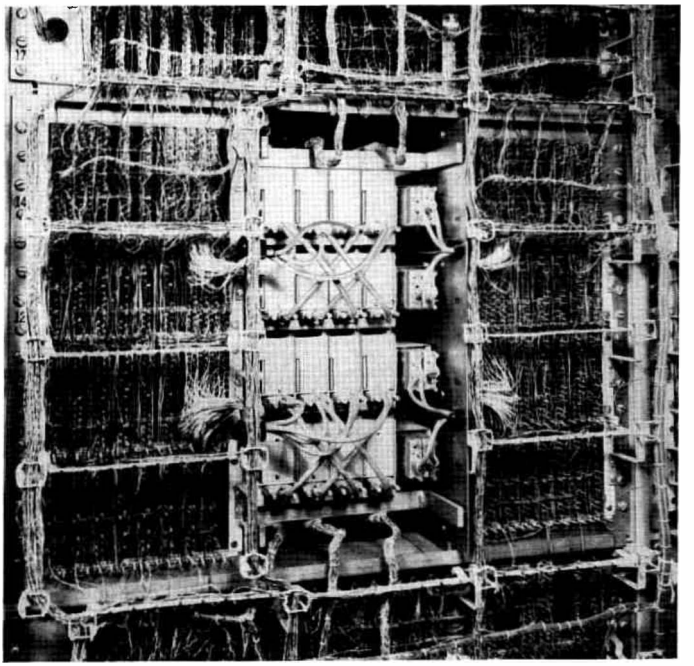


Fig. 32 — Call store — closeup of rear showing cabling of memory modules.

The readout and inhibit connections are carried from the memory unit to circuit packs above and below by two groups of three cables each. The left cable in each group carries inhibit signals, while the remaining cables are for readout. As shown, interconnections between circuit packs are made with a combination of surface wiring and loose wiring similar to that used in central control. Power distribution is by frame local cable.

The four memory modules are plugged together by double-sided connectors, which connect terminals on the edges of five printed circuit boards in each module with corresponding terminals on the edges of five boards in each adjacent module. Four of these boards carry readout connections with 50 connector terminals at each edge, while the fifth board carries a similar number of inhibit connections. Other connectors at the top of the upper module and bottom of the lower module connect with the readout and inhibit cables visible in the figure.

After plugging together, the modules are mounted in a framework which supports four apparatus mountings for circuit packs on each side. The five connectors at top and bottom which mate with this assembly are mounted in a floating manner on hinged brackets which may be

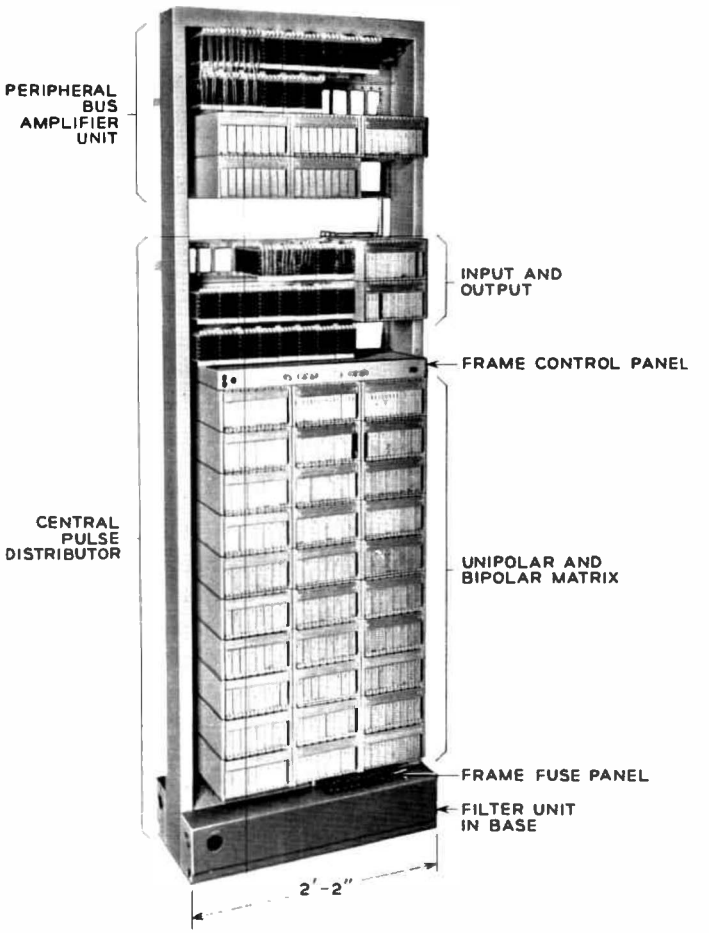


Fig. 33 — Central pulse distributor.

opened or closed from the front. This not only eases assembly and avoids alignment problems but also provides some mechanical advantage to assemble simultaneously five connectors of 50 contacts each.

6.6 *Central Pulse Distributor*

The central pulse distributor (CPD), shown in Fig. 33, is an electronic coincident-voltage transformer output selection matrix designed primarily to enable one of various peripheral units served by a common peripheral bus to accept information transmitted over this bus.

Five hundred and twelve outputs of the matrix are designed to produce pulses of single polarity for the enable function. An additional 512 addressable points of the matrix are combined into 256 bipolar outputs capable of generating pulses of either polarity. These are used to operate remotely located flip-flops associated with maintenance and diagnostic functions as well as certain trunk and service circuits with high repetition rates.

In some offices the CPD loads require a proportion of outputs differing from the 512 unipolar and 256 bipolar outputs. The frame has been wired to permit a limited amount of trading of 64 unipolar points for 32 bipolar points or vice versa in the central portion of the frame. This permits the CPD, through insertion of the appropriate complement of circuit packs, to have a range of capacities from 512 unipolar and 256 bipolar pulse outputs to 0 unipolar and 512 bipolar pulse outputs.

In the application of these frames to a variety of office sizes, it became apparent that an amplifying and load-distributing point would be required for peripheral bus systems having a large number of peripheral units. Since, for electrical reasons, the peripheral bus enable leads which originate in the CPD must be approximately the same length as their associated peripheral bus leads, the peripheral bus fan-out equipment was also mounted in the CPD frame. With this arrangement the bus and enable leads originate on the same frame. By running these leads along the same cable rack, they will be kept approximately equal in length. The amplifier circuit packs are provided in this universally wired frame unit whenever the office has from two to four bus systems, each with a maximum of 50 loads or 450 feet of bus cable between the central control and its most remote peripheral load.

6.7 *Master Scanner*

The master scanner is used to monitor various administrative and diagnostic points throughout the system. This scanner, like others on the line switching, junctor, and universal trunk frames, consists of a 1024-point ferrod sensor matrix and duplicated control equipments. These alternate on a periodic basis in controlling the interrogate and readout pulses of the matrix.

The matrix is divided into 64 groups, each consisting of 16 scan points which are scanned simultaneously. These groups are divided into two general categories, one containing supervisory scan points which are scanned every 100 milliseconds and the other containing directed scan points which are scanned as required by direction of a noncyclic program.

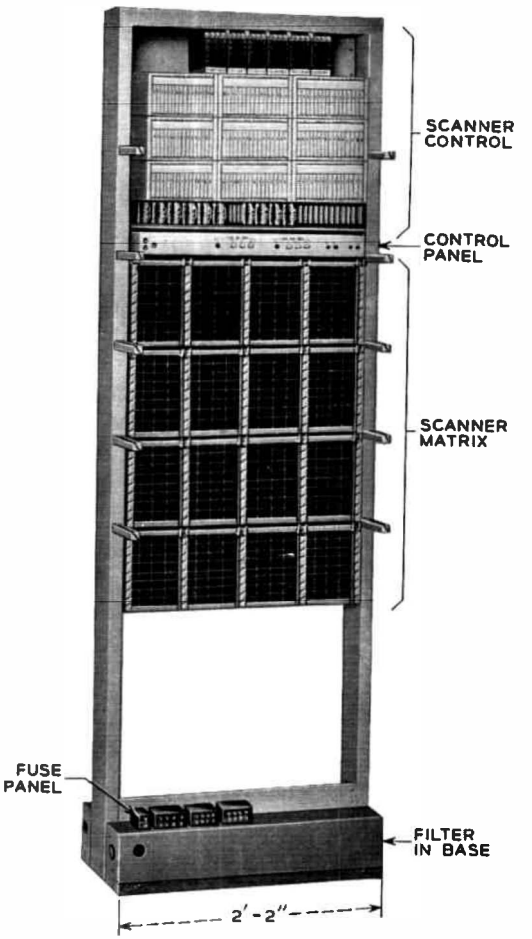


Fig. 34 — Master scanner.

With the exception of certain fixed points, which are the same for all offices and always appear in the same matrix location on the first master scanner, scan points are assigned as required on an office basis. Supervisory scan points are assigned in rows, starting at the bottom of the matrix, and directed points are assigned from the top down.

The master scanner, shown in Fig. 34, is a 2-foot, 2-inch single-bay frame. The electronic control equipment is located in the upper part of the frame to keep the address bus leads as short as possible, and the 1024-point matrix, equipped with 512 type 1E (sensitive) ferrod sensors (two sensors each), is located directly beneath this equipment to minimize the

length of the interrogate and readout pulse leads. To provide for detecting either (a) a contact closure or (b) a change in potential at the scan points of the connecting circuits, both ends of the two control coils on each sensor are brought out to terminals on its front face. For contact closure, all four terminals are cabled to a connecting circuit that is to be scanned. Two of the four connecting leads furnish battery and ground to the ferrod sensor control windings so they may detect a contact closure over the second pair. For potential change, these coils are strapped together, series-aiding, and are cabled to the circuit under surveillance with a single pair of wires to detect a change across its scanned points.

6.8 *Master Control Center*

Reliable system operation and the rapid diagnosis and repair of system malfunctions rely heavily on maintenance programs. Upon detection of a malfunction, either by circuit or program means, fault-recognition programs are called in to recover the system's call processing ability. These programs control any necessary switching of subsystems and request, via memory, an appropriate diagnostic program to localize the fault to a particular package. The results of the diagnostic programs are printed out via a maintenance teletypewriter. Using the maintenance dictionary, these results can be translated to the location of the fault by the maintenance personnel.

Routine test programs are provided to search for faults which are likely to go undetected in normal call processing. These test programs can be initiated either automatically on a scheduled basis or via a manual teletypewriter request.

The master control center includes the communication facilities between the maintenance personnel and the system. Most of these communications will occur via the maintenance teletypewriter.

The master control center consists of the three frames shown in Fig. 35. The two-bay frame at the left contains the maintenance teletypewriter and two magnetic tape recorders for automatic message accounting. The third and fourth bays from the left contain the system control, display, and trunk and line test facilities. The memory card writer occupies the single-bay frame at the right.

6.8.1 *AMA and TTY Frame*

The magnetic tape recorders shown in Fig. 36, used for AMA (automatic message accounting), are provided on an active and standby basis to insure a continuous capability. In contrast to other systems, these tape

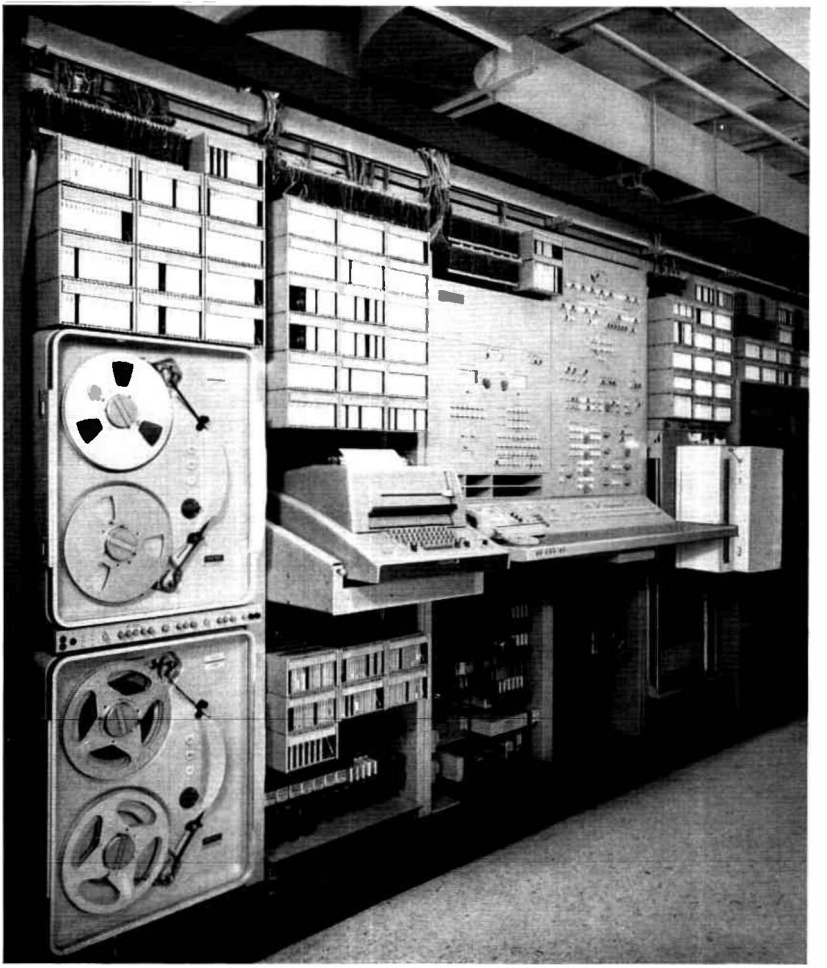


Fig. 35 — Master control center.

records are in the form of completely assembled call data, blocks of which are transferred from the system memory on a start-stop basis. The recorders are driven by three-phase, 208-volt, 60-cps motors supplied from the protected ac power plant to further insure continuity of recording.

The 35-type (keyboard send-receive page printing) teletypewriter is mounted on a retractible shelf to provide improved maintenance access while minimizing its projection into the maintenance aisle. A second

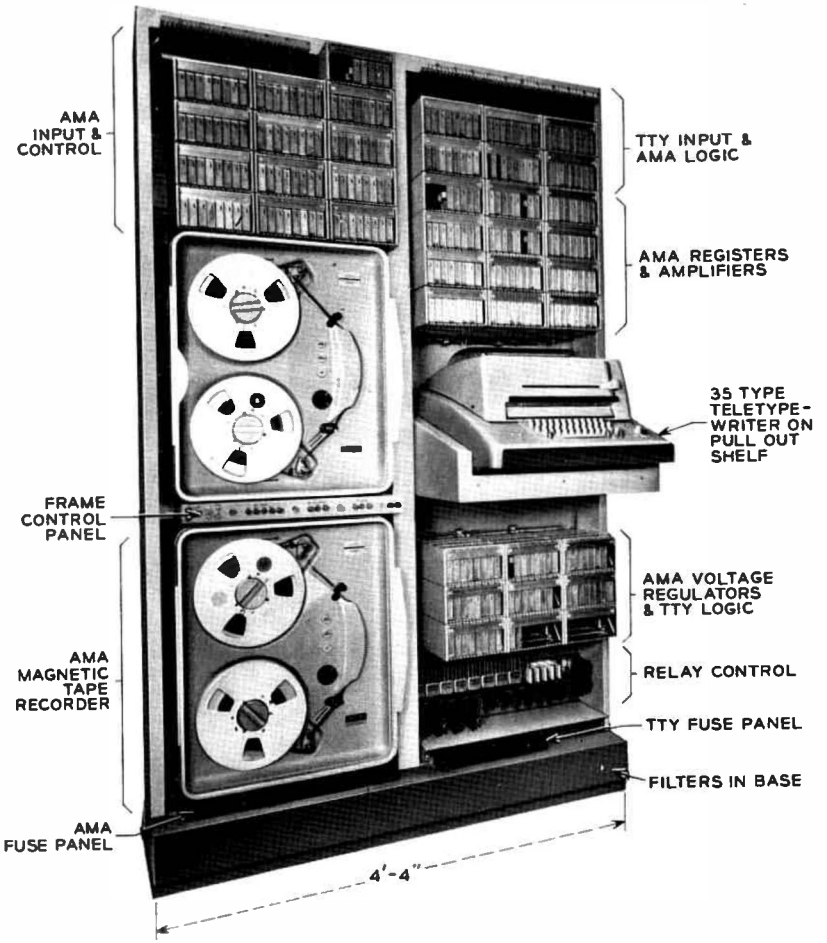


Fig. 36 — Automatic message accounting — teletypewriter frame.

maintenance teletypewriter, pedestal mounted, is installed in the office, when fully attended, or at a remote maintenance center when desired. This machine duplicates the access to the system provided by the unit in the AMA-teletypewriter frame.

6.8.2 *Control, Display and Test Frame*

The control, display, and test frame, located adjacent to the maintenance teletypewriter, provides a system monitoring, manual control, and test center. This equipment permits the maintenance man to observe

the current in-and-out-of-service status of various units, to assert manual control over the system, and to make a variety of line and trunk tests.

The left half of the keyshelf and control panel, shown in Fig. 37, contains a 23B transmission measuring set, a voltmeter, a clock, a telephone set (6-button 560-type or an 18-button 623-type), a number of lamps and pushbutton keys and a TOUCH-TONE dial. This equipment provides for trunk and line testing.

The right half contains lamps, pushbutton keys, and rotary switches. The upper one-third of the panel contains the system alarm lamps and a system block diagram lamp display. An individual display is provided for each central control and program store frame. Two lamps are provided for each other class of unit such as call store, line switching frame, etc. The rest of the panel is arranged in three operating areas.

The status lamps monitor the data routing flip-flops in the central control, program stores, and call stores, and indicate certain troubles such as peripheral control failure, system time-outs, etc. Keys are provided for instituting traffic control, retiring system alarms, and removing power from the frame.

The emergency action area provides manual controls for overriding the system when the system is unable to restore itself to normal operation or when maintenance activity or additions to the office require. The program interrupt control and data word controls permit manual interruption of the program, display of a 24-bit data word, and insertion of a data word into the system.

6.8.3 *Memory Card Writer*

The memory card writer is used for writing, or changing, the information stored on the memory cards of the program stores.[6] Normally, only translation information is changed, but special facilities not generally available in a telephone office permit the writer to be used for program changes as well. The memory card writer, shown in Fig. 38, consists of a single-bay frame 2 feet, 2 inches wide. Like the program store, the card writer requires a maintenance aisle at least four feet wide to permit safe handling of the card loader.

The memory card writer contains the 1A card writing unit and associated 1A card writing head as mechanisms which physically handle the cards and magnetize, or demagnetize, as required, each of the bit magnets. These mechanisms occupy 44 inches of frame height and are located in the lower portion of the frame, with bottom 12 inches above

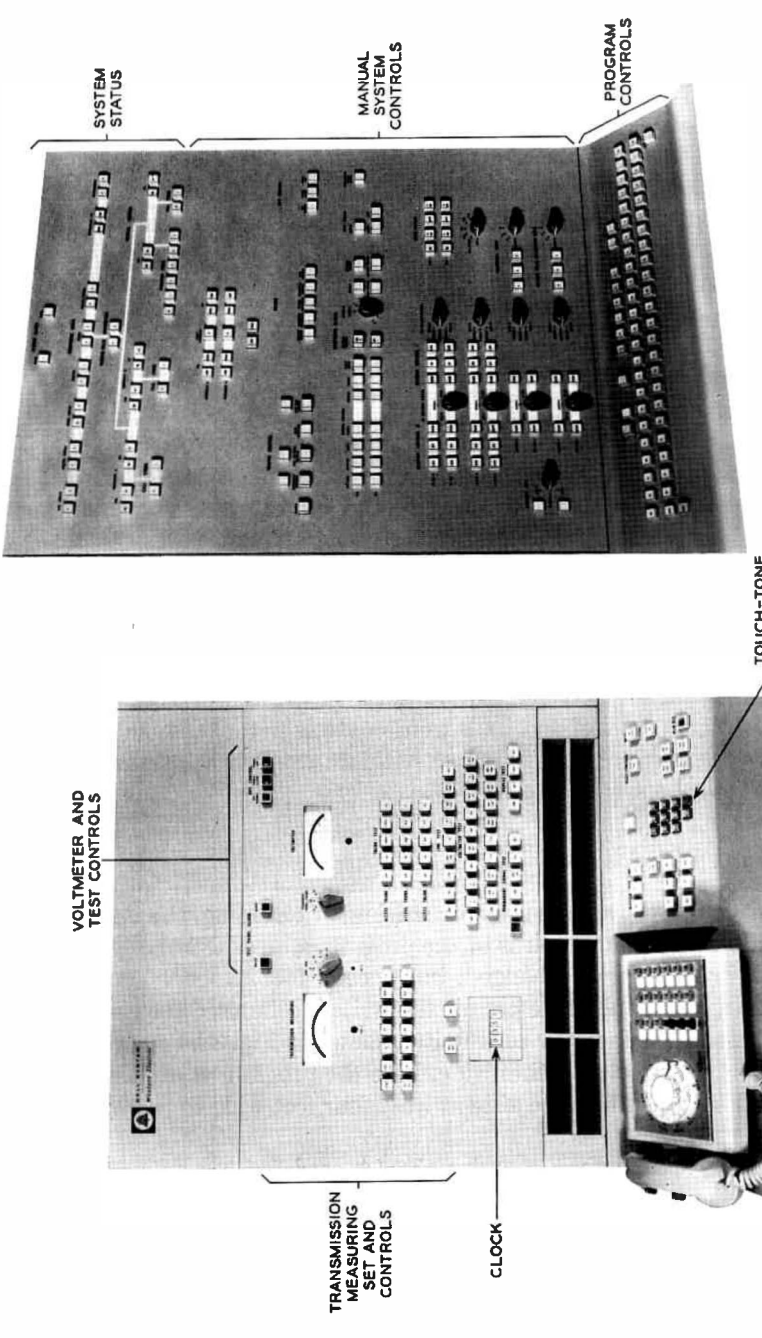


Fig. 37 — Panels of control display and test frame.

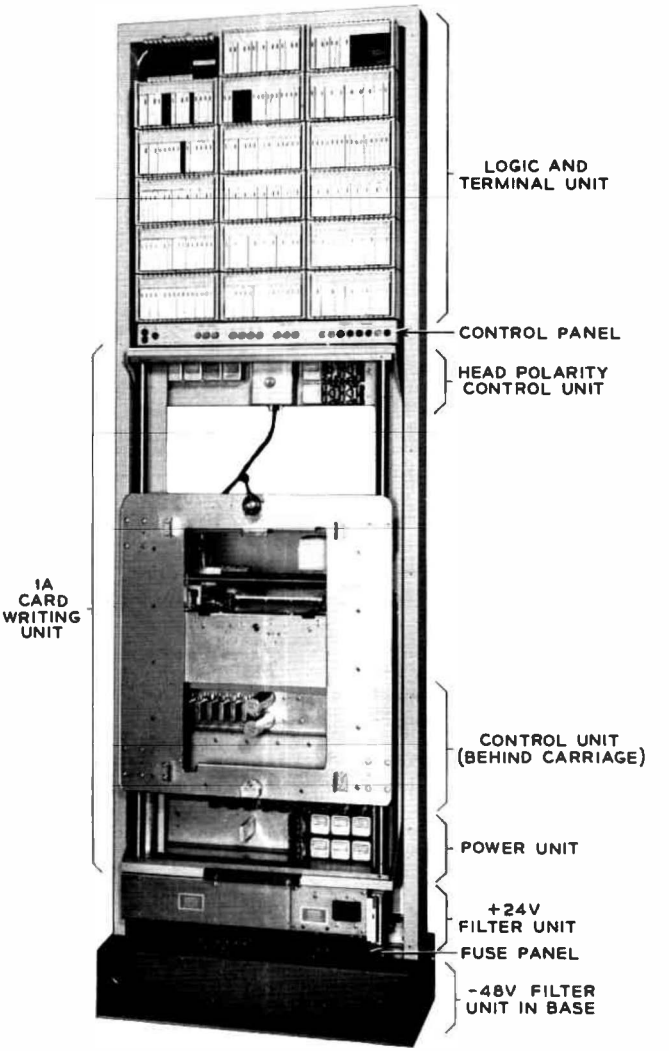


Fig. 38 — Memory card writer.

the floor for convenient attachment and removal of the card loader. The control panel is mounted immediately above the writing unit with the logic circuitry above. The head polarity section is mounted just below the control panel on the rear side of the writing unit, together with the connectors for the writing head.

Other control circuits which are closely associated with operation of the

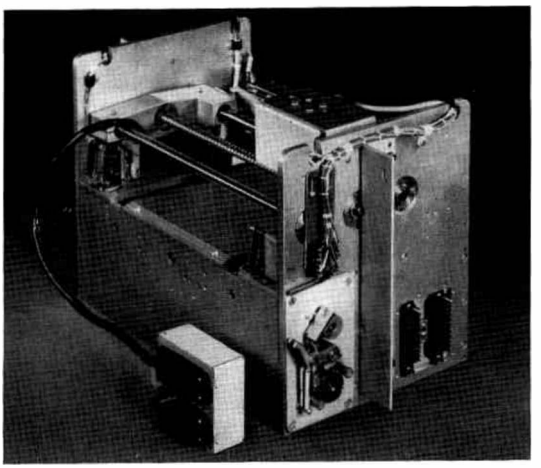


Fig. 39 — Card writing mechanism — end view.

writing unit are mounted near the lower end of this unit, on the rear side, together with a power control unit.

For normal operation, a loader with cards to be written is mounted to a movable carriage on the card writing unit with the left, or "A" end up. At this time the carriage is in its lowest position. The WRITE button on the control panel is then depressed to start the card writing sequence for pass A. When the 64 cards with upward-facing magnets have been written, a buzzer sounds and an INVERT LOADER lamp lights to signal that pass A is complete, and indexing pawls are automatically retracted, allowing the carriage and loader to return to the starting position. The speed of this downward motion is controlled by two hydraulic cylinders. A rubber bumper cushions the shock when the carriage reaches the bottom.

The loader is then removed from the carriage, inverted end-for-end, and reattached with B end up. The WRITE button is again depressed, starting the sequence for pass B. When writing is complete, the buzzer again sounds, an END lamp lights, and the carriage is again returned to its down position.

6.8.3.1 *Card Writing Unit.* The 1A card writing unit, shown in Fig. 39, consists of a framework with a central, easily removed mechanism for handling the cards, a 1A card writing head which is attached to this mechanism, and the rectangular carriage at the front, which is used to mount the card loader. The loader is supported on this carriage by a pin assembly at the bottom and by a lever-operated clamp at the top which engage notches in the ends of the loader. The vertical position of the

carriage is determined by racks at the sides and by pawls at the ends of the central mechanism which engage these racks. The carriage is located horizontally by ball bushings in the corners which ride on the two vertical guide rails.

The central mechanism is capable of removing the cards, one at a time, by means of two finger assemblies which can be driven backward and forward in slots of the writing table. These assemblies are shown in the extended position in Fig. 40. The L-shaped details at the sides actuate switches to signal whether the loader is correctly mounted for pass A or pass B, as may be required. The fingers are spring-tensioned downward against a stop surface. The front ends are hooked and tapered so that as the fingers are driven toward the cards, the tapered ends ride up over the edge and surface of the card until the hooks drop into small, rectangular perforations provided in the cards for this purpose. The fingers are then driven backward, drawing the card across the writing table until the card comes to rest against fixed backstops. The fingers are also spring-tensioned in a longitudinal direction to avoid high forces when the card strikes the backstops. The motion of the finger assemblies is stopped by operation of switches which brake the finger drive motor to a stop, and

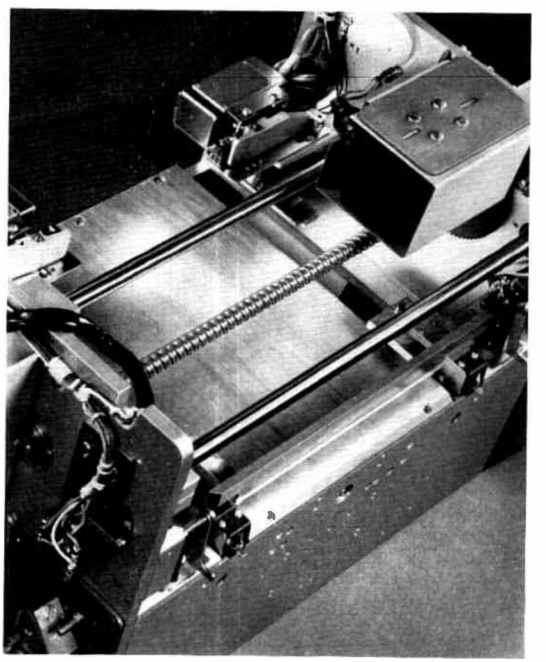


Fig. 40 — Card writing mechanism — top view.

start the drive motor for the writing head. The motors are 208-volt, 3-phase to increase reliability and to deliver a more uniform speed of head travel.

As the writing head passes over the length of the card, it magnetizes the initializing magnets, senses the location of these magnets, and writes each bit of each word as it passes by. Use of the initializing magnets for position-sensing avoids the need for critical mechanical tolerances in longitudinal card and head positions. After the last word on a card is written, switches in the top center of Fig. 40 sequentially disable the sensitive position-sensing circuits and reverse the head drive motor to return the head to its normal position at the left end. Other switches at the left then start a sequence of operations which cause the finger assemblies to insert the card back into the loader, disengage the card by raising the fingers until the hooks clear the top surface, withdraw the fingers to an intermediate position near the center of the table, raise the loader one step by means of the solenoid-operated pawls, drive the fingers forward a fixed distance sufficient to insure engagement of the next card, and repeat the process.

6.8.3.2 *Card Writing Head.* The card writing head, shown in Fig. 41, incorporates a number of design features which have been found necessary to meet its demanding requirements. The head is spring-tensioned against the surface of the card by means of four phosphor-bronze leaf springs. It is supported by two rollers (with ridges which rest on the card in the spaces between the magnets) and a single ball bearing outside the magnet area at the left end. This complex suspension is needed to

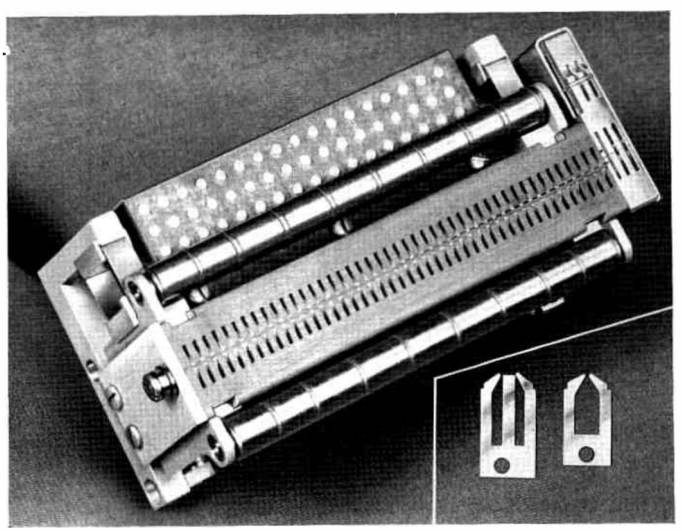


Fig. 41 — Card writing head.

maintain control of the spacing, designed for the range 0.0005 to 0.0015 inch, between the pole pieces and the magnets on the card.

The 45 writing sections of the head contain 6 laminations each, of the type shown in the inset at the right in Fig. 41. Each lamination is 0.006 inch thick, is made of Allegheny-Ludlum Company No. 4750 (similar to a purified No. 45 permalloy) to achieve higher saturation flux, and is insulated on the surfaces with magnesium methylate. The coils are wound in place on one leg of each group of laminations, using solenoid-winding techniques. Each section is shielded to prevent interference between sections. The sections are assembled in the head with the windings on alternate sides to fit on the required 0.100-inch centers. A phenolic spacer holds the air gaps in alignment.

Pole pieces for the initializing heads are similar to those of the writing sections but are of unlaminated No. 45 permalloy, with the coil on the upper yoke between the legs. Each of the two sensing sections contains 9 laminations of 0.004-inch thick No. 4-79 permalloy, which was chosen for high permeability at low flux densities. As shown in the inset, the sensing laminations have two air gaps with spacing which corresponds to the effective length of the initializing magnets. After all sections are assembled in the head, the surface is lightly ground to bring the tips of the laminations, gap spacers, and body to a common plane.

Early experience showed the need for a nonmetallic body for the writing sections to avoid excessive eddy current losses. Phenol fiber was chosen because of its close match in temperature coefficient of expansion with aluminum used for other structural parts.

The flexible cable which extends between the head and connectors at the top of the writing unit uses 28-gauge wire which is stranded and Teflon-insulated to avoid fatigue failures.

6.8.4 *Card Loader*

The 1A card loader, shown in Fig. 42, is used (a) to insert the twistor memory cards into and to withdraw them from the twistor memories in the program store, (b) to transport the cards between the program store and the memory card writer, and (c) to support the cards on the memory card writer.[6] The loader is a box-shaped structure with one side open and with narrow grooves in the top and bottom surfaces, shown in Fig. 43, to support the cards in the same relative position as in the memory module. The end castings, handles, top and bottom plates, and other structural parts are made of magnesium to minimize weight. Even so, the loader weighs approximately 24 pounds when empty and 40 pounds when filled with cards.

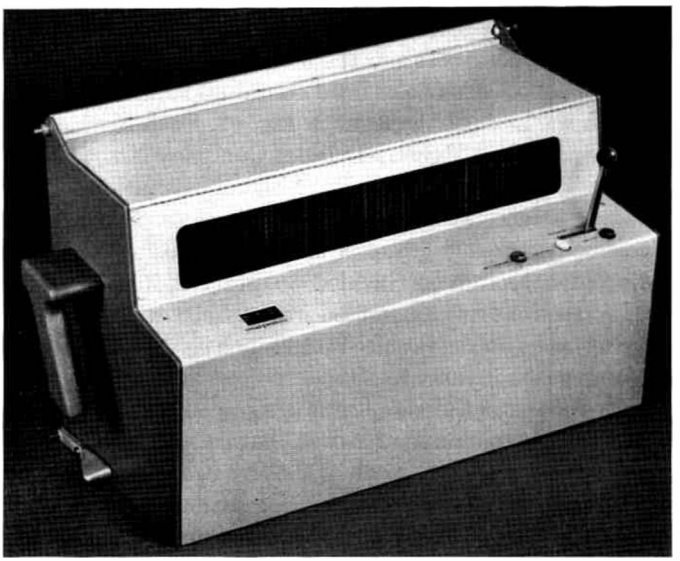


Fig. 42 — Card loader with cards — view of control side.

The cards are engaged by pins on individual finger-like actuators which are arranged in two rows and are attached to a common extruded-aluminum traverse bar. This bar is supported and driven by ball screw shafts at the ends, which are coupled through worm gear speed reducers to a small 115-volt ac-dc motor in the rear center. The screw shafts move the traverse bar through its 7-inch range of motion in about $\frac{1}{2}$ minute

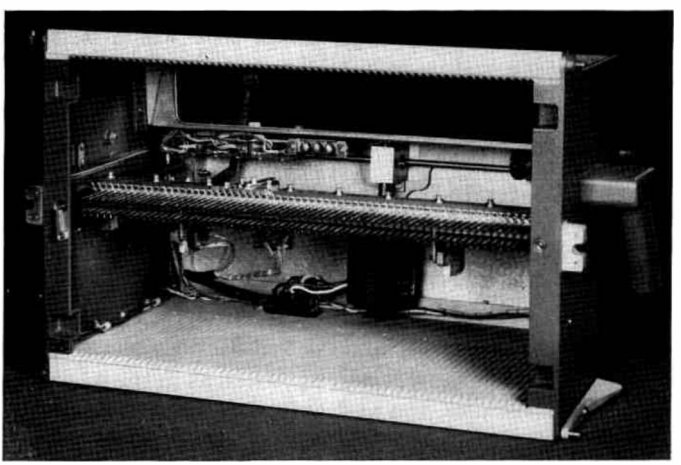


Fig. 43 — Card loader — front side showing card actuators.

with the motor operated on 48 volts dc. The actuators are pointed at the tips and are inserted between pairs of cards in the wider spaces opposite the twistor planes. Movement of an operating lever at the rear causes a gang rotation of the actuators, which causes small transverse pins to enter the openings provided in the cards for this purpose.

The insertion force exerted on each card is controlled within the range 4 to 6 lbs by means of individual, pretensioned springs and an interlock which removes power from the drive motor if the force on any card reaches the upper limit. This assures proper seating of each card in the memory module and protects the memory from damage if a card should jam or stick. Since all cards are inserted simultaneously, the loader must provide total seating forces of 500–700 pounds, sufficient to cause serious damage without the protective interlock.

In use, the loader is precisely positioned on the memory module by means of pins in the four corners, which engage slots in corresponding ears on the module. The cards are engaged or released from the drive mechanism by the operating lever, while the motor drive is controlled by INSERT, NEUTRAL and WITHDRAW buttons on the control shelf. The neutral position is achieved when the actuator pins are centered in the card slots, permitting engagement or release without disturbing the cards or developing high friction forces.

Some features of the loader provided to achieve reliable operation are as follows:

(a) A window above the control shelf permits viewing irregularities that might occur during operation.

(b) The motor receives power through two spring-loaded pin contacts which bear against disk-shaped contacts on the memory module. This construction permits appreciable misalignment in any direction without failure.

(c) A filter in the motor circuit prevents electrical interference from being transmitted to sensitive circuits in nearby frames.

(d) Many points of wear are protected by use of special bearing surfaces.

(e) An adjustable indicator on the control shelf helps the operator keep track of the store and memory module with which he is working.

(f) During card writing, surfaces of the loader engage microswitches on the card writer so that incorrect mounting for a writing pass is detected electrically.

(g) When the cards are fully withdrawn into the loader, the lower ends are held by friction grips which prevent the cards from shifting accidentally when the actuators are disengaged.

(h) A mechanical interlock permits the loader to be attached to the card writer only when the actuators have been disengaged as needed for the writing operation.

6.9 *Power Distributing Frame*

The power distributing frame is the battery load distributing point of the system. The three power feeders (−48 volts, ground, and +24 volts) from the power plant terminate on bus bars on the frame. These bars in turn supply the fuse blocks for individual frame feeder fuses. Two 35,000-μf capacitor banks are provided near the bottom of the frame to provide low-impedance shunt filters across the power supply feeders (−48 volts to ground and +24 volts to ground).

The individual load frames are supplied by feeder pairs or triples (as required by frame loads) from 5-, 15- or 30-ampere cartridge type fuses having $1\frac{1}{3}$-ampere alarm fuses in parallel with them.

Each power distributing frame has a 400-ampere peak capacity for each voltage and is located in the office area of the frames it serves. Duplicated circuits are fed from different power distributing frames.

6.10 *Ringing and Tone Supply Frames*

In keeping with the application of semiconductor circuitry to perform switching functions, new forms of ringing and tone supply equipment have been designed. Two sizes of plants are available to accommodate the wide range of office requirements — a $\frac{1}{2}$-ampere capacity plant, shown in Fig. 44, and a 6-ampere plant. In both sizes the frames are located in a switchroom frame line.

These plants (806H and 808A) provide ringing current, tones, and signaling interruptions. Twenty-cycle ringing (not audible) is generated by 110A frequency generators which furnish a regulated output. All tones except high tone consist of two signals, generated by transistor oscillators, which are added together and amplified in power transistor amplifiers. Duplicate 20-cycle ringing generators, tone generators, and interrupters are provided. Normally all generators are running, but only the interrupter supplying the load is powered. The outputs of both ringing generators are monitored for high and low voltage by the system; it selects the generators to be used and automatically transfers to the other set of generators in the case of failure. All generators and interrupters are transferred as a unit. Manual controls are provided to supersede automatic control when necessary.

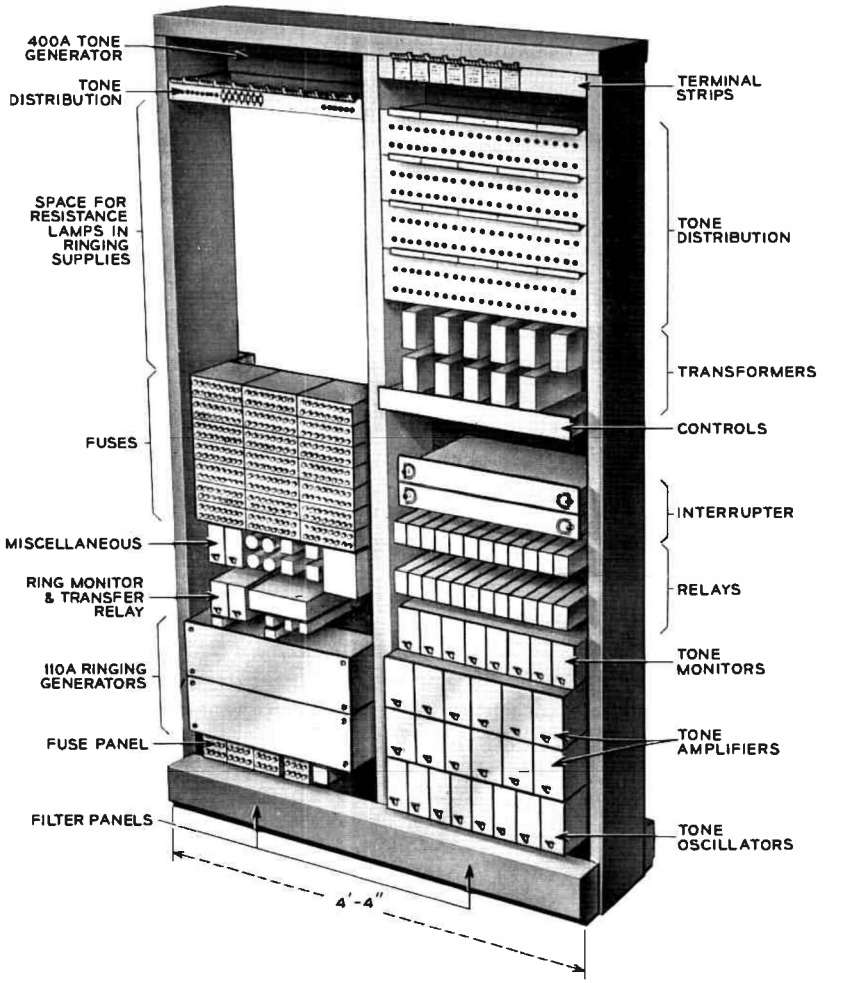


Fig. 44 — Ringing and tone supply frame (806H).

All ringing and tone supply fuses, decoupling resistors, and resistance lamps required by the various load frames are located on this frame.

6.11 *Recorded Announcement Frame*

The recorded announcement frame provides for a maximum of 6 announcements on a magnetic drum recorder (approximately $9\frac{1}{8}$ inches diameter and $1\frac{9}{16}$ inches long). Each announcement channel has a record-reproduce amplifier associated with it. Distributing resistors are

provided for each announcement channel to isolate the outputs, which may total 120 (20 per channel).

The supervisory control unit, a 624 telephone set, is used to select the desired channel for recording or monitoring. This unit, which may be remotely located, can serve two recorded announcement frames.

6.12 *Miscellaneous Frame*

This frame is designed to accommodate a variety of units which require neither signal distributor nor scanner control. These units include emergency manual lines, supplementary AMA tape recorders (for those offices requiring more than the two provided on the AMA-teletypewriter frame), a multiplicity of common systems units, the power for test battery supply, etc. They are designed to accept a number of standard power filter, fuse panel, and control panel combinations to meet varying office requirements.

The miscellaneous power frame is the above frame equipped with (a) the +4.5-volt supplies required by the central controls, (b) the +130-volt and −130-volt fuse panels, (c) the ac distribution panel for 208-volt, 3-phase and 120-volt, single-phase loads requiring protected or essential 60-cycle supply, and (d) the floor alarm units. Two of these frames are located near the first pair of power distributing frames to permit the central control +4.5-volt supply to be run with the ground return cable between central control and the power distributing frame to minimize stray noise pickup.

6.13 *Power Plants*

As shown in Table V, the power plants associated with a No. 1 ESS include:

(a) Two 111A battery plants with very large battery voltage swing tolerances, which avoid emergency cell switching and counter-cell switching. One is a −48-volt plant with a voltage range at the power distributing frames of −43.75 to −52.5 volts. The other is a +24-volt plant with a voltage range at the power distributing frames of +21.75 to +26.25 volts. Power from these plants in the power room is delivered to two or more power distributing frames in the switchroom.

(b) The ringing and tone supplies are located in the switchroom. These are described in Section 6.10.

(c) +130- and −130-volt dc-to-dc converters (610B power plant). These units convert the −48 volts to the potentials needed for coin control. Power from these plants is delivered to fuse panels on a miscel-

TABLE V — POWER SUPPLIES

Power Supplies	Type of Plant or Unit	Capacity (Rated)	Code
In power room			
−48 volt dc (−43.75 to −52.5 volts)	storage batteries (without emergency cell or counter cell switching) rectifier charged	10–800 amp	111A
+24 volt dc (+21.75 to +26.25 volts)			
+ 130 volt dc	dc-to-dc electronic conversion from −48 volts for coin control	¾ amp	610B
−130 volt dc		2 amp and 5 amp	651A
Reserve ac supply	dc motor-driven alternator for 120/208 volt, 1- and 3-phase power	1½ kw	504B
		5 kw	
In switchroom			
Ringing and tones on RT frame	electronic generator with a precise tone plan	0.5 amp	806H
		6 amp	808A
+4½ volts on MP frame	derived from +24 volts		
PBX talking battery filter on misc. frame	coil and capacitor panels	15, 25, and 50 amp	
±120 volt for Frames: AMA-TTY, CDT, RA, RT, Misc. for TTY data sets, test battery supply unit, 2A sending panel	commercial power with or without reliable supply distributed from MP frames		
Appliance outlets Frame lighting	distributed from ceiling-supported busway		
±208-volt, 3-phase for AMA-TTY and CW frames AMA recorders on misc. frame	commercial power with or without reliable supply distributed from MP frames		
±208-volt, 1-phase for RT frame			

laneous power frame in the switchroom and distributed to all frames in the office which require it.

(d) A small, emergency 504B alternating current plant (with an alternator driven by a dc motor). Protected power from the 208-volt, 60-cycle, three-phase alternator is delivered to a circuit breaker and fuse

panel on a miscellaneous power frame in the switchroom. From this panel single-phase and three-phase power is distributed to all frames in the office which require ac power at any time commercial power fails.

(e) An engine alternator to substitute for commercial power to charge batteries and supply essential ac loads after power failure has persisted for a time.

VII. INTERCONNECTING METHODS

The use of high-speed electronic circuits in packs assembled in large numbers has introduced a variety of restrictions in the unit, frame, and interframe wiring. Distributed impedances on circuit packs and their interconnecting wiring must be very rigidly controlled in a nanosecond pulse system. The control of transient noise requires (a) the use of compartmented cable racks, (b) segregated cable and wiring paths on the frames, (c) filters on all dc power supply feeders, and (d) special frame grounding practices. Some of the related problems and their solutions are discussed here.

7.1 *DC Power*

Early tests of the dc power distribution system coupled with tests of major functional elements (brass-board variety) showed the need to introduce load filters at the functional frames in addition to a common low impedance filter within the switching equipment area. Three conductors (+24-volt, −48-volt and a common ground feeder) are run from the power discharge fuses to each of the power distributing frames (centralized power distribution points) where low-impedance shunt filters (35,000-μf capacitor banks) are provided for the +24- and −48-volt system.

Individual frame filters (usually choke input L type or capacitor input π section) are designed to restrict the rate of change of current on the frame supply feeders to less than 0.1 ampere per microsecond. This limits the noise produced at the power distributing frame filter to less than one volt. The central filter can adequately attenuate the noise transmitted to other frames to less than 0.5 volt.

Power distribution leads are run as two- or three-conductor cables between the power distributing frame and the individual frame filter panels. The frame feeder sizes have been matched, as have the filters, to the frame load requirements to insure adequate system operation under conditions of commercial power failure (within the engineered battery

reserves). A maximum drop of one volt* between the power distribution (PD) frame and the individual frame fuse panel bus is anticipated (within a maximum loop distance of 175 conductor feet).

All even-numbered frames (except the program store and frames having duplicated control equipment such as the network frames) will be fed from an even-numbered PD frame and all odd-numbered frames from an odd-numbered PD frame. Two sets of feeders, one from each PD of a pair, will be run to those frames having duplicated control equipment.

To minimize noise caused by stray ground circulating currents or by transient noise potentials within the building, all frames and cable rack are insulated from the building at the time of installation. The frames (and cable rack) are bolted together and individually bonded to a No. 6 AWG frame aisle ground conductor. This ground network connects to the ground bars of the PD frames to provide personnel protection. The PD frame ground bars are interconnected with 750 MCM cables. This ESS ground network is connected to the central office ground, the protection frame ground and ac power entrance ground at only one point (ac neutral or ground return conductors are not connected to the system ground within the ESS switchroom.)

Cable racks (and conducting cable sheaths) from any uninsulated frames (including the protector frame, power room frames, and any in other switchroom areas) are insulated from the ESS frames by insulating pads, fasteners, or isolating sections. Power plant ground returns are similarly insulated from their power plant frames.

Limited amounts of +130 volt and −130 volt power (voltage limits 125 to 135 volts) are required for coin control but no other No. 1 ESS functions. These are distributed to the coin trunks via the 130-volt distributing fuse panels located on the miscellaneous power frame.

7.2 *AC Power*

Some critical ac loads which can tolerate a service interruption of only a few seconds are: (a) AMA magnetic tape recorders, (b) recorded announcement machines, (c) maintenance teletypewriters and data sets, (d) master time-of-day motor, (e) tone interrupter, and (f) 20-cycle static subharmonic ringing generator in smaller offices.

Normally these loads will be operated from commercial ac service. In the event of ac power failure, the 504B emergency power plant will

* In the case of the junctor frame, this figure is allowed to be 1.7 volts maximum, since this will not result in denied service or a reduction in supervisory range but only a slight transmission impairment on limiting loops.

supply the needed 3-phase, 208-volt and single-phase, 120-volt, 60-cycle power within 5 seconds.

This protected ac supply as well as the essential ac supply furnished by the standby engine alternator is distributed from an ac distribution panel located on a miscellaneous power frame. This panel is equipped with circuit breakers for the 3-phase and single-phase, 280-volt loads, and fuses for 120-volt, single-phase loads. A power factor correction panel is also supplied to compensate for highly inductive loads.

New cables (410A and 411A) were coded to provide 18-gauge triples and pairs for ac distribution. These PVC-insulated cables have a distinctive yellow sheath to permit easy identification.

7.3 *AC and DC Cabling*

The ac and dc distribution cables are run from the miscellaneous power and power distributing frames in the rear (wiring aisle side) of the frame line cable rack. Power cross-aisle racks are provided as required to distribute these cables to their proper frame lines. There, ac cables are dressed down the rear of one frame upright while the dc cables are contained within one or more frame uprights.

7.4 *DC and VF Signaling*

DC signaling leads, relay connections, and tip and ring leads through the networks and trunks are subjected to a wide variety of transient noise pulses. In an all-relay switching system, these transients usually do not cause a system malfunction, because of the inherent insensitivity of the relay network.

7.5 *AC Pulse Signaling*

Master scanner pairs and manual override pairs from the master control center are not very noisy nor very vulnerable to noise. These are relatively long cables which have intimate contact with logic circuitry, so there is some danger that any noise they generate or transmit could disturb more sensitive areas. Their exposure to relay noise could introduce this noise into vulnerable areas of logic. The 0.5-microsecond unipolar pulses carrying data and control information between the central control, central pulse distributor, master control center, stores, and peripheral units are the most vulnerable to the noise of signaling connections. They are run as balanced circuits and brought into each frame through shielded transformers, with the lead length on the unbalanced output to the cable receiver kept less than 18 inches.

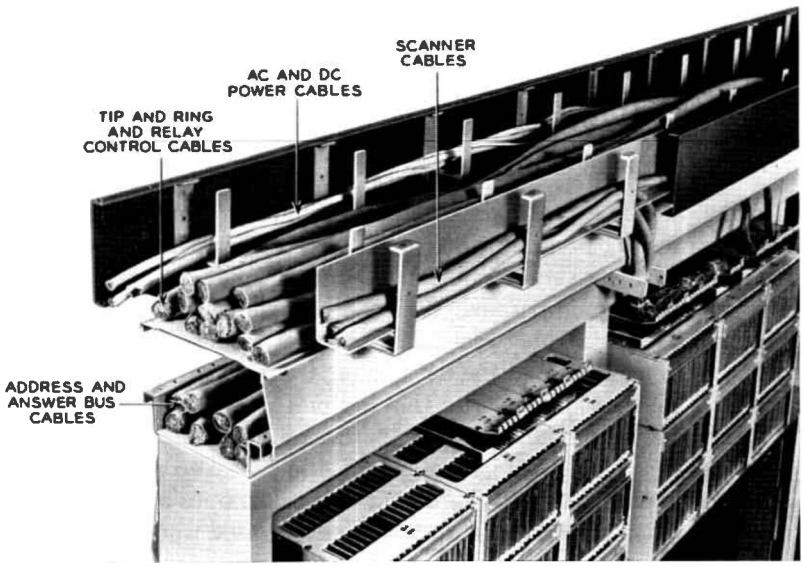


Fig. 45 — Compartmented cable rack.

The 0.5-microsecond signal distributor bipolar pulse leads are not noisy and are less sensitive to noise than the unipolar. However, they are still a problem, since they enter frames without using shielded cable receiver transformers, and these leads cannot be kept as short as those to the cable receivers.

7.6 *Cabling Practices*

Studies and laboratory tests showed that the economical answer to these interframe cabling problems lay, not in the use of shielded wire or coaxial cable, but in a compartmented cable rack as shown in Fig. 45. The 0.5-microsecond unipolar and bipolar pulse leads are run in the lower compartment (having removable front and rear covers) with minimum lead length exposure between the compartment and the transformers mounted at or near the top of each frame.

The scanner cables are run in a shielded channel at the front (maintenance aisle side) of the cable rack where they can drop down to frame terminal strips with relatively short exposures. The tip and ring leads and relay control leads are placed in the center top section of the cable rack. These cables are run over the scanner cables, down the front of the frame upright to the frame or unit terminal strips or ferreed switch ter-

minals. AC and dc power distribution cables are run in the rear top section.

Separate cross-aisle racks are provided to maintain this needed separation between different classes of leads.

7.7 *Frame Wiring Methods*

During the brassboard development stages of central control and stores, the testing program proved that some wiring methods common to relay switching systems could not be used for high-speed pulse circuits. The lengths and parallel paths of signal wires had to be controlled to minimize crosstalk between circuits. This resulted in circuit pack placement restrictions and the development of new wiring practices and associated hardware suitable for this equipment.

Where present practices of unit surface wiring and interunit loose wiring and local cables can be used, this is the logical economical answer. In general, trunk units and trunk and network frames are wired in this way. This wiring method, as shown, can also be used where circuit packs and interconnections are less congested and circuit design permits.

Where high densities of circuit packs and wired interconnections exist, as in central control, special wiring procedures prescribe specific routings for surface wiring, loose wiring, and local cables. The following basic wiring rules are recommended:

(a) Use most direct route via horizontal and vertical runs.

(b) Do not sew signal wires into a cable.

(c) Surface or loose wiring can be used provided a multiple does not exceed 6 feet and other multiples do not follow the same path for more than 3 feet.

(d) Twisted pairs are used (one wire grounded) when multiples are 6 to 10 feet in length.

The wiring rules illustrated in Fig. 46 are the outgrowth of this development. Fig. 47 shows these rules applied in a wired frame. The individual mounting plates are first surface wired in the shop and verified with a dc test. The associated mounting plates of a unit are surface wired together. The loose wiring support details are then mounted on the units, and the units in turn are installed on the bay or frame. A preformed loose wiring harness is mounted on the support details and the leads are terminated.

VIII. DISTRIBUTING FRAMES

In many large central offices, the main distributing frame (MDF) now determines the length of the building. No. 1 ESS employs frames one-

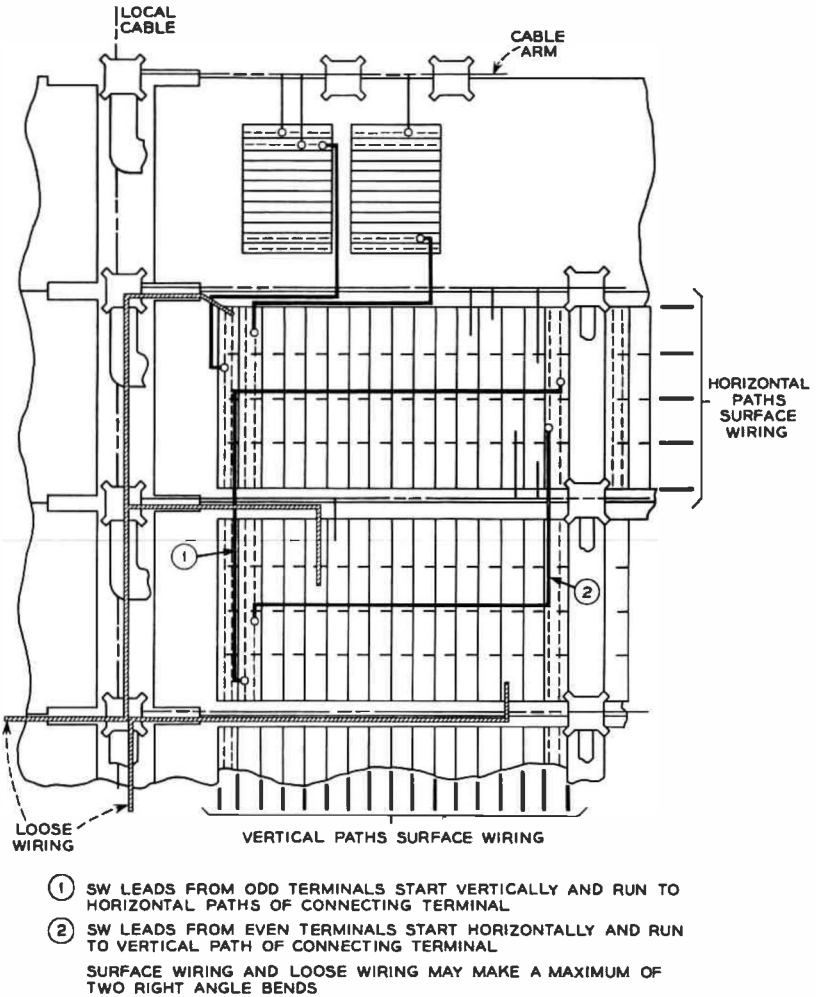


Fig. 46 — Central control wiring rules.

half the height of these large MDF's and requires one-third the floor space of existing offices. Distributing frames of radically new design were needed to match the space economy of the new system. Fortunately, the new programmed logic, the electronic memories, and the nonexistence of sleeve leads eliminated some of the need for cross-connection capacity formerly required. Fewer jumpers are required and fewer changes are needed in the jumpers that remain. In preceding systems, the frequency of jumper changes is increasing and their costs are climbing every year. What was needed for No. 1 ESS, therefore, was a design which would

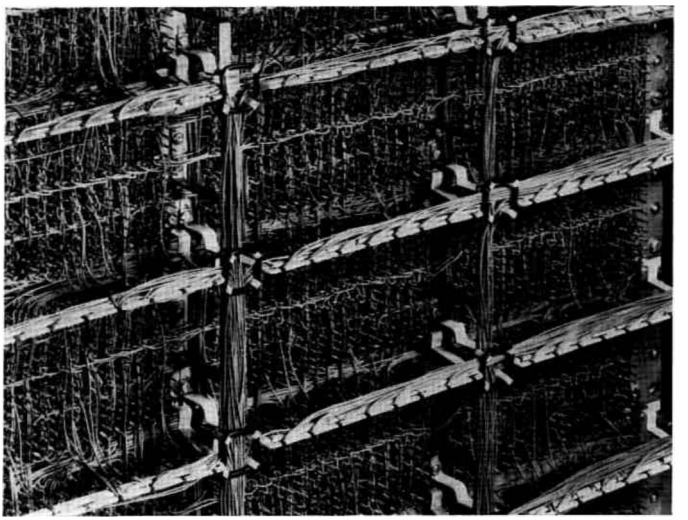


Fig. 47 — Central control wiring methods.

shrink the distributing frames to a compatible size and correspondingly reduce the cost of cross connections. This was accomplished with the new frame designs.

8.1 *Protector Frame*

One module of single-sided protector frame 6 feet, 6 inches long, 8 feet high and 1 foot deep (see Fig. 48) provides the protectors for 6000 outside plant pairs. A protector unit (without heat coils) guards against lightning and other high foreign potentials and serves tip and ring conductors of a pair. The connector for 100 protector units is shipped with a stub cable long enough to reach its termination in a cable vault or on a wall rack. The module has 12 verticals of five connectors each. Provision is made for test desk circuit appearances as well as for every needed maintenance access to any circuit or group of circuits. A compact loudspeaker system incorporates the microphone in a frame upright and the loudspeaker in the upper right-hand corner of the module. No plugging-up panels are required since lines in trouble are automatically routed to trouble intercept by central control. A conductor identification circuit will permit the cable splicer to identify pairs without manual help at the central office.

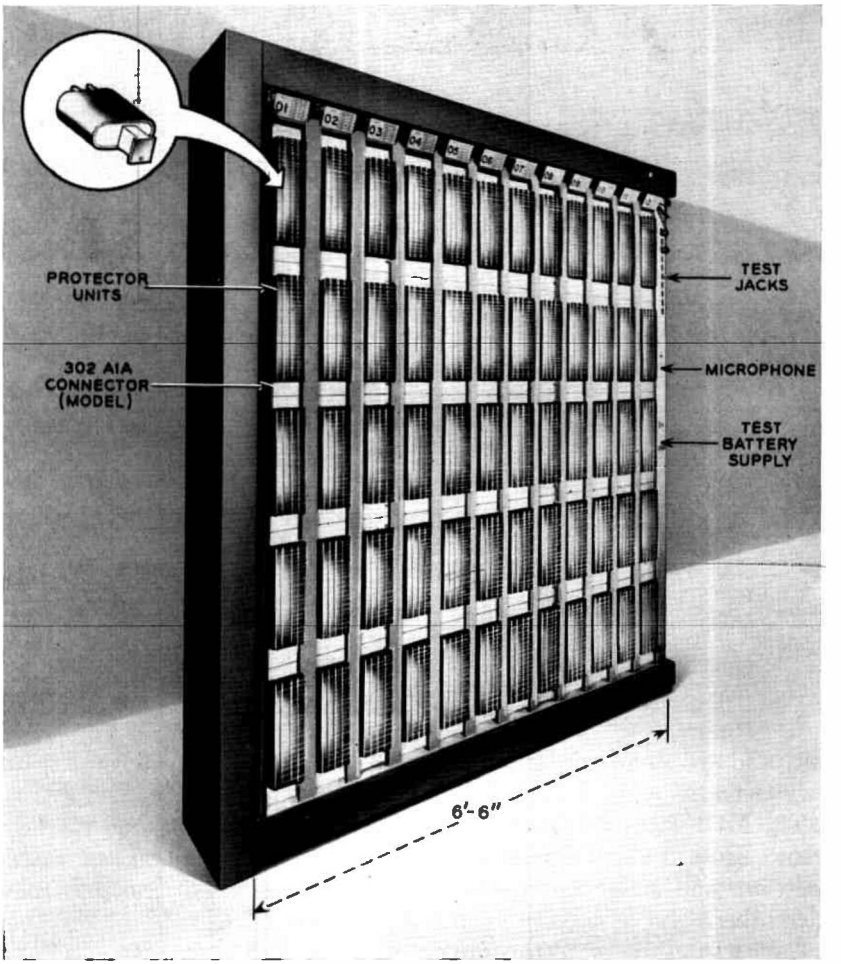


Fig. 48 — Protector frame.

8.2 *Main Distributing Frame (MDF)*

The MDF, shown in Fig. 49, makes a corresponding reduction in size by eliminating most long jumpers. Outside plant pairs and inside equipment pairs are interspersed in a manner which greatly reduces jumper length, the space for storing jumpers, and the space for making cross connections. The cost of making changes is reduced by the elimination of most long jumpers, by the introduction of "quick-connect" terminal strips, by avoiding changes completely where program and circuits per-

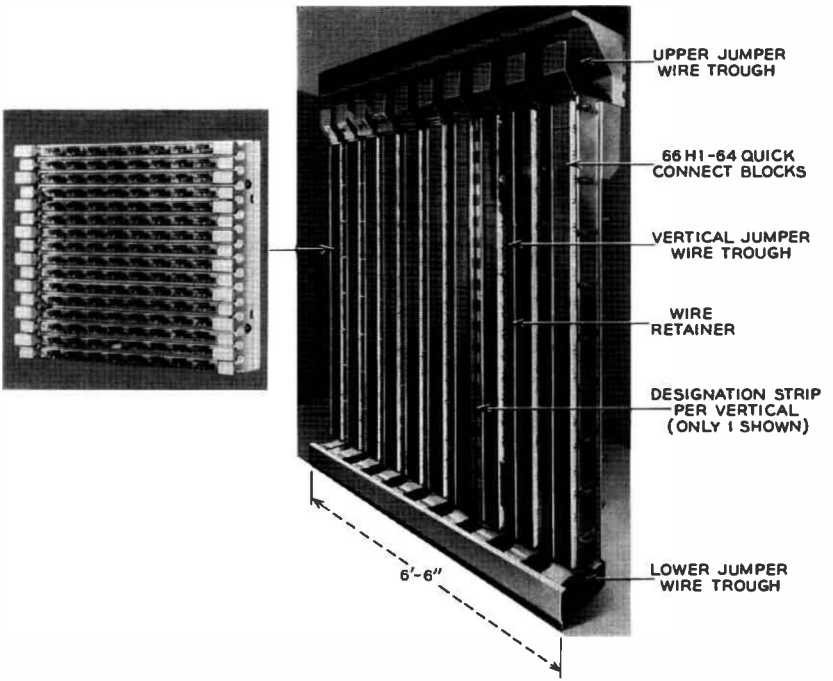


Fig. 49 — Main distributing frame.

mit, and by simple designation procedures which make terminations easy to find.

The MDF faces the protector frame across a 4-foot aisle, and the two frames grow at about the same rate. The frameman will find associated protector and MDF terminations across this aisle usually not far from each other.

Cables from the protector frame are interspersed with system switchboard cables on the single-sided MDF. Line link networks and auxiliary line circuits are cabled there directly; trunk, service, and associated auxiliary circuits appear first at an intermediate distributing frame (IDF) where they are cross connected to tie-cables to the MDF. Thus the trunk circuits have two jumpers each, for multiplied access to outside plant pairs.

A module of single-sided MDF is 6 feet, 6 inches long, 8 feet high and 1 foot deep. This gives 6000 pairs from outside plant access to as many as 6080 central office pairs. Connecting blocks on each of ten verticals have 1216 "quick-connect" terminal pairs in one plane on the front (each terminal arranged for two cross connections) and solderless

wrapped terminals in one plane on the rear. The five odd-numbered verticals terminate 1200 protector frame pairs each. The five even-numbered verticals terminate 1216 system pairs each. Cabling patterns distribute protector and system pairs on their interspersed verticals in a manner designed to permit the great majority of all cross connections to be between adjacent verticals. Jumper troughs above and below the connecting blocks provide for the longer jumpers between nonadjacent verticals.

Lines from line link networks are distributed horizontally across the midsection of the system verticals. Line auxiliaries such as long-line circuits and bridge lifters are similarly distributed below them. The cables to the intermediate distributing frame (to cross connect to trunks and trunk auxiliaries there) are distributed horizontally across the upper sections of these verticals.

Each fully equipped vertical with pairs from the protector frame presents 1200 outside plant pairs for direct cross connection to as many as 2432 system pairs on the two adjacent verticals (left and right). The 1200 pairs on an intermediate vertical include 20 pairs from each of 60 different 100-pair cables from outside plant.

This vertical distribution of outside pairs combined with the horizontal distribution of system pairs guarantees a wide exposure of outside to system pairs of different types on adjacent verticals. To provide further homogeneity as an office grows, the protector frame modules are equipped in numerical sequence, as at present, but at each stage of growth, protector pairs are distributed over three MDF modules in a manner such that the last MDF module in a line-up is one-third filled when the second last module is two-thirds filled and preceding modules are completely filled.

Service observing cross connections for lines and PBX trunks are made on the rear of the MDF so that this activity does not interfere with the cross connecting of outside to system pairs on the front of the frame. Circuits from the service observing desk (and from a No. 6 service observing set) are distributed over small jack panels located at the top rear of the five verticals for protector frame pairs. Patch cords cross connect from these jacks to the rear of nearby line terminals as required.

To make all terminations easy to locate, a vertical designation strip is hinged in front of each vertical jumper trough. When rotated to the left or right, it covers the adjacent one-half of the jumper trough on that side. In either position it clearly designates all terminals in the vertical. Designation areas are in colors to assure quick identification of out-

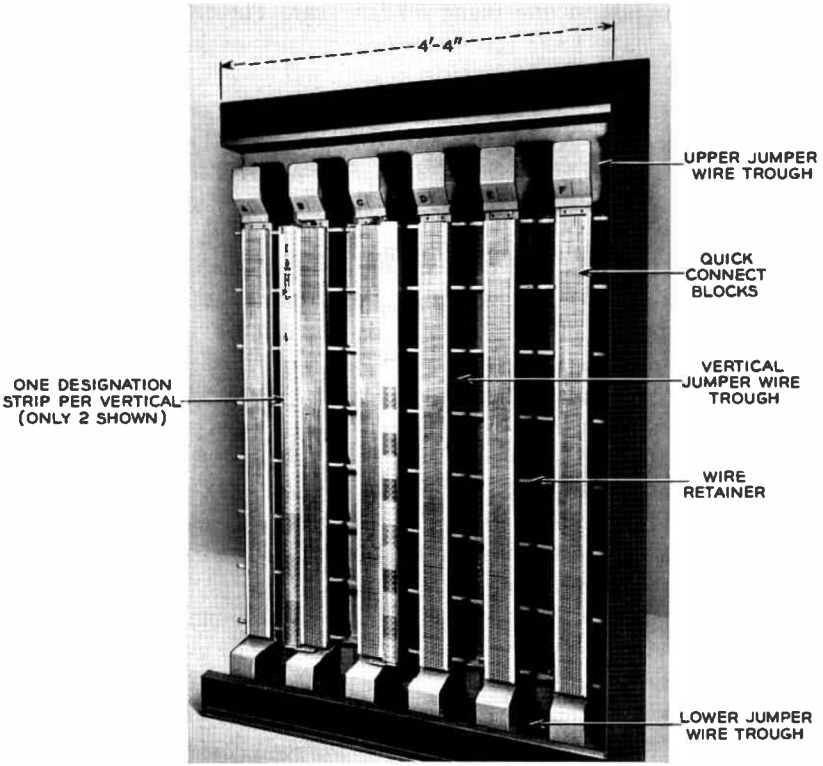


Fig. 50 — Trunk distributing frame.

side and system pairs and of the blocks of associated terminals in each. Blue and gray in alternate horizontal stripes identify the blocks of outside pairs; yellow and gray identify the system pairs.

8.3 *Trunk Distributing Frame*

A trunk distributing frame (TDF), shown in Fig. 50, fits in a frame line-up. It is similar to the MDF except that it has six verticals of 1216 terminal pairs each in a module 7 feet, 0 inch high and 4 feet, 4 inches long, and provides for only one connection on each quick-connect terminal. It terminates all cables from trunk link networks on the odd verticals and all circuits from universal and miscellaneous trunk frames on the even-numbered verticals to give short jumper cross connections between networks and trunks. This permits the traffic to trunk link networks to be balanced from time to time without recabling.

8.4 *Intermediate Distributing Frame*

The intermediate distributing frame (IDF) will normally be located at the head end of the MDF and in line with it, the two frames growing in opposite directions. Here the IDF will employ the same distributing frame module as the MDF. Occasionally, the IDF may be located in a frame line-up when shorter cable runs are needed to meet transmission limits. Here, the IDF employs the same module as the TDF.

In either case, the new IDF terminates all cables from equipment frames for trunk circuits, service circuits, and trunk auxiliaries (for carrier, repeaters, etc.) on connecting blocks of alternate verticals. Tie-cables from the MDF are interspersed on intervening verticals in a manner which permits most cross connections to be between adjacent verticals. Trunk access to protector pairs is thus multiplied by providing two jumpers (both usually short): one from the trunk to an MDF tie-cable appearance and one at the MDF between the protector and tie-cable terminals.

8.5 *Junctor Grouping Frame*

In No. 1 ESS, a new approach was developed for the distribution of junctors. In previous systems, junctors were cabled from the link frames to the junctor grouping frame (JGF), where they were interconnected directly or by jumpers in fixed patterns, depending on the number of line-link and trunk-link frames in the office. No 1 ESS uses plug-ended cords and jacks instead of jumpers. They are particularly attractive for additions, since junctors can be quickly redistributed by merely rearranging patch cords which have 16 pairs in each.

No. 1 ESS derives added benefits from this extremely flexible junctor redistribution. Junctor patterns in this system are a function of traffic type (intraoffice, interoffice, or tandem) as well as of office size, and this increases the number of patterns and the frequency of changes to balance traffic because: (a) Junctor circuits as well as network junctors are terminated on the junctor grouping frame, as shown schematically in Fig. 51. These circuits provide for the completion of intraoffice traffic without going through trunk-link frames. (b) Likewise, provision is also made on the junctor grouping frame for completing trunk-to-trunk traffic without going through the line-link frames.

The JGF shown in Fig. 52 fits in regular frame lineups under standard cable rack and is 2 feet, 2 inches wide. Frames are provided in pairs, one pair being required for each 16,000 lines in the average office.

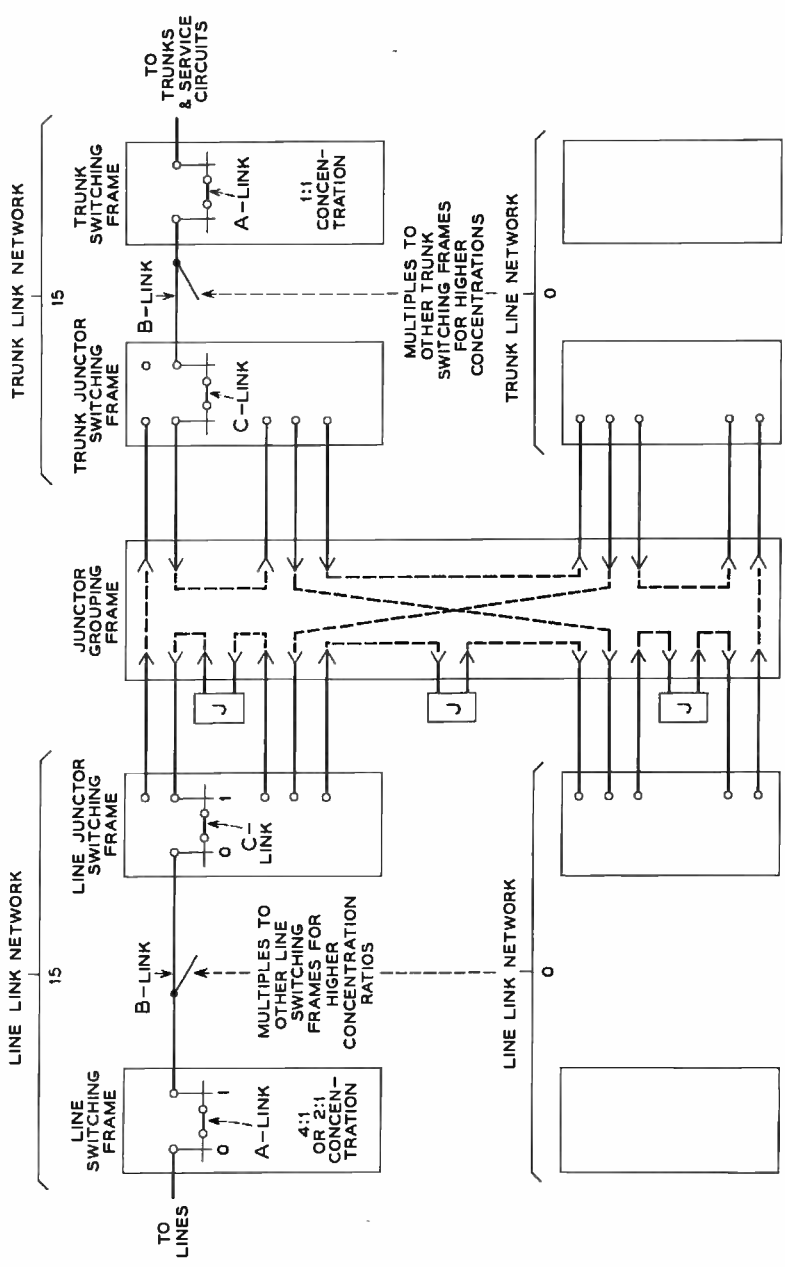


Fig. 51 — Typical connections for line link network and trunk link network.

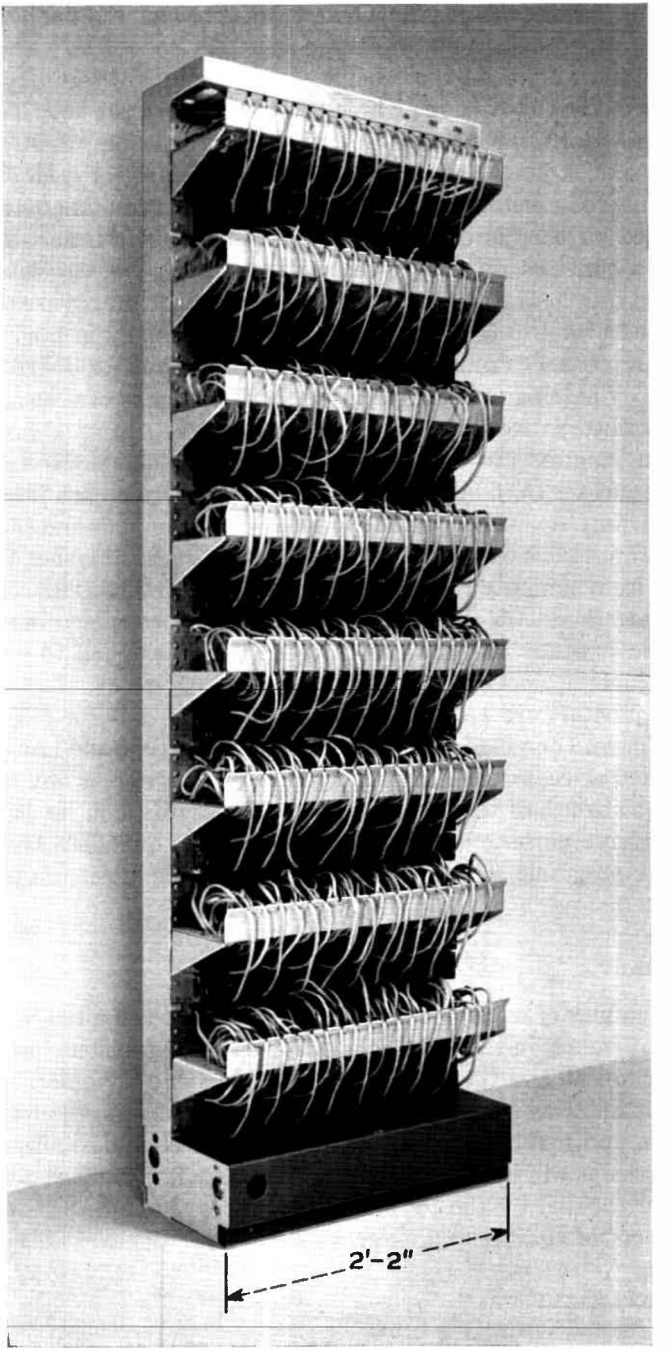


Fig. 52 — Junctor grouping frame.

On the front are eight cord shelves, each having eighteen 32-terminal plug-ended cords above and eighteen 32-terminal jacks below. The plugs and jacks on a frame are arranged in nine vertical files, each of which contains sixteen plugs and sixteen jacks, two of each for each shelf. All even-numbered junctors from one line-link or one trunk-link network are terminated on one file of an even-numbered grouping frame, and odd-numbered junctors on the corresponding file of the associated odd-numbered frame. The junctor circuits from a junctor frame are similarly distributed over two files; but being only half as many in number, two junctor frames are terminated on a pair of files. Junctor distribution is achieved by patching the plugs of one shelf into jacks of the same shelf in a prescribed pattern.

Sixteen junctors are terminated on each plug and sixteen on each jack, one junctor from each grid of a network (or from each junctor circuit subgroup). A slip is wired between the plug terminals and their associated terminations on the rear of the frame in such a manner that two junctors from similarly numbered grids or junctor subgroups are never coupled together. This slip permits all verticals to be cabled in identical fashion but spreads the junctors differently on each shelf to guarantee minimal junctor blocking.

When junctors are redistributed, all plugs of even- (or odd-) numbered frames on one shelf will be disengaged at one time and immediately reconnected as required. This will remove from service at any one time only one-sixteenth of the junctors from each network and junctor frame and thus have no serious effect on traffic if done outside a busy hour. Computer programs are being used to establish the optimum plug and jack patterns.

IX. SUMMARY

The wedding of a generic stored program with a limited variety of equipment frames in the No. 1 electronic switching system provides a new, economical, and versatile tool for the telephone customer. This system is introducing many new patterns in manufacture, installation, operation, and maintenance to the Bell System. The flexibility of this system concept will be demonstrated in the next few years as additional services are offered by the telephone companies through the continuing development of additional programs.

X. ACKNOWLEDGMENTS

As is the case in all complex systems, the apparatus and equipment described here are the result of the efforts of many people in several de-

partments of Bell Telephone Laboratories, particularly Systems Engineering, Systems Development, and Device Development. In addition, our associates in the engineering and production departments of the Western Electric Company have made many contributions.

REFERENCES

1. Keister, W., Ketchledge, R. W., and Vaughan, H. E., No. 1 ESS System Organization and Objectives, B.S.T.J., this issue, p. 1831.
2. Harr, J. A., Taylor, F. F., and Ulrich, W., Organization of No. 1 ESS Central Processor, B.S.T.J., this issue, p. 1845.
3. Downing, R. W., Nowak, J. S., and Tuomenoksa, L. S., No. 1 ESS Maintenance Plan, B.S.T.J., this issue, p. 1961.
4. Neumiller, E. G., The Electronic Central Office at Morris, Illinois, Bell Laboratories Record, **40**, Dec., 1962, pp. 402–407.
5. Genke, R. M., Harding, P. A., and Staehler, R. E., No. 1 ESS Call Store — A 0.2-Megabit Ferrite Sheet Memory, B.S.T.J., this issue, p. 2147.
6. Ault, C. F., Gallaher, L. E., Greenwood, T. S., and Koehler, D. C., No. 1 ESS Program Store, B.S.T.J., this issue, p. 2097.
7. Stammerjohn, L. W., An Evaluation of Design and Performance of the Permanent Magnet Twistor Memory, INTERMAG Conf. Proc., April 6–8, 1964, Washington, D. C.
8. Connell, J. B., Hussey, L. W., and Ketchledge, R. W., No. 1 ESS Bus System, B.S.T.J., this issue, p. 2021.
9. Chevalier, J. G., and Eisenhart, R. K., No. 1 ESS Circuit Packs and Connectors, B.S.T.J., this issue, p. 2441.
10. Pferd, W., Quick-Connect Clip Terminal, Bell Laboratories Record, **40**, June, 1962, pp. 202–207.
11. Cagle, W. B., Menne, R. S., Skinner, R. S., Staehler, R. E., and Underwood, M. D., No. 1 ESS Logic Circuits and Their Application to the Central Control, B.S.T.J., this issue, p. 2055.

No. 1 ESS Circuit Packs and Connectors

By J. G. CHEVALIER and R. K. EISENHART

(Manuscript received January 9, 1964)

The mechanical packaging design of any large electronic system can have a profound effect on the cost of the system and on its operating reliability. This article describes the No. 1 ESS packaging design and discusses the considerations which influenced it. Printed wiring packaging techniques are used for the individual circuit packs. A new connector and multiple board mounting, which allow considerable flexibility and a high degree of package density, were designed specifically for this application. A test program has established that the connector will perform reliably for a 40-year design life.

I. INTRODUCTION

The No. 1 ESS must compare favorably with existing switching systems in cost, reliability of operation, space requirements and ease of maintenance. The extent to which these basic objectives are achieved will depend to a considerable degree upon the mechanical packaging arrangement used for its electronic circuitry. The system must be packaged so that it can be mass produced economically and yet will operate with high reliability over a 40-year life.[1] Space requirements dictate compact packaging, although a high degree of miniaturization is not required. Once installed, any part of the system must be readily accessible for maintenance or replacement should the necessity arise.

At the present state of the packaging art, these objectives are best served with conventional printed circuit packaging techniques. The combination of a printed wiring board circuit pack and a molded wire spring connector is the basis of the packaging concept which was adopted for the No. 1 ESS.

II. CIRCUIT PACKS

Typical circuit packs are shown in Fig. 1. Each package measures approximately $3\frac{3}{4} \times 6\frac{7}{8}$ inches and can, where component sizes permit.

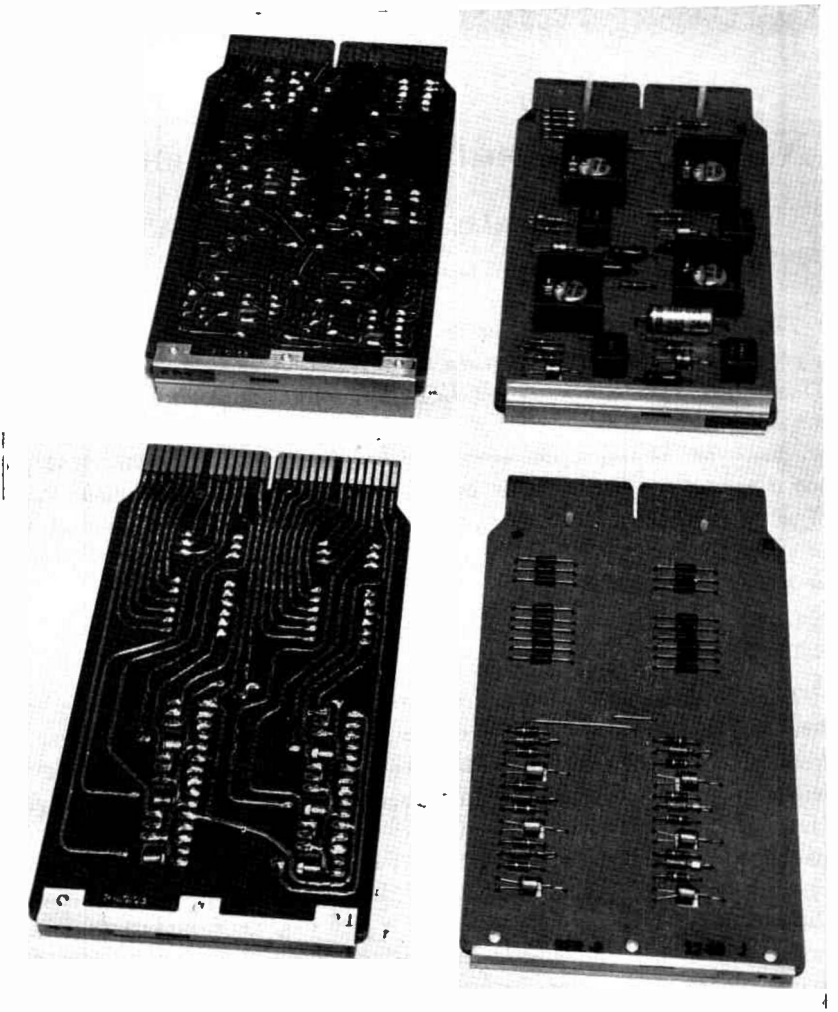


Fig. 1 — Typical No. 1 ESS circuit packs.

be mounted on 0.400-inch mounting centers. Twenty-eight contacts are provided on each circuit pack for supply voltages and interconnections with other circuit packs.

The printed wiring board is made from a fire-retardant grade of paper-based phenolic material. Its thickness is $\frac{3}{32}$ inch. In manufacture the board is blanked from sheet stock so that the grain of the laminate runs parallel to the $3\frac{3}{4}$-inch dimension of the board. Warp across the

board contacts is minimized by this orientation. Most of the board warpage will occur in a direction perpendicular to the grain, and this is in the long direction of the board. Experience has shown that the $\frac{3}{32}$-inch boards have sufficient rigidity so that a stiffening frame is not required to control the warpage.

All the circuit packs use single-sided circuitry: i.e., the printed wiring is confined to one side of the board. This practice avoids the potential reliability hazards of through-connections and also simplifies the manufacturing processes. Special means are occasionally necessary to permit one printed conductor to cross another. Wire straps are used for this purpose. Nominally, the conductors are 0.050 inch wide with minimum spacings of 0.050 inch. In a few instances it is necessary to decrease one or both of these dimensions below the nominal value because of space limitations on the printed wiring board. The printed wiring is terminated in 28 contact fingers at one end of the board. These fingers are 0.070 inch wide by 0.380 inch long and are plated with a wear-resistant gold to serve as the circuit pack contacts. The conductors elsewhere are coated with a 0.001-inch thickness of 50/50 solder applied in a roller coating operation.

Most of the components are of the axially-leaded type. They are mounted on the blank side of the board with their leads brought through holes and soldered to the printed wiring on the opposite side of the board. The leads of the transistor used in the logic circuitry are specifically arranged to facilitate mounting on the printed wiring board. The base and emitter leads are brought through glass seals at the bottom of the case, while the collector lead projects from the top of the case. The transistor is then mounted so that its body is suspended over and projects down into a square cut-out in the board. This minimizes the projection of the transistor body above the board and decreases the width required by the circuit pack. For manufacturing purposes all component lead holes are located by the coordinates of a grid pattern with 0.150-inch spacings in the long direction of the board and 0.250-inch spacings in the short direction. The circuit pack provides space for about 70 typical components, although one circuit pack has as few as 6 components and another as many as 84.

After component assembly and electrical testing, an acrylic lacquer is spray applied to the wiring side of the board except in the area of the contact fingers. This coating prevents the printed wiring and the insulating surface from coming into contact with contaminants such as dust, condensed moisture or fingerprints. The coating is thermoplastic and can be removed in the area of soldered joints when repairs are necessary.

As shown in Fig. 1, a short length of aluminum extrusion is riveted to the front end of the board. Code information identifying the circuit pack is printed on this strip. On the other end of the board near the contact fingers two nylon pegs are forced into holes in the board. These nylon spacers together with the identifier strip serve to support the circuit pack when it is laid component-side-down, thus protecting the components from damage.

In a typical 10,000-line No. 1 ESS office approximately 12,600 circuit packs will be required.[2,3] Thirteen per cent of these circuit packs will be identical, i.e., of the same circuit or code. Sixty-five per cent of the circuit packs will use only 17 codes. The entire system will require about 150 different codes. Component sizes dictate that about half of the 12,600 circuit packs can be mounted on 0.4-inch centers, while the rest must be mounted on 0.8-inch and larger centers.

III. CONNECTOR

A new connector was designed specifically to accommodate the No. 1 ESS circuit packs. Two departures from convention were incorporated into the design. First, the circuit pack contacts are provided on the board as an integral part of the printed wiring. It is not necessary to add a contact-carrying applique or plug unit to the board. Second, the connector uses single contacts instead of bifurcated or twin contacts.

Bifurcation is undoubtedly worthwhile in relays where contact action is intermittent and the major cause of contact failure is dust or other foreign particles. But printed circuit contacts are essentially static contacts, and contact trouble is more likely to be due to corrosion films which interfere with metal-to-metal contact. In this case, the two requirements most essential to reliability are an inert, pore-free contact finish and a high contact force. If these requirements are met bifurcation is unnecessary. If they are not, neither single nor bifurcated contacts will provide the 40-year reliability needed for the No. 1 ESS.

The contact springs of the connector are made from spring-tempered, Grade A phosphor-bronze wire. The wire is 0.036 inch in diameter and is solder coated for corrosion resistance. Twenty-eight of these wires are molded into a common phenolic strip to form the wire spring assembly shown in Fig. 2. The springs are divided into four groups of 7 springs each. Within a group the springs are located on 0.110-inch centers. The spacing between groups is somewhat greater to allow clearance space for rivets which must pass through the phenolic strip. One end of each spring is flattened and serrated for wire wrap connections, with

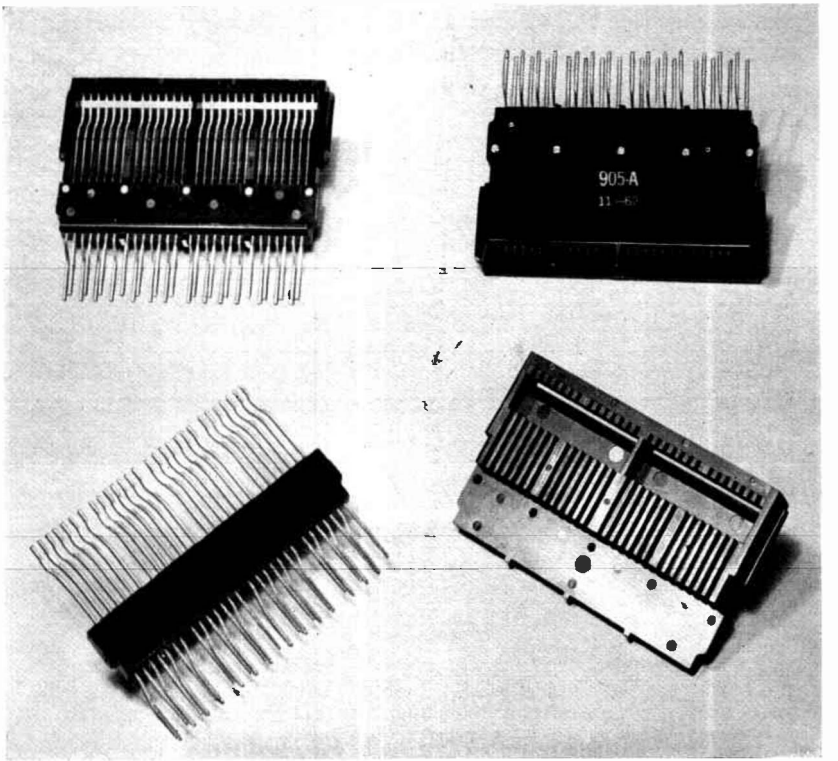


Fig. 2 — The connector assembly and the two main piece parts.

alternate terminals offset to allow access for the wire wrapping bit. The other end of each spring is formed to the correct configuration for the contact spring. A small contact button is welded to each spring at the point where contact will be established with the board.

The wire spring assembly is riveted to a second molded phenolic detail which is called the retainer. As the two parts are assembled the tips of the contact springs are forced against a surface of the retainer and are thus given an initial deflection or pretension. The retainer and the completed contact assembly are shown in Fig. 2. Fig. 3 shows in considerably more detail the pretensioning arrangement and the configuration of the springs. The tips of the springs are well protected from snagging. Even when the springs are fully deflected the tips will not project out of the contact assembly.

Fig. 3 also shows a magnified cross section of the teardrop-shaped contact button. The rolled overlay button consists of a 0.003-inch

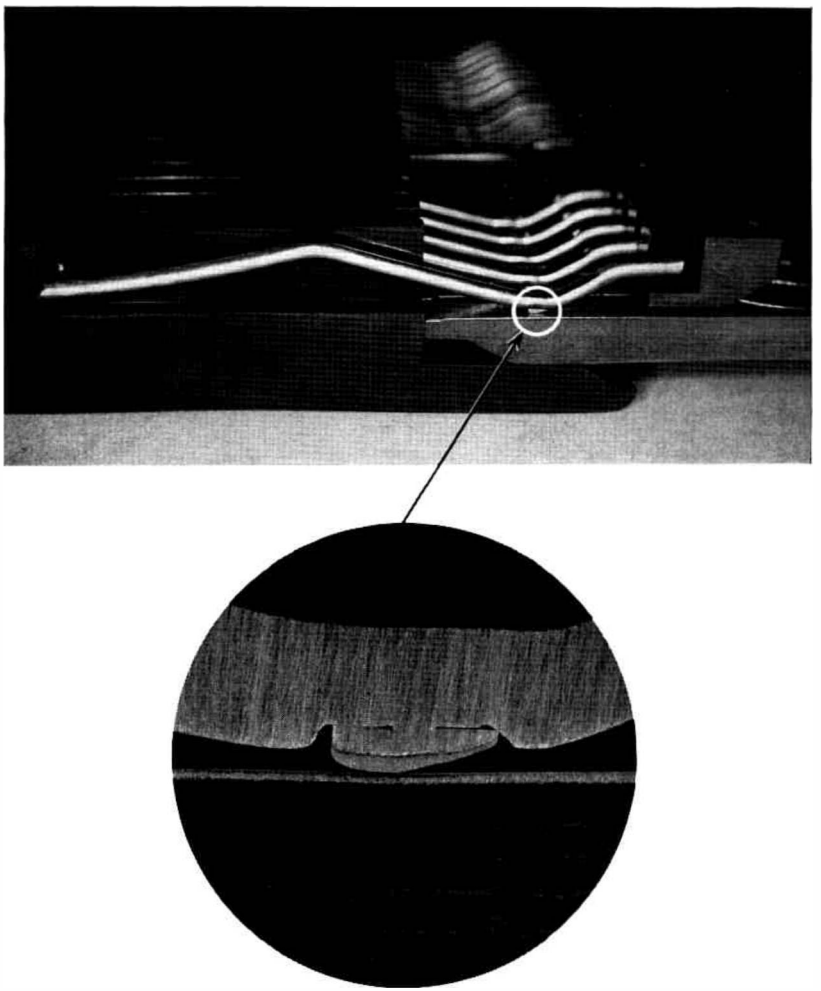


Fig. 3 — Connector spring and gold overlay contact cross section.

thickness of 24-karat gold and a base section of 80/20 copper-nickel alloy. The overlay material is supplied in tape form. It is cut to size and resistance welded to the contact springs in a continuous manufacturing process.

As a board is inserted into the connector, the board tongue enters a molded pocket in the retainer. Mating surfaces on the tongue and retainer pocket align the board so that the printed contact fingers engage and wipe against the spring contact buttons. Eventually the spring

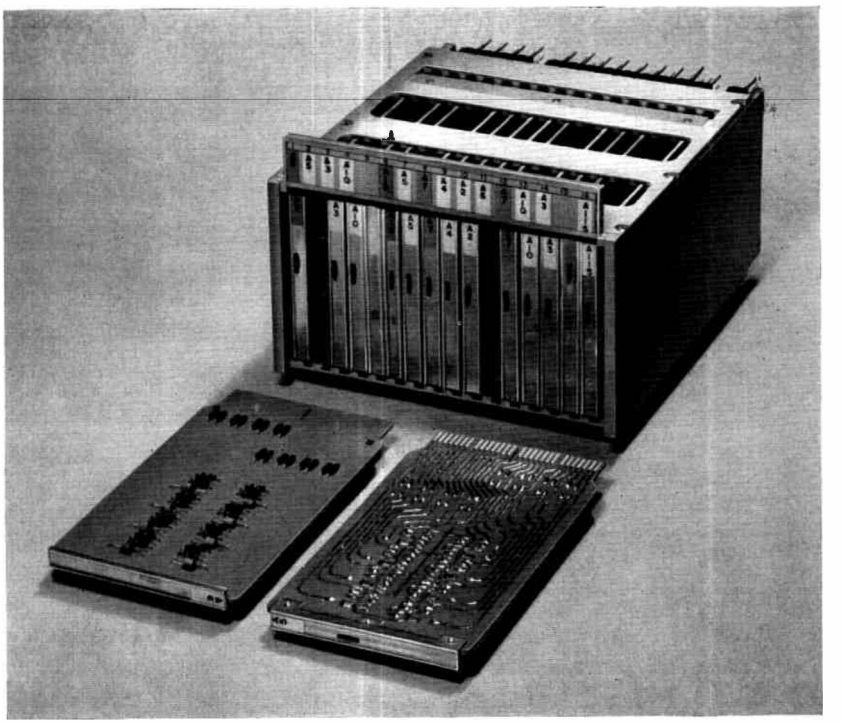


Fig. 4 — Circuit pack and connector mounting.

tips are lifted off the pretensioning surface, and the full contact force is applied to the board, forcing the tongue against the back of the retainer. The back surface of the board is specially contoured, as shown in Fig. 3, so that the contact force builds up gradually as the board is inserted. This minimizes the force required to insert the board.

IV. CIRCUIT PACK AND CONNECTOR MOUNTING

The mounting for the circuit packs and connectors is shown in Fig. 4. It is composed of upper and lower guide walls and two end plates which are assembled into an open box structure. All parts are aluminum die castings. Both of the guide walls and both end plates are identical parts, so that only one die is required for each. Two rectangular cutouts are provided in each of the guide walls to allow for the circulation of air between the circuit packs. The interior surfaces of the guide walls are lined with slots which are located on 0.400-inch centers. These slots locate the connectors and the circuit packs in the mounting. Molded

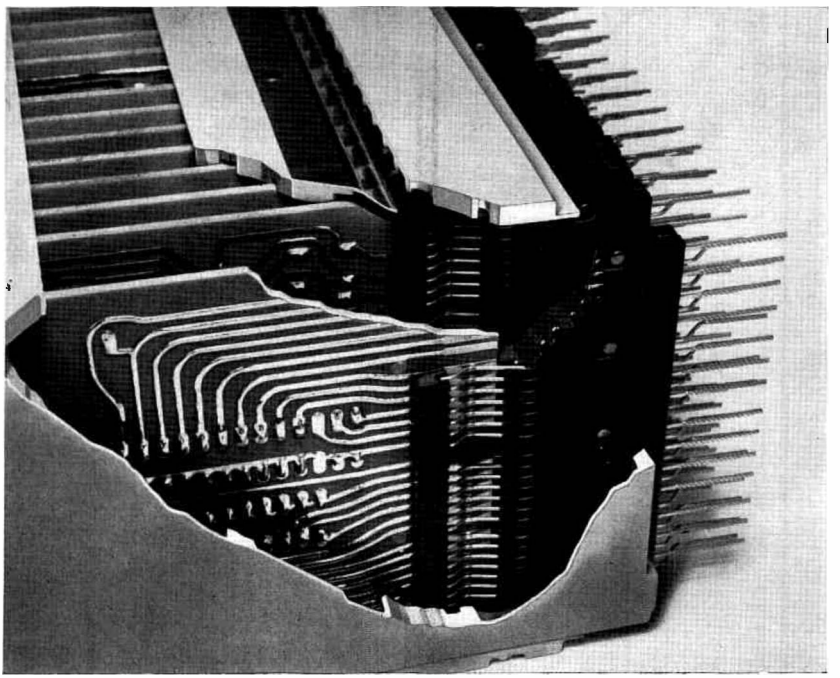


Fig. 5 — Cutaway mounting showing connector retaining latch.

keys at each end of the connector fit closely in the slots, which also align and guide the circuit pack into engagement with the contact springs. The slot entrances are funneled to facilitate entrance of the board. The connectors are retained in the mounting by stops in the slots and by spring latches which are riveted to each of the guide walls. As shown in Fig. 5, the latches are formed so that they project into the slot area. As the connectors are assembled in the mounting, the spring latches are deflected outward by the molded keys on the retainer. When the connector is completely inserted, the spring latches will snap back over the keys, thus locking the connector in place. A connector can be easily removed by deflecting the upper and lower latch springs past the point where they restrain the molded keys.

A designation strip is provided on the front of the mounting so that the circuit packs can be readily identified. The strip is rotated 180° to remove or insert packages. Package designations in the form of a letter-number combination and a color coding system are included on the designation strip and also on the circuit pack identifier. Correct location of a circuit pack is assured by matching the code number and

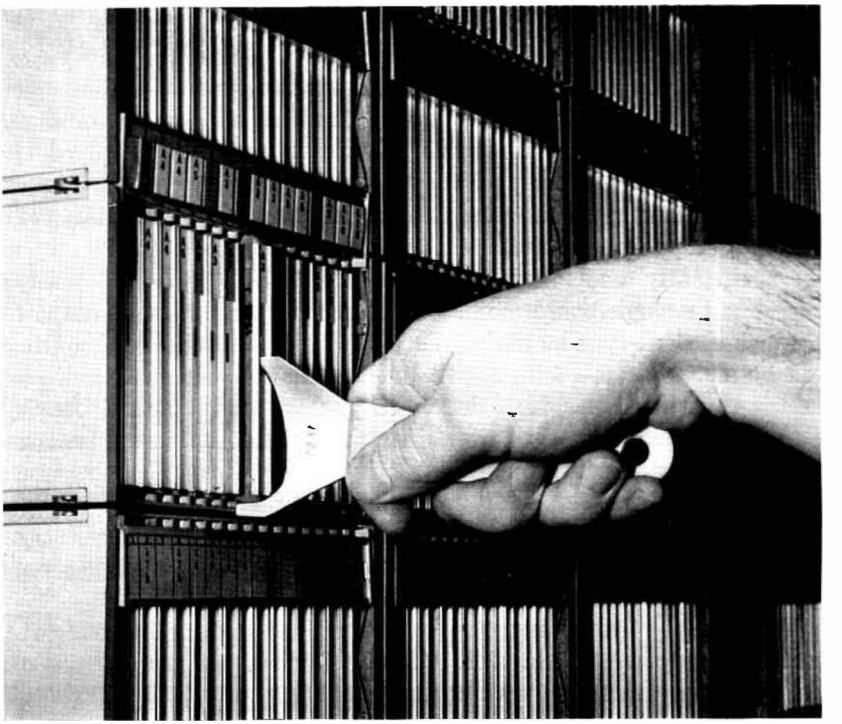


Fig. 6 — Circuit pack extractor tool.

color on both the designation strip and the package identifier. This method should provide adequate and reasonable safeguards against incorrect package insertion. The ultimate would of course be a mechanical coding scheme which would accept the correct code of circuit pack at a given location and make it physically impossible to insert any of the other 150 existing codes. This method was rejected as being prohibitive from the standpoints of cost, mechanical complexity and space consumption.

V. EXTRACTOR TOOL

Because of the close spacing of most circuit packs, a special tool is generally necessary to remove them. The extractor tool is shown in Fig. 6. The top prong of the tool is hooked into a slot in the circuit pack identifier and the bottom prong is rested against the lower guide wall. The board can be removed easily and in a controlled fashion by rotating the tool about the bottom pivot point.

VI. DESIGN ADVANTAGES

The new connector is quite flexible in that it permits the circuit pack spacings in the mounting to be varied as required by component sizes or other considerations. Four-tenths of an inch is the minimum spacing, but any integral multiple of this value is also possible. The maximum circuit pack capacity of the mounting can also be changed simply by varying the lengths of the guide walls. The basic connector and the end plates need not be changed. Actually, the No. 1 ESS mounting is currently available in two forms, the 16-board version shown in Fig. 4 and a single-board mounting. Either version can be mounted on a 4-inch channel-type mounting plate. Three of the 16-board mountings can be mounted on a 4-inch by 25-inch mounting plate. The new connector thus permits a high degree of package and terminal density. Terminal spacings are realized which approach the minimum practical for wire-wrapped terminations. A view of the wiring field is shown in Fig. 7. Three rectangular projections appear on the base of the retainer where the terminals are separated for rivets. These projections are painted white to accentuate the division of the terminals into groups of 7 and aid the craftsman in identifying terminals.

The design of the connector minimizes the undesirable effects of manufacturing tolerances. The pretensioning feature permits the use of a relatively compliant spring, thus reducing the variation in contact forces due to board thickness and other manufacturing tolerances. The alignment keys and the reference surfaces which locate the contact spring and the printed wiring board are all on one molded part, the retainer. The relative locations of these surfaces is thus tool controlled. In addition, the contact assemblies are located in the mounting in the same slots that are used to guide the printed wiring boards.

A significant advantage is achieved by the fact that the springs are molded in phenolic for a length of approximately $\frac{5}{8}$ inch. This feature provides a high degree of mechanical independence between the contact and terminal ends of the spring. Terminal or wiring disturbances are not expected to affect contact stability.

The welded gold button has several important advantages over the alternative possibility of an electroplated gold contact spring. The button provides a relatively thick layer of gold at the exact spot where it is needed at a cost which is less than that of electroplated gold. The wear resistance is thus optimized. Also, the possibility of a porous, corrosion-prone contact surface is eliminated, at least insofar as the contact spring is concerned.

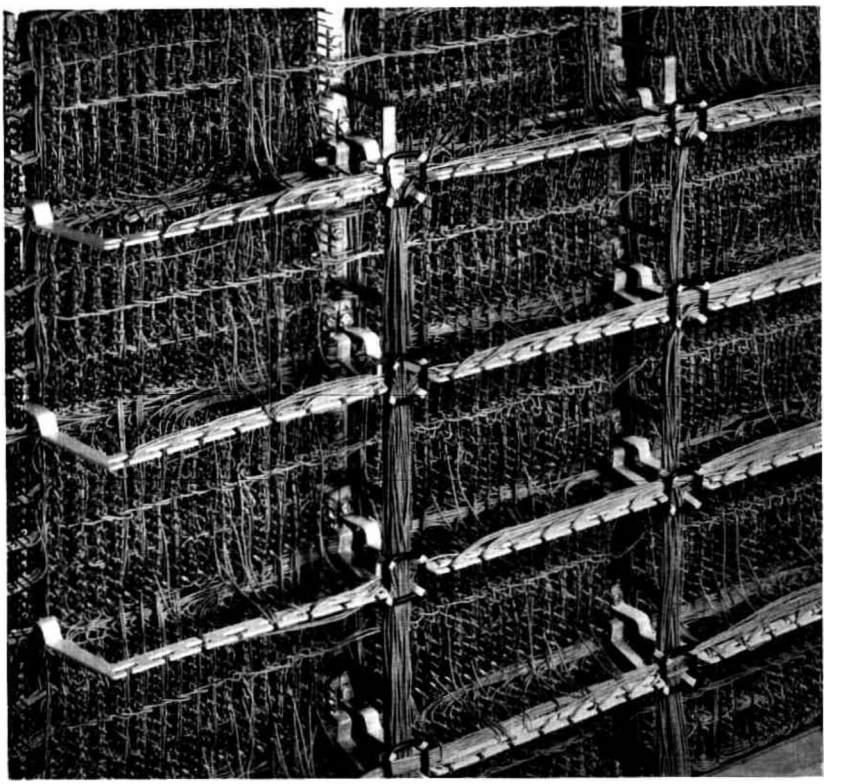


Fig. 7 — Connector wiring field.

The stepped contour which is machined on the back edge of the board facilitates board insertion. The contour delays the application of full contact force until the board fingers are well underneath the contact buttons. The buttons cannot wipe with appreciable force against the board insulation at the leading edge of the board fingers. There is thus no possibility of contaminating the contacts with wiped-on insulating material. This is a frequently overlooked source of contact contamination.

VII. TESTING PROGRAM

An extensive mechanical and electrical testing program was conducted to completely evaluate the performance of the connector.

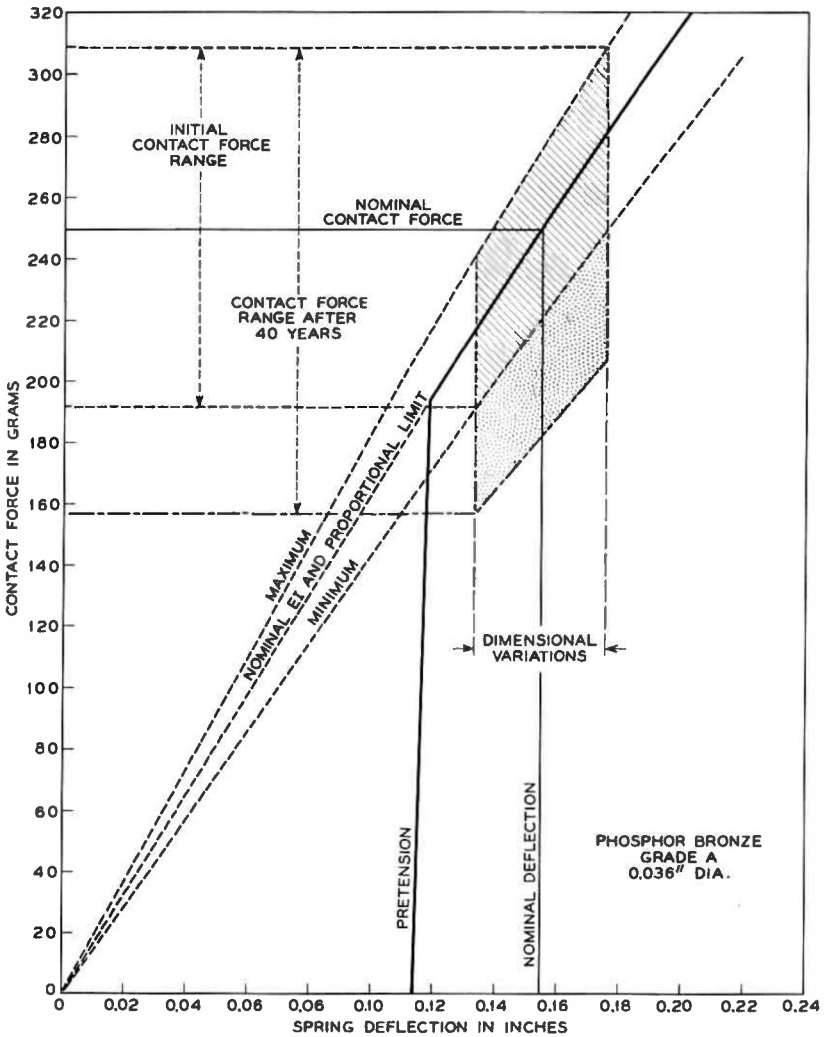


Fig. 8 — Connector spring characteristics.

7.1 *Mechanical Tests*

The force-deflection curve for the contact springs as obtained on the Instron Tester is shown in Fig. 8. Nominally, the springs are pretensioned with a force of 195 grams and will apply a contact force of 250 grams when an 0.093-inch thick board in inserted. There are many factors which can cause the contact force to assume a different value.

Variations in the spring properties of the wire or its size can, for a given deflection, appreciably affect the contact force. The extent of this effect is indicated by the maximum and minimum *EI* and proportional limit curves. The contact force will also be affected by dimensional variations in the molded connectors and the printed wiring boards. When all possible variations are considered and their effects accumulated arithmetically, the contact force is found to vary between 190 and 308 grams. These are initial values. There will be some degradation of the contact force with time, due principally to stress relaxation of the phosphor bronze wire. Stress relaxation studies conducted on the contact springs have indicated that over a 40-year period the contact force may diminish by about 15 per cent. Thus the minimum contact force could diminish to 157 grams in the unlikely event that all of the possible variations assumed the maximum unfavorable values.

7.2 *Contact Finish Tests — Corrosion Resistance*

A considerable part of the testing activity was pointed toward establishing the preferred contact finish for the printed wiring boards and connector springs. Good corrosion resistance is essential. The connector will be expected to provide a high degree of electrical contact stability in all of the wide variety of environments in which it might be located. The contact finishes must also have sufficient wear resistance to withstand the number of contact wipes which could occur over a 40-year life. A wear life of 200 insertions and withdrawals was adopted as the design requirement for the board contact finish and 500 insertions and withdrawals for the contact finish of the connector.

The corrosion resistance of a number of contact finishes was evaluated in laboratory tests and in outdoor exposures at Columbus, Ohio, New York City, and Kure Beach, North Carolina. Tin, solder and 24-karat gold were the contact finishes investigated.

Tin and solder were included in the hope that one or both of them might prove to be a low-cost contact finish which would be suitable for use with the No. 1 ESS connector. Both are soft metals which form hard and brittle tarnish films. Generally such films are easily fractured by the application of a force sufficient to cause deformation of the underlying metal. It was felt that the relatively high forces of the ESS connector and the plowing action of wiping contacts would be sufficient to disrupt all tarnish films, so that clean metal-to-metal contact could be established. While this condition was essentially realized, there was one important side effect. The residue of disrupted tarnish films proved to be a source of contact contamination. This residue, composed of

minute fragments of broken films, is created with each contact wipe in which a tarnish film is ruptured. With successive insertions in certain environments the debris can accumulate on the contact surfaces to the point where it interferes with metallic contact, thus causing high and unstable contact resistance. In environments of high humidity this point is reached in just a few insertions. On the basis of this behavior tin and solder were judged unacceptable for the ESS application. Gold, on the other hand, performed well in the environmental tests and satisfied the contact stability requirement in every respect.

The environmental tests established the advisability of restricting the circulation of air past the contacts. Contacts so protected were consistently less affected by corrosive atmospheres than unprotected contacts. When the circuit packs are spaced on 0.400-inch centers, the close spacing of the connectors affords sufficient protection against air circulation. For larger spacings, a cover is available which covers the open side of the connector and effectively restricts air flow past the contacts.

7.3 *Contact Finish Tests — Wear Resistance*

Although the corrosion resistance of 24-karat gold is excellent, its wear resistance was found to be quite poor. A 0.0001-inch thickness on the printed wiring board contacts would frequently be worn through in less than 10 insertions into the connector. Alloy golds with much better wear resistance are commercially available. These finishes are electrodeposited as alloys which generally contain less than 1 per cent of either cobalt or nickel. Their corrosion resistance is comparable to that of pure gold. A cobalt-gold alloy with a Knoop hardness of 160 proved to be the most suitable for the contact finish of the ESS circuit packs. A minimum thickness of 0.0001 inch is specified. The contact finish of the spring is provided by the pure gold overlay of the welded button.

With this finish combination the following type of contact wear occurs. As the soft gold button comes into contact with the hard gold board finish, cold welds will be established between the two surfaces at some of the several points of contact. These welded junctions will be broken and others established and subsequently broken as the two surfaces are wiped together. Some of the welds will be weak at the interface of the two contact metals and failure will occur there. Other welds will be strong at the interface and failure will occur sometimes within the bulk of the soft gold button and sometimes within the bulk of the board conductor copper. In the first case, where failure occurs within the gold,

a fragment of the pure gold will be left welded to the hard gold finish on the board. When failure occurs within the copper conductor, a small fragment of the hard gold finish will be plucked out of the board and left welded to the button. In either case both contact surfaces are roughened and surface damage will progress rapidly with successive insertions. This welding and plucking type of surface damage is typical of that which generally occurs when soft metal is slid on a hard metal.

The amount of surface damage depends to a remarkable degree upon the type and thickness of the contaminant films which are on the contact surfaces. Certain films which form naturally, such as water vapor or adsorbed gas films, are inevitable on normal contact surfaces and can provide a surprising amount of lubrication. The degree of lubrication, however, is generally inadequate for the wear requirements of 200 cycles on the board and 500 cycles on the connector. Additional contact lubrication is necessary if these requirements are to be met consistently.

A contact lubricant, in addition to minimizing surface damage to the contact finishes, must not degrade the contact resistance. The lubricated surfaces must not be sticky or they will become contaminated with dust and other particulate matter. The lubricant must also stay on the contact surface where it is applied. It should not creep or migrate even in warm environments. Tests have established that these requirements are best satisfied for No. 1 ESS purposes with a thin film of one of the microcrystalline waxes. The lubricant is applied from a dilute trichloroethylene solution (0.5 per cent of wax by weight) to the contact surface of the board. It can be applied either by dipping or spraying.

VIII. CONCLUSIONS

It is concluded from the results of the testing program that the ESS connector in its present form will perform reliably for the 40-year design life. There were no indications that its performance would be improved by adding a contact-carrying plug unit to the board instead of using the board conductors as contact fingers. There was similarly no indication that the design should be modified to include contact bifurcation.

IX. ACKNOWLEDGMENTS

The developments described here represent the combined efforts of many people in Bell Telephone Laboratories and the Columbus Works of the Western Electric Company. The authors wish to express their appreciation for all these contributions. The outstanding work of a few

should be given special recognition. The late J. H. Mogler made substantial contributions to the initial connector design, while J. A. Bachman and M. T. Skubiak have contributed significantly to the present design.

REFERENCES

1. Downing, R. W., Nowak, J. S., and Tuomenoksa, L. S., No. 1 ESS Maintenance Plan, B.S.T.J., this issue, p. 1961.
2. Cagle, W. B., Menne, R. S., Skinner, R. S., Staehler, R. E., and Underwood, M. D., No. 1 ESS Logic Circuits and Their Application to the Central Control, B.S.T.J., this issue, p. 2055.
3. Ferguson, J. G., Grutzner, W. E., Koehler, D. C., Skinner, R. S., Skubiak, M. T., and Wetherell, D. H., No. 1 ESS Apparatus and Equipment, B.S.T.J., this issue, p. 2355.

PROCESS III — A Compiler-Assembler for No. 1 ESS

By N. A. MARTELLOTTO, H. OEHRING and M. C. PAULL

(Manuscript received January 20, 1964)

A description of a compiler-assembler program named PROCESS III is given. This program is used to translate No. 1 ESS symbolic source programs to No. 1 ESS binary object programs. Included is a discussion of the PROCESS language, the Compool and macro facilities of the compiler, and some requirements that motivated its design.

I. INTRODUCTION

No. 1 ESS is a stored-program control telephone switching system.[1,2,3] The program consists of more than 100,000 instructions, each 44 bits in length, made up of 37 information bits and 7 check bits. To write such a program in binary (or octal) machine language is clearly impractical. To efficiently produce such large programs, modern techniques include the use of mnemonics or a symbolic language by the programmer. We say the programmer writes a *source* program in some kind of symbolic language, whereas the central control[2] executes an *object* program in its machine (binary) language.

This article mainly describes the vehicle that translates No. 1 ESS source programs to No. 1 ESS object programs. Certain other items explain the progress of a No. 1 ESS program from inception as a source program to its final state as an object program.

The vehicle developed for translating from No. 1 ESS source programs to No. 1 ESS object programs is itself a program; this program, named PROCESS III,* is executed on the IBM 7094 general-purpose computer. In keeping with the current usage for the words "compiler" and "assembler," PROCESS III is said to be a "compiler-assembler," since it performs both functions, as will be described. Consequently, in this article the words "compiler" and "assembler" are used interchangeably unless otherwise noted.

* PROCESS is an acronym for PROgram to Compile ESS programs.

1.1 *Background*

A compiler should come early in the development of a stored program system. The sooner one can translate to the object program, the sooner interpretive simulation may be undertaken; in turn, this means one may begin to feel confident sooner about such things as adequacy of order structure, and construction and size of the object program. PROCESS III was started early; although there were some definite ideas initially as to what some of its requirements would be, it was necessary to build a flexible structure to accommodate the many new requirements that would arise as the development of the No. 1 ESS program progressed.

Early in the development of No. 1 ESS, it was recognized that the object program would be large, and that it would be written, compiled, and tested in small sections of 1000 to 5000 instructions. Also known was the desirability of not having to disturb all parts of this large program in the semipermanent twistor memory when small sections were being tested and recompiled. Yet the sections had to communicate with each other and transfer control to each other. These considerations led to one basic design criterion for PROCESS III: while still insuring that communication and control among sections remain intact, it must be possible to recompile and reinsert sections of the total object program without disturbing the whole.

A more fundamental design criterion for the compiler arose from a lack of knowledge of the precise nature of the source program. It was known, of course, that the program was to do data processing in connection with telephone calls. Such actions are difficult to describe completely in a simple or mathematical or uniform way. That is to say, there did not exist a "telephone language" that could be used to describe completely the telephone data processing functions. However, it was known that some telephone functions are amenable to simple, mathematical, uniform description; furthermore, as learning took place, it was hoped that the balance of the telephone functions could in time be described in a straightforward manner. Thus, a fundamental design criterion for PROCESS III became flexibility: not only must the compiler be able to handle known straightforward descriptions of telephone functions, but it must also be capable of being used to construct, accept, and retain new descriptions of telephone functions.

The requirements of independent compilation, integrity of communication among sections of object programs, and flexibility have been met by PROCESS III because it:

(1) normally produces relocatable object programs,

(2) has a communications pool (Compool) facility, and

(3) has a powerful macro facility.

A relocatable object program has each of its instructions assigned an address relative to zero, the address of its first instruction. The address field of each No. 1 ESS instruction in a given program section P may in turn refer to one of four kinds of numbers:

(a) constants; these are said to be absolute numbers, since they will not be altered by subsequent processing;

(b) a local program point, in which case the number is a relocatable address within the range of the object program instruction addresses of P;

(c) a global program point, in which case the number is a pseudo-address that refers to a program other than P and therefore cannot be assigned explicity at the time P is compiled; and

(d) fixed call store or program store locations; these numbers are absolute addresses outside the program P but within the range of the two memories.

The total No. 1 ESS object program consists of many relocatable sections like P. Each section contains various numbers, as described. An immediate need for proper execution of the total object program by central control is a loading scheme for inserting this program into the twistor program store. The scheme is implemented by another IBM 7094 program called a "loader." The loader accepts as inputs many object programs generated by PROCESS III and produces as output a single unified (linked) object program containing only absolute addresses. Thus, having determined where sections of the object program will reside in the twistor store, the relocatable program points of (b) and the pseudo-addresses of (c) are changed by the loader to absolute addresses. The absolute numbers of (a) and (d) are not altered. The loader also has the ability to accept a more current version of a section (or sections) of the object program without disturbing the remaining sections. This means that fewer twistor cards need to be remagnetized during the checkout and debugging phases. Finally, the loader will generate and prefix 7 check bits to each 37-bit instruction. Since the Hamming code for these bits is a function of the 37 information bits *and* the absolute address at which the 37 bits reside in the twistor store, they cannot be generated prior to load time.

The Compool facility of PROCESS III enables one to refer conveniently to addresses of type (d). A major consideration in the design of the total object program is the temporary (call store) memory configuration. Temporary memory must be assigned to the call registers,

the network map, various queues, and so on.[3,4] Since there are many programmers involved in constructing the large No. 1 ESS object program, it is efficient to centralize the assignment of temporary memory. The Compool of PROCESS III provides this centralization. Having once defined call store areas in the Compool, the many individual programmers need not concern themselves about such areas except to refer to them as necessary. In their programs, reference to these areas must be made symbolically, but there is no need for individual definition of such areas; the communications problem among programmers using the same call store areas is thereby substantially reduced by the Compool facility.

The Compool, therefore, is a collection of symbols that are assigned call store addresses; these symbols correspond to areas set aside in memory in a particular configuration for the purpose of doing telephone data processing. Of course, the configuration may change as the development of the object program progresses. In this case, PROCESS III is used to update the Compool, and the new configuration is used in compiling all source programs thereafter. It may be necessary for source programs to be recompiled, depending on the change in call store configuration. This too is partially automated: when a Compool run (i.e., a change in call store configuration) is made, PROCESS III refers to its "bookkeeping file" to ascertain which source programs, if any, need to be recompiled. A list is printed out and the individual programmers are notified. They need only recompile without concerning themselves as to precisely what call store changes have been made.

It was mentioned that PROCESS III has a powerful macro facility. In this context, a macro is defined to be a fixed amount of code, in some language, that will result in a variable amount of object program when it is properly "called." Just as basic instructions have variable fields (e.g., address, index, masking),[2] so do macros; in the case of macros, the variable fields are called "parameters." Since calls occur in the source program, the names of the macros, or more simply the macros, are said to be part of the source language. If one is given the ability to define his own macros he may thereby extend the source language. A set of macros which has been incorporated permanently into PROCESS III constitutes just such an extension. These permanent macros are saved as a special part of the Compool so that they may be refined and extended still further as more is learned about telephone data processing functions. In addition, individual programmers may define and call their own macros.

The balance of this paper is devoted primarily to three things:

(1) an explanation of how call store configurations are defined,

(2) a description of the source language and some of its extensions, and

(3) a discussion of the tools available to construct extensions to the source language.

II. STORAGE ALLOCATION

The design of the compiler was influenced by the real-time and space-limited nature of the resulting object program on the one hand, and the demands of an intricate basic machine order structure on the other. An obvious solution to the space problem was to pack information into subunits smaller than the natural dimensions of the available memory units (37 binary bits in the program store and 23 bits in the call store). Thus it was desirable to provide means in the compiler for naming such subunits, called "items," in memory.

The object program organization[3] itself demanded of the compiler the ability to define several types of homogeneous blocks of temporary memory, each composed of several basic memory units. A group of call registers[3,4] is essentially a group of similar blocks of memory, where each word within one block serves the same functions as the corresponding word in all the other blocks. For example, the high-order bit of the second word in each block may indicate whether this call register is controlling a telephone call or not. Memory blocks of this call register type are called "scatter tables" and are defined in PROCESS III by the following statement:

```
(1)  XX      SCATABLE   N1,N2
```

Beginning at an address called XX, statement (1) reserves space in memory for N1 tables each containing N2 words.

The statement,

```
(2)  YY      TABLE      N1,N2
```

defines N1 tables each containing N2 words. Memory defined with TABLE differs from that defined with SCATABLE; all the words in any one of the N1 tables defined with TABLE have the same function, but the function may vary from table to table. For example, with N1 = 2, the rightmost 17 bits of each word in the first table might be used to store the line equipment number, while the second table might consist entirely of trunk equipment numbers in the rightmost 15 bits.

It is also convenient to be able to define one continuous block of storage. The compiler accepts statements like:

```
(3)  ZZ      BLOCK     N
```

This reserves N words of space in memory and names the address of the first word ZZ.

In order to refer to memory items smaller than the basic word by name, one must be able to define them. The statements below serve this purpose:

```
(4)  YY      TABLE     10,50
(5)  YY0     LAYOUT    ABB----------CCCCCCCC
(6)  IT1     ITEM      A
(7)  IT2     ITEM      B
(8)  IT3     ITEM      C
```

Statement (5) lays out all 50 words of the first table in the set of 10 tables defined by (4) in the call store area. The high-order bit is defined as one item, the next 2 bits as a second item, the dashes indicate bits not being defined, and the rightmost 8 bits constitute a third item. Statements (6), (7), and (8) assign names to the three items.
Thus IT1 is the name assigned to the item indicated by A in the LAYOUT statement, IT2 is the name of the B item, and IT3 the name of the C item. In a similar way items may be defined for SCATABLE and BLOCK.

The following characteristics of items are useful in manipulating them, as will become obvious in the following sections.

```
A.ITEM = the address of the item
M.ITEM = the mask of the item; i.e., a 23-bit constant with all ones
         in the position of the item and zeroes elsewhere
D.ITEM = the displacement of the item from the right
S.ITEM = the size of the item; i.e., the number of bits contained in
         the item.
```

The qualifiers A., M., D., and S. are prefixed to the name of the item to refer to these characteristics. For example, the items defined in statements (6), (7), and (8) have the following characteristics:

```
A.IT1 = A.IT2 = A.IT3 = address named YY0
M.IT1 = O.20000000*
```

* PROCESS III assumes integers to be decimal unless qualified by O. to indicate they are octal.

```
M.IT2 = 0.14000000
M.IT3 = 0.377
D.IT1 = 22
D.IT2 = 20
D.IT3 = 0
S.IT1 = 1
S.IT2 = 2
S.IT3 = 8
```

III. SOURCE LANGUAGE

3.1 *Basic Orders*

Some of the storage allocation or storage defining facilities of PROCESS III were tailored for use by the No. 1 ESS basic orders. Since the nature and aims of the basic order structure have been discussed in some detail in an earlier article,[2] only some of their pertinent characteristics in connection with the use of item qualifiers are illustrated here.

```
(9)        MK        YY0
```

This instruction moves the 23-bit contents of the word named YY0 into the K register. The following instruction (10) does exactly the same thing:

```
(10)       MK        A.IT1
```

The triplet below [instruction (11)] sets the logic register to the mask of IT2, moves the contents of the item IT2 into the K register, masking out everything but the item by logical product (PL), and then right adjusts it in the K register by using the displacement of the item.

```
(11)       WL        M.IT2
           MK        A.IT2,,PL
           HC        D.IT2
```

The same actions are accomplished in a different way by the next three instructions:

```
(12)       WX        A.IT2
           MK        M.IT2,X,PS
           HC        D.IT2
```

Because of the PS option the mask is set up and used in the same order.

Given a source program input such as the basic order instructions in (9), (10), (11), or (12), PROCESS III will produce an object program as output with the property that for each input instruction there will be exactly one output instruction. This one-to-one correspondence between input source program instructions and output object program instructions defines an assembling process. On the other hand, a compiling process is defined to be a one-to-many correspondence between input source program "instructions" and output object program instructions. For a compiler the input "instructions" are called "statements," and the set of all statements meaningful to the compiler is called a "language." PROCESS III will accept as inputs basic order instructions and statements of its own language, called PROCESS, in any mixture.* The source language of No. 1 ESS programs, therefore, is nominally PROCESS plus the basic order instructions. It will be shown that the PROCESS language can be extended.

3.2 *PROCESS Language*

The basic goal of the PROCESS language is to provide the No. 1 ESS programmer with a means to write his source programs quickly and efficiently. As with any higher-level programming language, a program written in PROCESS cannot be better in terms of object program length than the same program written with basic orders by an *expert programmer*. When used properly, however, the PROCESS language gives object programs that are no worse than the great majority of those written with basic orders by average programmers.

The PROCESS language is intended to provide the fundamental programming tools needed to write telephone programs. The functions required in any program, including telephone programs, can be classified into three categories:

(1) moving data from one place to another,
(2) making decisions using these data, and
(3) performing arithmetic or logic operations on these data.

While these requirements were predetermined, there were some additional telephone-oriented programming problems that evolved as the programming effort got under way. These problems were solved readily because of the flexible macro facility of PROCESS III and the resulting ease in extending the source language by defining new procedures.

Essentially, the PROCESS language consists of a set of procedures

* Hence the designation of PROCESS III as a "compiler-assembler."

that can be called by the programmer with varying input information, called "parameters." A description of the allowable parameters and procedures follows.

The nature of parameters may be discussed on two levels:

(*i*) On the lower level, parameters can assume the identity of any of the basic units of data handled by the basic orders. The various possibilities are:

(a) full memory words, indexed or not
(b) items or partial words, indexed or not
(c) constants
(d) central control registers.

For (a) and (b), indexing is specified by enclosing the address and index in parentheses, as (address, index).

(*ii*) Parameters can also assume a more general nature that enables a programmer to nest procedures. In this case a parameter can be represented by:

$$\mathrm{OP}(a_1 , a_2 , \cdots , a_n)$$

Here the operator OP of the parameter can be any of the basic procedures of the language or the character C; the use of C indicates that the body of the parameter $(\cdots)$ should be complemented. Each a_i again can be of the form $\mathrm{OP}(b_1 , \cdots , b_m)$ or one of the forms defined by (a) through (d) above.

The general-purpose procedures of the PROCESS language are:

1. Data moving

MOVE OP(x),b,c,d,$\cdots$

Function: Move the quantity specified by x to the destinations designated by b, c, d, and so on. If OP is a basic procedure, perform this operation on x before moving the result to b and c, etc.

2. Decision making

(a) IF OP(x),($r_1 , r_2 , \cdots$),OP(b),($d_1 , d_2 , \cdots$)

Function: If the quantity x or the result of the operation (if any) performed on x has the relation r_i to b or to the result of the operation (if any) performed on b, then control is transferred to a program location named d_i (i = 1,2,3). The allowable relations r_i and their meanings are:

E = arithmetically equal
NE = arithmetically not equal

LE = arithmetically less than or equal to
GE = arithmetically greater than or equal to
L = arithmetically less than
G = arithmetically greater than
XE = logically equal
XU = logically unequal.

(b) COMP $OP(x),r,(b_1, b_2, \cdots, b_n),$
$(d_1, d_2, \cdots, d_n)$

Function: If the quantity x or the result of the operation performed on x has the relation r to b_i, program control is transferred to d_i ($i = 1, 2, \cdots, n$).

(c) IFOR $OP(x),r,(b_1, b_2, \cdots b_n),d$

Function: If the quantity x or the result of the operation performed on x has the relation r to any one of quantities b_i ($i = 1, \cdots, n$), program control is transferred to d.

(d) IF $x,r,b,OP_1(c_1),\cdots,OP_n(c_n),$
$ELSE(OP_{n+1}(c_{n+1}),\cdots,OP_m(c_m))$

Function: If x has the relation r to the quantity b, then perform the procedures $OP_1, \cdots, OP_n$; if not, then do $OP_{n+1}, \cdots, OP_m$.

(e) IF $x,r,b,OP_1(c_1),\cdots,OP_n(c_n),$
$ALSO(OP_{n+1}(c_{n+1}),\cdots,OP_m(c_m))$

Function: If x has the relation r to the quantity b then perform the procedures $OP_1, \cdots, OP_n$; in any case then do the procedures $OP_{n+1}, \cdots, OP_m$.

3. Arithmetic and logic procedures

(a) SUM $OP(x),OP(y),b,c,\cdots$

(b) DIFF $OP(x),OP(y),b,c,\cdots$

(c) AND $OP(x),OP(y),b,c,\cdots$

(d) OR $OP(x),OP(y),b,c,\cdots$

(e) EXOR $OP(x),OP(y),b,c,\cdots$

Function: Perform the indicated operation on the parameters x and y after executing the function of the operators on x and y, if any, and put the result into b, c, etc.

4. Loop control

```
m       LOOP        i,f,c,v
        ⋮
m       ENDLOOP
```

Function: These two statements control a loop that begins with the call of LOOP and ends with the call of ENDLOOP which specifies the same m (name) in the location field. i and f are the initial and final values of the loop variable. At the end of each pass through the loop, the loop variable is incremented by c. When this new value exceeds f, control passes to the next instruction outside the loop; otherwise control is transferred to the beginning of the loop. v specifies the loop variable to be used. If v is not specified, the central control register Z will be used as a loop variable. It is possible to nest loops within loops. If the same loop variable is specified for more than one loop, the value of that variable is saved and reset when entering and leaving another loop.

5. Initialization facility

```
        INIT        b,c,···
```

Function: Place c and any following parameters into consecutive locations starting with the location specified by b.

6. Unconditional transfer of program control

```
        GO*TO       OP(x),(b1 , b2, ··· , bn)
```

Function: At execution time, x or the result of the operation performed on x, if any, specifies a number i, and program control is transferred to b_i. If b_i is not specified, program control is transferred to x.

Some typical calls for these procedures are:

```
START   LOOP        1,10,1
        MOVE        (ITEM,X),FULL
        IF          (ITEM,Y),E,0,DEST
        IF          (0,Y),E,L,MOVE(0,(0,Y)),
                      ELSE(MOVE(L,(0,Y)))
DEST    SUM         SUM(1,(ITEM,X)),K,(0,Z)
START   ENDLOOP
        GO*TO       SUM(Y,(ITEM,Z)),(D1,D2,D3)
```

The procedures described constitute the initial general-purpose subset of the PROCESS language. A realistic example showing the use of many of these procedures is given in Appendix A.1.

As the needs of the No. 1 ESS program became clearer, special telephone-oriented procedures were added to this initial set to extend the source language. For instance there are procedures to implement a change in network (CIN), a change in peripheral circuit configuration (CIC) or signal distributor (SD) actions. These procedures have enabled the programmer to implement such functions in a higher-level and more descriptive language, thereby relieving him of details involved in writing source programs at the basic order level.

Procedures in the PROCESS language are in fact macro calls. The corresponding macro definitions for the language are retained by the compiler in its Compool. To extend the source language, one defines new macros. The extensions may be global or local. Global extensions to the source language are macros that have proven to be of widespread use among the programmers; these are entered into the Compool and become part of the PROCESS language, capable of being used thereafter by all programmers. Local extensions to the language are macros that are defined and called by individual programmers in their own programs. Obviously, there may be many different kinds of local extensions to the source language. Extensions, whether global or local, are constructed using special macro orders in the macro definitions.

IV. MACRO DEFINITIONS

Each macro definition is associated with a definite name (or set of names) called a "macro name." When the compiler encounters a macro name in the operation field of the source program it looks for the definition associated with that name. The compiler then executes the orders in the macro definition, which results generally in No. 1 ESS code being generated. This code varies, depending on the parameters of the macro call.

A macro definition has the form:

$$
\begin{array}{lll}
\text{DEFIN} & \text{op}_1 & \text{dum}_1\,,\text{dum}_2\,,\cdots,\text{dum}_n \\
 & \text{order}_1 & \\
 & \text{order}_2 & \\
 & \vdots & \\
 & \text{order}_k &
\end{array}
$$

ENDEF

EQUAL op_1 $op_2, op_3, \cdots, op_n$

$op_1, op_2, op_3, \cdots, op_n$ are all names of this definition. $dum_1, dum_2, \cdots, dum_n$ are dummy parameters. The body of the definition consists of the series of orders: $order_1, order_2, \cdots, order_k$. These orders are of four types: No. 1 ESS instructions, macro calls, macro orders, and pseudo operations. The macro orders are a special subset of the PROCESS language useful mainly in writing macro definitions. They instruct the compiler to take certain actions *during compile time*. To help distinguish macro orders from other types of orders, the symbol * is the first character of each macro order name. The meaning and syntax of the four order types are discussed more fully in this and succeeding sections.

The actual notation used by programmers in writing macro definitions is limited by available keypunch symbols, which leads to awkward notation in some cases; for these cases a notation more suitable for exposition is used here. As previously, the remainder of this section uses small letters for variables, whereas capitals and special symbols such as $ are used literally. A detailed example of a macro definition, a macro call, and the operation of the compiler in expanding the definition is given in Appendix A.2.

4.1 *Parameter References*

Depending on the exact nature of a macro call, different codes are generated by its macro definition. In order to express this dependence of the code to be generated on the various parts of the macro call, a general scheme for referring to these parts is needed. The form of a macro call is:

loc op $p_2, p_3, \cdots, p_n$

A call is composed of a location (also called p_0), an operation (also called p_1) and a series of parameters. Syntactically, the location and operation are strings of six or fewer alphanumerics. The parameters, on the other hand, may have some internal structure. A parameter p_s is either a string of alphanumerics (including the null string) or it has the form $o_s(p_{s,1}, p_{s,2}, \cdots, p_{s,n_s})$ in which s is an ordered set of (position) integers, and o_s is any string of alphanumerics. An o_s is called the *operator* of parameter p_s. p_s with o_s removed is called the *body* of p_s. The *strip* of p_s is the body of p_s with the outside parenthesis removed. The *remainder* of p_s is defined only for $s = 2, 3, \cdots, n$. The remainder of p_s

is p_s , p_{s+1} , $\cdots$, p_n . The location, operation, and the parameter parts, operator, body, strip, and remainder of any p_s may all be referred to directly within a PROCESS III macro definition.

There are two methods to refer to parameters and parameter parts. The first method is by using "parameter indices." There are six "parameter indices" named I0,$\cdots$,I5 for use in writing macro definitions. A parameter index may be set to any parameter by the macro order *SET. For example:

*SET x,I0,s

This order sets parameter index I0 to parameter p_s if it exists; otherwise control jumps forward to location x in the macro definition. There is a companion macro order *ADV. For example:

*ADV x,I0,t

Assuming t is an integer and I0 is originally set to p_s , where $s = s_1 , j$, then this order will set I0 to $p_{s_1,(j+t)}$ if such a parameter exists, and control passes to the next order; otherwise control jumps forward to location x in the macro. For example, if I0 is set to $p_{2,1}$ and $t = 1$, then after a successful *ADV, I0 is set to $p_{2,2}$.

One may refer to the parameter part to which a parameter index is set as follows: assume Ij is set to p_s

"*O""Ij" refers to the operator of p_s .
"Ij" refers to the body of p_s
"*S""Ij" refers to the strip of p_s
"*R""Ij" refers to the remainder of p_s .

The second method allows one to refer to parameters p_1 , p_2 , $\cdots$ p_n and their respective parts directly by referring to the dummy parameters written on the DEFIN statement. The DEFIN statement has the form:

DEFIN op dum$_2$,$\cdots$dum$_j$,$\cdots$,dum$_n$

dum$_j$ may be any string of six or fewer alphanumerics. In writing the macro definition following a DEFIN statement, one may refer to parameters and their parts as follows:

"LOC" refers to p_0
"ZOP" refers to p_1
"*O""dum$_j$" refers to the operator of p_j
"dum$_j$" refers to the body of p_j

"*S""dum$_j$" refers to the strip of p_j
"*R""dum$_j$" refers to the remainder of p_j.

If by using an expression available for referring to parameters one refers to a parameter that does not exist, then that expression refers to the symbols MSP (missing parameter).

All the above ways of referring to parameters are called parameter references ⟨pr⟩. Parameter references can be concatenated with each other and can be concatenated with alphanumerics. (Alphanumerics exclude special symbols.)

† A concatenated parameter reference ⟨cpr⟩ is defined to be of the form:

$$x \text{ or } \langle \text{pr} \rangle \text{ or } \langle \text{cpr} \rangle \langle \text{cpr} \rangle^*$$

in which x is any string of alphanumerics.

† A ⟨pr⟩ refers to a parameter. A string of alphanumerics x refers to x. A concatenation of x's and ⟨pr⟩'s refers to the concatenation of the symbols to which the x's and ⟨pr⟩'s refer (referents). Generally, the concatenation of any set of expressions refers to the concatenation of the referents of the individual expressions.

4.2 *Numerical Indexing*

There are three "numerical indices," M0,M1,M2, available for use in writing macro definitions. These indices refer to numbers. A numerical index may be set to a number with the macro order *SFI. For example:

*SFI x,M0,n_1,n_2

This sets index M0 to n_1 with a limit of n_2, where both n_1 and n_2 are positive integers. If $n_2 < n_1$, the index will not be set and a jump forward to x will be executed. There is an associated macro order *AFI. It is written:

*AFI x,M0,n

If M0 is set to j when this order is encountered, then M0 will be set to $j + n$ provided $j + n$ does not exceed the limit established on the last *SFI order that referred to M0, and control jumps back to location x in the macro; otherwise control passes to the next order.

* This recursive definition states that an x or a ⟨pr⟩ is a ⟨cpr⟩ and that any concatenation of ⟨cpr⟩'s is also a ⟨cpr⟩.

† This and any paragraphs similarly marked may be omitted without loss in continuity by those not interested in the detailed syntax of macro definitions.

One may refer to the number to which a numerical index Mj refers by the expression "Mj".

† ⟨cnr⟩ is defined to be a concatenation of numerical index references. A ⟨cmr⟩ is defined to be of the form:

⟨cnr⟩ or ⟨cpr⟩ or ⟨cmr⟩⟨cmr⟩

⟨cmr⟩ in itself is not significant; it is used as a convenience in a definition given below.

4.3 *Naming Symbol Strings and Parts of Symbol Strings*

A string of alphanumerics may be named and later referred to by this name. Also, space must be allocated to hold the strings to which the name refers. One method of naming strings and at the same time allocating space is accomplished outside all macro definitions with the NAME statement:

```
NAME    nam             siz,strg
```

nam is the name of the string, siz is the maximum-length string to which this name refers, and strg is an initial string of symbols to which nam refers. The name of a string is limited to six characters.

A method of renaming strings is with the macro order *ST. For example,

```
*ST             s,p
```

gives the string s the name p previously assigned by a NAME statement. Later the string s may be referred to by writing $[p]$.

† A string reference ⟨sr⟩ is defined to be of the form [⟨nr⟩] in which ⟨nr⟩ is defined to be of the form:

⟨cmr⟩ or ⟨sr⟩ or ⟨cmr⟩⟨sr⟩ or ⟨sr⟩⟨cmr⟩

An ⟨sr⟩ of form [⟨nr⟩] refers to the string whose name is the referent of ⟨nr⟩ as defined by using the NAME or *ST statements. If the referent of ⟨nr⟩ is not such a name, then [⟨nr⟩] refers to UN (undefined name).

4.4 *Special Functions*

Since it is expected that the parameters of a macro call will in many instances be the names of temporary storage elements such as items, registers, full words, and so on, means are provided for referring to properties of storage elements. These properties are:

(1) Type: [T.⟨nr⟩] refers to different characters, depending on what ⟨nr⟩ refers to.

If ⟨nr⟩ refers to:	[T. ⟨nr⟩] refers to:
an item	S
a full word	F
a number	W
a register item	P
a register	R
none of the above	UN

(2) Item properties: if ⟨nr⟩ refers to an item, then [S.⟨nr⟩], [D.⟨nr⟩], [M.⟨nr⟩], refer respectively to the size, displacement and mask of this item. If ⟨nr⟩ is not an item all three expressions refer to UN.

4.5 *Reference Expressions*

† A reference expression is a basic element in writing macro definitions. Recalling the various allowable bracketed expressions (i.e., [⟨nr⟩], [T.⟨nr⟩], [S.⟨nr⟩], [D.⟨nr⟩], [M.⟨nr⟩]), let ⟨csr⟩ be any concatenation of these, or null. A reference expression ⟨r⟩ is defined as any concatenation of ⟨nr⟩'s and ⟨csr⟩'s. ⟨r⟩ is of the form:

⟨nr⟩ or ⟨csr⟩ or ⟨r⟩⟨r⟩

† A legitimate reference expression always refers to some string of symbols. The interpretation of this string of symbols in turn depends on its position within the macro string.

4.6 *Conditional*

It has been shown how one can refer to parameters and various functions of parameters. Any such reference has been called a "reference expression," and has been symbolized by ⟨r⟩. The problem now is to produce code that depends upon these parameter functions. To do this some way of specifying decisions is required. The conditional is provided for this purpose.

One of the forms of the conditional is:

$$\$c, q_1, q_2, n_1, n_2\$$$

Syntactically, c, q_1, q_2, n_1, n_2 are all of the form ⟨r⟩. A legitimate c refers to the letters C, E, G, or L and indicates the type of comparison to be made. q_1 and q_2 refer either to strings of symbols or numbers that are to be compared depending on the interpretation of c. n_1 and n_2 refer to numbers that indicate how many characters following the conditional (after the second \$) are to be omitted: n_1 characters if the condition is met, n_2 characters if the condition is not met. For the condition indicated

by the letter C, the compiler compares the two strings referred to by q_1 and q_2 for identity. For the conditions indicated by the letters E, G, and L the compiler compares the two numbers referred to by q_1 and q_2 to determine if q_1 is respectively equal to, greater than, or less than q_2.

Another conditional is of the form:

$$\$X, p, n_1 , n_2\$$$

Syntactically, n_1 and n_2 are ⟨r⟩'s, but p must be a parameter reference ⟨pr⟩. The interpretation of this conditional by the compiler is: if the parameter referred to by p exists, omit the next n_1 characters; if not, omit the next n_2 characters (after the second \$).

Finally there is:

$$\$U,n\$$$

which means "omit the next n characters." n is of the form ⟨r⟩.

In general, conditionals may be concatenated with each other and with reference expressions.

4.7 *Form of Orders Used in Writing Macros*

A macro definition is composed of a series of orders. The form of these orders is:

loc $\quad$ op_1 $\quad$ $p_2 , p_3 , \cdots , p_n$

or

loc $\quad$ * $\quad$ $\text{op}_2 , p_2 , p_3 , \cdots , p_n$

loc is an ⟨r⟩ consisting of six or fewer characters; op_1 is any concatenation of ⟨r⟩'s and conditionals totaling six or fewer characters in length. Since six characters do not allow many ⟨r⟩'s or conditionals for op_1, the second form is available, in which op_2 is the same as op_1 except there is no limit on its length. A parameter p is either a concatenation of ⟨r⟩'s and conditionals or of the form $\text{op}(p,p,\cdots,p)$.

Thus an order used in a macro consists of conditionals which must be performed, reference expressions which must be interpreted, and operations which must be performed. The compiler does these things in the following fixed sequence.

(1) The conditionals are performed in sequence from left to right. A conditional is performed by:

(a) first interpreting all ⟨r⟩'s in the conditional (substituting referents for references) and then,

(b) certain parts of the order are omitted, depending upon the kind of conditional and substituted referents.

(2) All ⟨r⟩'s in that part of the order which remains are now interpreted.

(3) The resulting order (called an interpreted order) is performed. The resulting order is one of four types:

(a) an ESS instruction in the format required of such an instruction. If the compiler arrives at one of these in a macro definition, the ESS instruction is made part of the compiled object program.

(b) a macro call of the form described in Section 4.1. If the compiler arrives at one of these it transfers control to the definition of this macro and begins executing the orders in that definition.

(c) a macro order. Some of these already have been described, namely, *SET, *ADV, *ST, *SFI, and *AFI. The remainder of the macro orders are described below.

(d) a pseudo operation. The compiler executes the pseudo operation just as though it had been part of the input source program (see Section 5.1).

4.8 *Additional Macro Orders*

The remaining macro orders are all jumps or skips of one sort or another. Let x be any string of six or fewer alphanumerics, n a number.

$$*\mathrm{J}\begin{pmatrix}\mathrm{F}\\ \mathrm{B}\end{pmatrix} \qquad x$$

means jump $\begin{pmatrix}\text{forward}\\ \text{back}\end{pmatrix}$ to the location x.

$$*\mathrm{JF} \qquad \mathrm{OUT}$$

means jump out of this macro definition.

$$*\mathrm{S}\begin{pmatrix}\mathrm{F}\\ \mathrm{B}\end{pmatrix} \qquad n$$

mean skip $\begin{pmatrix}\text{forward}\\ \text{back}\end{pmatrix}$ over n orders.

$$*\mathrm{X}\begin{pmatrix}\mathrm{F}\\ \mathrm{B}\end{pmatrix} \qquad x$$

means execute the order at location x $\begin{pmatrix}\text{forward}\\ \text{back}\end{pmatrix}$ of this execute order and then return to the order directly after this execute order.

In the case of the execute order, the location x must be in the same definition as the execute order. In the case of the jump or skip orders the transfer of control can be outside the macro in which the jump or skip occurs.

A jump forward to location x in the definition of a macro called MAC causes the compiler to look for x in MAC, forward of the *JF order. If x is not found in MAC, the compiler continues to look forward of where the call for MAC occurred. This process continues until the x is found or the end of the input program is reached. A jump back, *JB, is executed similarly, except that if PROCESS III gets back to the input source program without having found an x, it will not look any further; the compiler will then process the next order in the source program. The skip macro orders follow corresponding rules. A detailed example of how the compiler handles a macro call is given in Appendix A.2.

V. SOME RELATED DETAILS

There are many features of PROCESS III that have been omitted for the sake of brevity. However, a few details are mentioned below in an effort to complete the general facilities of the compiler.

5.1 *Pseudo Operations and Output Listing*

PROCESS III has a variety of pseudo operations. Pseudo operations are orders to the compiler that cause it either to generate data or to take some special action. Many of the special actions have to do with print control of the output listing. Two typical pseudo operations are:

```
OCT      1000007777777
SPACE    2
```

The first generates 37 bits of data consisting of the octal number shown; the second causes two blank lines to be "printed" on the output listing.

The output listing of the compiler is part of the documentation of the No. 1 ESS program. The listing contains the symbolic source program as written by the programmer, and also an octal representation of the object program. An example is shown in the Appendix, Section A.3.

5.2 *Machine Restrictions*

An interesting feature of PROCESS III is its ability to check for (and sometimes correct) certain violations in the source program. In addition to the usual checking performed by an assembler (e.g., unde-

fined and multidefined symbols), PROCESS III checks for illegal sequences of basic order instructions. These sequences, usually couplets or triplets, are illegal because of timing restrictions of the No. 1 ESS central control. The compiler either flags the violations or inserts EE (no operation) instructions to correct the sequence.

5.3 *Input and Correction Features*

The input to PROCESS III may be either tape or cards; in the case of cards, two formats are available, symbolic or crunched. Symbolic card input means that there is a single basic instruction or order or procedure or pseudo operation per physical card; crunched card input is simply a compressed version of the symbolic information, so that more than one instruction is introduced per physical card. With crunched input every instruction in the source program is assigned a sequence number. These sequence numbers may be used by the programmer to modify his source program conveniently when he needs to recompile.

VI. SUMMARY AND CONCLUSION

A description of PROCESS III, a compiler-assembler for No. 1 ESS, has been given. The emphasis has been on the factors influencing the design of the compiler, the built-in PROCESS language and the facilities available for extending the source language.

The approach used in the design of the compiler has proved very useful, primarily because of the flexibility it has provided. Outstanding among the merits of this approach is the fact that there now exist several telephone-oriented procedures in a language understandable to programmers. This is not to say, however, that PROCESS III is the final answer to a "telephone language." The authors feel that it is accurate to say that PROCESS III has laid a solid foundation for a future PROCESS *n*.

VII. ACKNOWLEDGMENTS

To acknowledge all contributors to the design and implementation of a compiler at this late date would be very difficult. The art of designing compilers has matured considerably in recent years but not so dramatically that one can point to unique clear-cut breakthroughs. A new compiler is almost always a few new ideas mixed in with many old ones. So it is with PROCESS III. Thus the authors single out no specific articles in the literature — thanks are due to all workers in this field. We should

like to mention, however, S. H. Unger, under whose direction a predecessor compiler was built; N. S. Friedman, who programmed the macro definition and executive routines; R. E. Archer, who programmed the Compool and loader facilities; and W. C. Jones, under whose direction some early work was done on PROCESS III.

APPENDIX

A.1 *Realistic Example of a Telephone Function and Its Program*[3,4]

The example shown in Fig. 1 is a realistic subprogram taken from the coin charge sequence of No. 1 ESS. It shows the application of the general-purpose procedures in programming telephone functions. It also demonstrates the usefulness of programmer-defined procedures such as LINK, which links two call registers, and SZREG A, which generates a program to hunt and reserve an idle call register specified by A. The accompanying flow chart (see Fig. 2) shows the close correspondence between the procedures of the PROCESS language and the telephone functions depicted on the sequence chart.

A.2 *Detailed Example of Macro Definition and Macro Call*

Definition of a macro named MV:

```
DEFIN   MV      A,B,C                   Order
        *SET    OUT,I0,2                1
        MK      "*S" "A"                2
XYZ     *ADV    OUT,I0,1                3
        *SF     $C,[T."I0"],R,0,2 $ 2,0 4
        KM      "*S" "I0"               5
        *JB     XYZ                     6
        W"I0"   0,K                     7
        *JB     XYZ                     8
ENDEF
```

Purpose: to move the contents of A to B to C, $\cdots$. A may be an indexed or unindexed memory location. B may be an indexed or unindexed memory location or a register. Example: assume the macro call is

```
MV      JACK,X,(JILL,Y)
```

where JACK and JILL are call store locations and X and Y are index registers.

Upon seeing this call, the compiler goes to the definition of MV. The steps taken by the compiler in expanding this macro call follow:

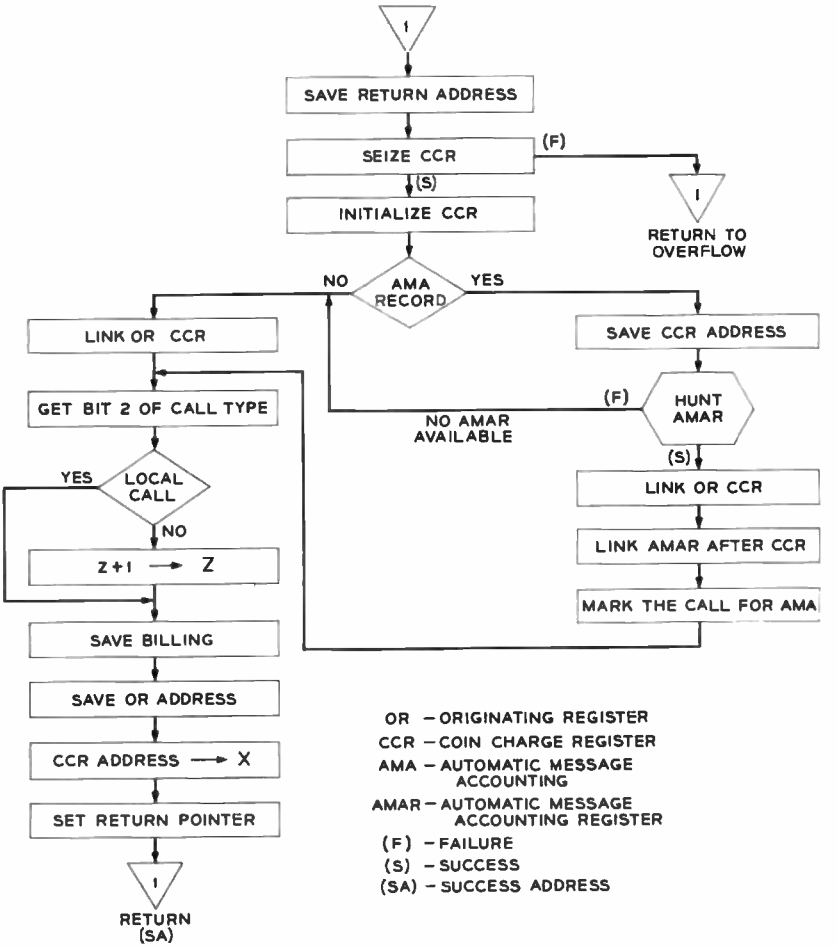


Fig. 1 — Subprogram from coin charge sequence.

(1) it sets I0 to JACK; (order 1)

(2) it produces:

MK JACK (order 2)

(3) it advances I0 to X; (order 3)

(4) it interprets the conditional,

SC,[T.X],R,0,2S

of order 4, which results in order 4 being interpreted as

*SF 2

```
        PART ØF CØIN CHARGE PRØGRAM



                   7           EXTERN    ØRØVF2,AMSZNØ
                   9   52X2    MØVE      J,IO
                  10           SZREG     CNC(K),ØRØVF2
                  11   XX      LØØP      1,V.S.CNC-1,1,Z
                  12           MØVE      0,(1,KA)
                  13   XX      ENDLØØP
                  14           IF        (RAMA,X),NE,V.RECRD,NØAMA1
                  15           MØVE      Y,T1
                  16           GØ•TØ     (AMSZNØ,,J)
                  17           GØ•TØ     NØAMA1
                  18           GØ•TØ     NØAMA1
                  19           MØVE      T1,Y
                  20           LINK      ØR(X),CCR(Y)
                  21           LINKA     AMA(Z),CCR(Y)
                  22           MØVE      1,(AMAR,Y)
                  23   MAIN    AND       (CTYP,X),2,Z
                  24           IF        (CZL,X),E,1,ØR(Z,1,Z)
                  25           MØVE      (BILL,X),(CHIN,Y)
                  26           MØVE      X,T1
                  27           MØVE      Y,X
                  28           SETPT     PT1
                  29           GØ•TØ     (TO,M)
                  30   NØAMA1  LINK      ØR(X),CCR(Y)
                  31           GØ•TØ     MAIN
   0000064        32           END
```

Fig. 2 — Flow chart for sequence of Fig. 1.

(5) it arrives at order 7, which produces:

WX 0,K

(6) it returns to order 3; (order 8)

(7) it advances I0 to (JILL,Y); (order 3)

(8) it interprets the conditional, finding that (JILL,X) is not a register and skips to order 5;

(9) it produces:

KM JILL,Y (order 5)

(10) it returns to order 3; (order 6)

(11) it cannot advance I0 any further and therefore jumps out of the macro definition. (order 3)

A.3 *Example of an Output Listing*

Fig. 3 is a typical output listing.

Starting on the extreme left, the interpretations of the columns are:

(1) relocatable locations assigned to the object program

(2) a three-character console code corresponding to the operation part of the instruction

(3) the 37-bit octal representation of the instruction

(4) sequence numbers: each statement in the source program has such a number

```
                                     PART ØF CØIN CHARGE PRØGRAM
                                          6          EXTERN   ØRØVF2,AMSZNØ
                                          7   S2X2   MØVE     J,T0
                           0C0C0000           S2X2     SYN    X.
0000000   120 000500 C015621                           JM     T0
                                          8          SZREG    CNC(K),ØRØVF2
0000001   010 040040 00C0004                           T      YASCNC,,J
0000002   010 C00040 00000C01                          T      ØRØVF2
0000003   750 C3364 00000000                           WK     0,Y
                                          9   XX     LØØP     1,V.S.CNC-1,1,Z
                           0C000014           9F0      SET    V.S.CNC-1
                           000C0C01           910      SET    1
0000004   730 00354 0C000001                           WZ     1
                           0C000005           XX       SYN    X.
                                         10          MØVE     0,(1,KA)
00000C5   042 12021G C0C0001                           EZEM   1,KA
                                         11   XX     ENDLØØP
0000006   730 03754 00000C01                           WZ     V.910,Z
0000007   430 07614 000C0014                           CWR    V.9F0,Z
000001C   037 000166 C0CC005                           TCLE   XX
                                         12          IF       (RAMA,X),NE,V.RECRD,NØAMA1
0000011   720 00350 0C0C001C                           WL     M.RAMA
0000012   350 025642 C0C0016                           MK     A.RAMA,X,PL
0000013   742 04361 0C0C001C                           CWK    V.V.RECRD*E.3
0000014   033 000146 C000060                           TCAU   NØAMA1
                                         13          MØVE     Y,T1
0000015   130 000540 0015622                           YM     T1
                                         14          GØ*TØ    (AMSZNØ,,J)
0000016   010 C40040 0000002                           T      AMSZNØ,,J
                                         15          GØ*TØ    NØAMA1
0000017   010 000040 C0C0060                           T      NØAMA1
                                         16          GØ*TØ    NØAMA1
0000020   01C 000040 00C0060                           T      NØAMA1
                                         17          MØVE     T1,Y
0000021   300 001400 C015622                           MY     T1
                                         18          LINK     ØR(X),CCR(Y)
0000022   122 000510 C014C04                           XM     00C04
0000023   005 04C022 0000003                           ENTJ   CØLINK
0000024   750 03364 00000000                           WK     0,Y
                                                       EXTERN CØLINK
                                         19          LINKA    AMA(Z),CCR(Y)
0000025   202 03501C 00C0C00                           MB     A.Y4LI,Z
0000026   100 C3440 4C040C00                           LM     M.Y4LI,Z,ES
0000027   720 00350 37777777                           WL     M.Y4LINK
0000030   350 C31642 C0C0C02                           MK     A.Y4LINK,Y,PL
0000C31   132 030552 C0C0C02                           ZM     A.Y4LINK,Y,EL
0000032   112 034452 C000C02                           KM     A.Y4LINK,Z,EL
                                         20          MØVE     1,(AMAR,Y)
0000033   750 00364 04000000                           WK     V.1*E.20
0000034   720 C0350 0400000C                           WL     M.AMAR
```

Fig. 3 — Typical output listing.

(5) source and object program symbolic statements: the indented statements were generated by the compiler and were not part of the source program.

REFERENCES

1. Keister, W., Ketchledge, R. W., and Vaughan, H. E., No. 1 ESS System Organization and Objectives, B.S.T.J., this issue, p. 1831.
2. Harr, J. A., Taylor, F. F., and Ulrich, W., Logical Organization of No. 1 ESS Central Processor, B.S.T.J., this issue, p. 1845.
3. Harr, J. A., Hoover, Mrs. E. S., and Smith, R. B., Organization of the No. 1 ESS Stored Program, B.S.T.J., this issue, p. 1923.
4. Carbaugh, D. H., Drew, G. G., Ghiron, H., and Hoover, Mrs. E. S., No. 1 ESS Call Processing, B.S.T.J., this issue, p. 2483.

No. 1 ESS Call Processing

By D. H. CARBAUGH, G. G. DREW, H. GHIRON
and MRS. E. S. HOOVER

(Manuscript received January 15, 1964)

A telephone call goes through six basic stages: (1) detection of a service request, (2) interpretation of the digits dialed, (3) alerting the called customer, (4) establishing a talking connection, (5) call disconnect, and (6) charging. The way in which programs, assisted by circuits and temporary memory, advance a call from one stage to another in No. 1 ESS is described in this article.

1. INTRODUCTION

As mentioned in the description of systems objectives[1] for the No. 1 electronic switching system (ESS), the problem posed to the designer of a switching system intended for the widest possible Bell System use becomes: to provide economical means for switching a wide range of traffic, composed of many types of calls, each differing in some degree from the next.

Previous solutions to this problem resulted in systems composed of a multiplicity of switches and relay circuits, each performing a certain set of functions. A call was processed by the proper set of circuits at the proper time. In the more modern systems, common control circuitry takes over the most complicated parts of the decision-making process. For example, the No. 5 crossbar system, as presently developed, is so versatile that it is able to offer a wide range of services.

In the No. 1 ESS the principle of the common control has been carried even further by the use of a stored program electronic data processor, which consists of a central control and associated electronic memories. Relay circuits, such as trunk and service circuits, are kept as simple as possible.[2] Many of the functions performed by the circuits in previous systems are performed by the central control under the direction of the stored program.

The logical organization of the central control was designed to take advantage of the speed inherent in electronics for processing large num-

bers of calls. The variety of services is provided by the use of the stored program. A wide range of new services can be offered in the future with a fairly modest development of new circuitry. To provide existing offices with these new features will require few wiring changes, but will require a change of program.

The No. 1 ESS programs, like all large programs, are divided into functional blocks of instructions. Some of these blocks of instructions can be called on by other blocks to perform some specific function, such as looking up translation data in memory. Some blocks are basically concerned with an efficient input-output procedure. Others constitute the "mainline" programs, which advance the progress of the call. In all cases, similar programming actions are grouped together as much as possible so that a single program will suffice for many variations.

Each block of program requires the use of a "notebook" in temporary memory (call store) in which to leave data in the course of processing the call. In many cases, these areas of memory must be assigned at the start of a call and be kept intact over a period which usually lasts as long as a particular phase of the call. By analogy to relay registers which perform the memory function in relay systems, these areas of the call store are referred to as "registers." In dividing the total program into blocks, it is necessary to associate specific areas of the call store with only a few programs so that the formats of these areas need only be known to a limited number of programs. The need to engineer or assign areas of call store for most of the memory functions puts certain restrictions on the way the program is divided.

Although the basic job of the central control is call processing, certain administrative tasks are a necessary concomitant, such as taking traffic measurements and accepting data from the teletypewriter printer concerning changes made in the class of service and directory numbers assigned to lines. The stringent requirements for reliability which have come to be expected of telephone service have resulted in extensive maintenance programs designed to detect and diagnose equipment failures. In order to process a high volume of telephone traffic, the call-processing programs are largely separated from the maintenance and administrative programs. At specific points in normal call processing, checks are made to determine whether the system has behaved as expected; if not, maintenance programs are called in to determine the reason. Routine maintenance and administration functions are fitted in at times which will not hamper the system's basic tasks of call processing.

In the following sections of this paper the manner in which the system

supervises calls, receives and transmits signals, and connects calls in the switching network is described so that the program implementation can be more easily understood. The basic divisions of the call processing program are then outlined, and the corresponding organization of call store memory is described. Finally, a simple line-to-line call is traced to illustrate how the hardware, programs, and temporary memory are brought into play to process a typical call.

II. SUPERVISION AND SIGNALING

Each line has an appearance on a line link network.[3] Associated with each appearance is a line ferrod sensor,[4] which is used to detect the flow of line current when a telephone customer lifts his receiver to request service. Cutoff contacts of a ferreed switch remove the bridged windings of the ferrod sensor during signaling and conversation so that they will not impair transmission or limit signaling range. The line ferrod is the initial supervisory point for all line service requests, as shown in Fig. 1.

After a service request is detected, the line is connected to a digit receiver: a customer dial pulse receiver if the customer has a rotary dial on his telephone, a customer TOUCH-TONE receiver if the customer subscribes to TOUCH-TONE service. Either digit receiver provides dial tone to indicate that the system is ready to serve the request. In addition, the receiver is provided with facilities to test for crosses to foreign potentials which would simulate a service request, and to make a tip party test on two-party lines. The customer dial pulse receiver is capable of following subscriber dial pulses. The customer TOUCH-TONE receiver will both follow dial pulses and detect the presence of TOUCH-TONE signals. The digit receivers have appearances on trunk link networks. At the time a line is connected to a digit receiver, the

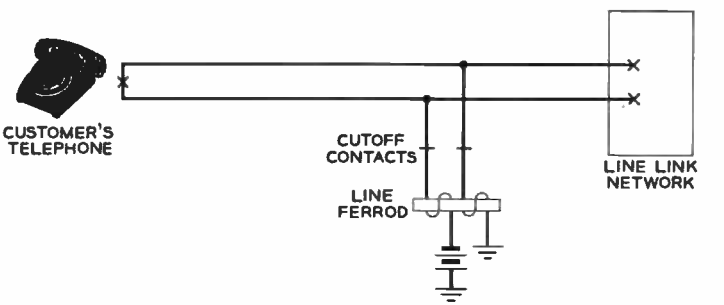


Fig. 1 — An idle customer line supervised at its line ferrod.

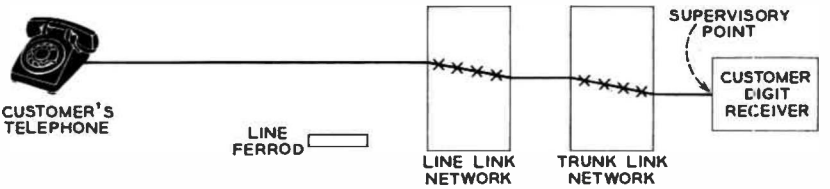


Fig. 2 — A line supervised at customer digit receiver during dialing.

cutoff contacts of the line ferrod are opened and supervision is transferred to the digit receiver, as shown in Fig. 2.

A line is supervised at a digit receiver until the completion of dialing, at which time the call will take one of many possible courses. If the requested number is in the same office, a ringing circuit will be utilized. If the requested number is busy, a connection to a busy tone circuit will be established. If the requested number is in another central office, an outgoing trunk and a digit transmitter will be brought into play. In the latter case, the supervision of the calling line normally remains at the digit receiver until the completion of outpulsing.

On an intraoffice call, audible ringing is supplied to the calling line by a tone circuit. The tone circuit provides audible tone and supervision of the calling line in order that a call abandonment during ringing may be detected.

Twenty-cycle ringing voltage is supplied through a small number of ringing circuits rather than through relays in the incoming or intraoffice trunk circuits, as in previous systems. Each ringing circuit has a trunk link network appearance and is connected to a called line only until the call is answered or is abandoned. Various line tests are made by the circuit. Power cross detection guards the ferreed crosspoints of the switching network and other circuits. A pretrip test guards against the possibility of falsely charging a calling customer, and a check is made that ringing current is flowing out toward the called phone. Three groups of ringing circuits are provided, one for each ringing phase, as shown in Fig. 3. The same is done for the audible tone circuits. When ringing is to be established, a connection to a ringing circuit in the active phase is made, providing virtually immediate ringing.

The ringing circuit is provided with a ring trip relay which stops the flow of ringing current as soon as the called customer answers. A scanner ferrod sensor provides the means by which a program detects the answer. Similarly, scanner ferrods are the means by which the results of the tests mentioned earlier are determined. The network connections and the points at which the calling and called lines are super-

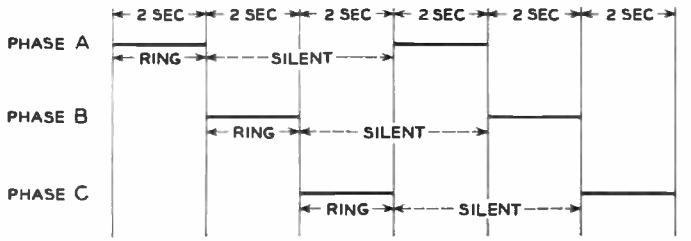


Fig. 3 — Three phases of ringing voltage provided for immediate ringing.

vised at this stage of an intraoffice call are shown in Fig. 4. At the time the ringing connection is established, the called customer's line ferrod is removed by opening its cutoff contacts.

On intraoffice calls, when customers are connected for talking, junctor circuits provide supervision and talking battery. Associated with each circuit are two scanner ferrod sensors, one to supervise each of the customers, as in Fig. 5. Bridged supervision is utilized and dc isolation is provided by the capacitor coupling used for speech transmission.

To summarize: the supervision of a calling line, during an intraoffice call, originates at its line ferrod, is transferred first to a digit receiver, then to an audible ringing tone circuit, and finally to one side of a junctor circuit. Abandonment of the call at any stage can be detected at one of these circuits. Similarly, the called line's supervision is removed from its line ferrod when a ringing connection is established and is transferred to the ringing circuit to detect answer. It is then transferred to the other side of the junctor circuit to detect its disconnect. Fig. 6 shows how supervision is passed from one circuit to another.

If the call destination is a customer in another central office, an outgoing trunk circuit to that office is selected and a digit transmitter is connected to the trunk circuit to transmit the called number. No. 1

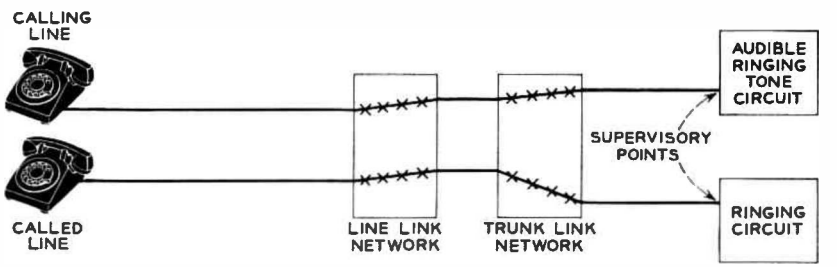


Fig. 4 — Supervision of calling and called lines during ringing.

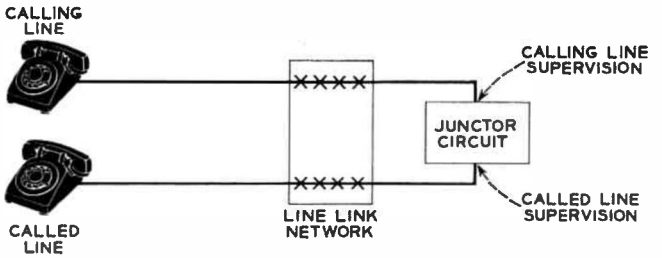


Fig 5.—Calling and called lines supervised at junctor circuit during talking.

ESS outgoing trunks are designed to provide: (1) supervision toward the calling line, (2) supervision toward the distant office, (3) speech transmission, (4) bypassing of all supervisory elements so that trunk tests can be made and pulsing can be performed by the transmitter, (5) compensation for different subscriber loop lengths, and (6) lightning surge protection.

Different types of digit transmitters are provided, to meet the needs of the offices which may be connected with No. 1 ESS. Dial pulse, multifrequency, revertive, and panel call indicator pulsing are available. Each transmitter, with an associated program, can test for the continuity and proper polarity of the trunks to which it connects, can detect start pulsing signals from a distant office, and can generate signals of the proper type.

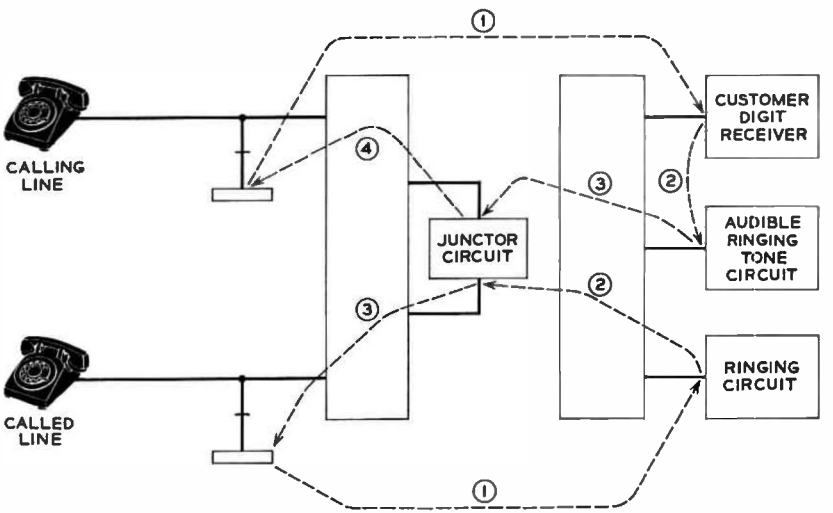


Fig. 6 — Transfer of supervisory points during an intraoffice call.

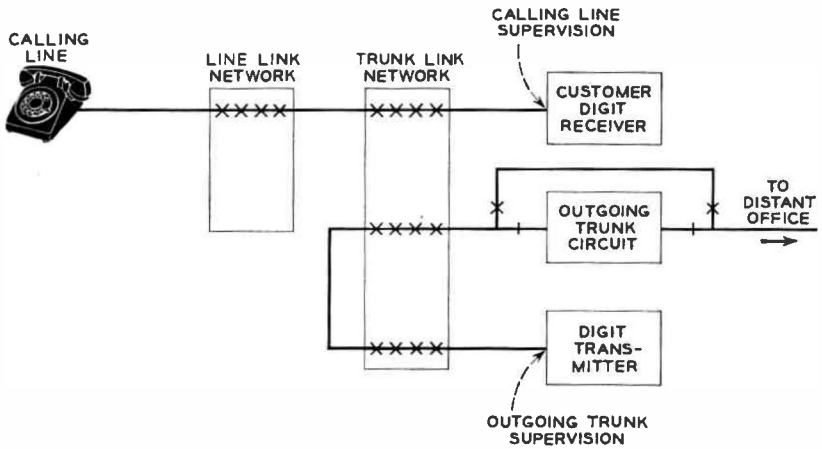


Fig. 7 — Supervision of an outgoing call during outpulsing.

Since a digit transmitter is connected to the selected outgoing trunk during the outpulsing stage of an outgoing call, the supervision of the calling line normally remains at the customer digit receiver until outpulsing is completed. Similarly, the outgoing trunk is supervised at the digit transmitter rather than in the trunk circuit. The network connections and supervisory points during outpulsing are shown in Fig. 7. After outpulsing, while ringing is being applied in the distant office, and during talking, the supervision of both the calling line and the trunk are at the trunk circuit, as shown in Fig. 8.

III. PROGRAMS INVOLVED IN CALL PROCESSING

In a program-controlled system such as No. 1 ESS, the circuits involved in advancing a call from one stage to another do not perform these actions by themselves. Control signals generated by programs

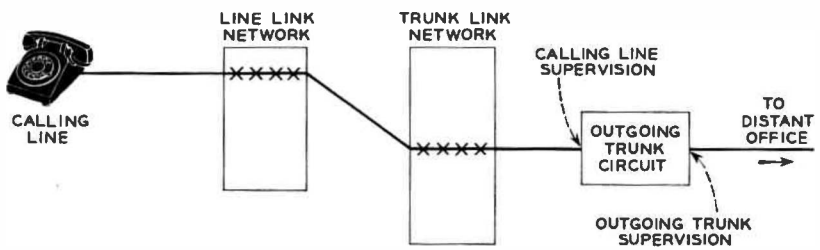


Fig. 8 — Supervision of an outgoing call during talking.

cause the circuits to change from one state to another. Similarly, control signals from the outside environment or changes in circuit states do not by themselves cause any system actions to take place. They activate scan points which are read and interpreted by programs. The programs determine the meanings of the scanner readings and perform the logic to decide what action should be taken.

The programs associated with call processing may be generally classified in three categories: (1) those which detect changes in the outside environment and constitute inputs to the ESS and those which produce changes in the outside world and constitute system outputs (these programs are referred to as input-output programs); (2) call control programs, which have only call-related purposes and whose function is to advance a call to completion; (3) programs of such a generally useful nature, which perform such a frequently used function, and which are sufficiently well defined that they may be considered to be service routines, used at will by call processing programs and others.

3.1 *Input-Output Programs*

3.1.1 *Input Programs*

The programs which detect system inputs are designed to be relatively simple, highly efficient programs which report changes or events to call control programs which analyze the report and perform any required actions. This is done because there is a very large number of inputs (scan points) to be interrogated regularly. The number of changes detected at any one time is expected to be quite small.

The program which detects line service requests interrogates all line ferrods in the office approximately ten times per second. The line scanners are arranged so that 16 line ferrod sensors are read simultaneously. The supervisory line scan program reports the origination to a call control program and continues its round of line scanner interrogation.

Another input program detects and counts dial pulses generated by customer dialing. It interrogates the scan points associated with the pulsing relays in customer dial pulse receivers 100 times per second. In addition to counting pulses, the program measures interdigital intervals and, when it determines that a string of dial pulses has ended, reports the count to a call control program which will determine whether any action should be taken. The program also performs two other auxiliary functions for which reports are made. One is to report when the first pulse of the first digit is detected, to tell the call control program that

it is time to remove dial tone. The second is to perform permanent signal and partial dial timing. If no pulses are received for an interval of 16 to 32 seconds, the program reports this to the call control program, which will handle the call from that point on.

A third input program scans the ferrods controlled by ring trip relays in ringing circuits. This program is activated ten times per second by an executive control program.[5] When the called customer answers, it reports to a call control program, which removes the ringing connection and establishes a talking connection.

The supervisory scanning program of junctor circuits looks only for changes from off-hook to on-hook. A change of state of a junctor ferrod from off-hook to on-hook may be a momentary hit on the line or an inadvertent switchhook jiggle. In addition, the customer may be flashing to initiate a special service request. Consequently, the supervisory junctor scan program reports the change to a hit-timing program which times the duration of the on-hook signal for an interval sufficiently long to discriminate among hits, flashes, and true disconnects. The result is reported to a call control program, which decides the appropriate action to be taken.

A trunk supervisory scan program detects a number of signals, because incoming trunks, outgoing trunks, and a variety of service circuits may be intermixed in a trunk scanner. The program can deduce some information from the changes which it is designed to detect. A change from off-hook to on-hook, for instance, indicates the start of a disconnect or a flash, and reports to the hit-timing program, which then performs the same timing functions as were described for the junctor supervisory program. A change from on-hook to off-hook on an incoming trunk constitutes a request for service, but on an outgoing trunk, it could indicate that the call was answered at a distant office.

The ability to discriminate between service requests on incoming trunks and answers on outgoing trunks is not built into the trunk supervisory program. It instead reports to a call control program that a change in the supervisory state of a trunk has occurred and goes on to see whether any more such changes have occurred on other trunk or service circuits. It is up to the call control program to sort out the requests from the answers.

3.1.2 *Output Programs*

When the No. 1 ESS data processor meets its external environment, it encounters an entirely different time scale. The program processes

parts of a call at a very high speed and on a time-shared basis. To operate relays in trunk circuits, activate network controllers, record call charge information on magnetic tape, or transmit a teletypewriter message takes a relatively long time. The processing of other call elements cannot be delayed to wait for these actions, so the program determines what action needs to be performed and buffers data for an output program which specializes in converting the buffered data to the desired action.

Outputs are distributed by the No. 1 ESS peripheral bus system to signal distributors, network controllers, teletypewriter control circuits, and others. The data, consisting of addresses and control information, are stored in temporary memory areas called peripheral order buffers (POB's). The program which controls the transmission of the data is called the peripheral order buffer execution program. It is responsible for seeing that the correct addresses and instructions are sent to the controllers, checking that the proper action was taken in response to the instructions, and reporting back to the call control program the success or failure of the requested action.

3.2 *Call Control Programs*

Each call control program performs a specific function, usually related to a stage in the progress of a call. This separation permits each program to be of a manageable size and to perform a reasonably well defined function. It also makes the addition of new features relatively easy, since the new feature will only affect a few programs.

On a normal intraoffice call, the call control programs which are brought into play and which are responsible for the handling of the call at various stages are: (1) the dialing connection program, used to set up a dialing connection; (2) the digit analysis program, used to record and analyze the digits dialed, and to determine the destination of the call; (3) the ringing and answer detection program, used to establish the ringing connection, detect the called customer's answer, and establish the talking connection; and (4) the disconnect program, used to control the disconnect of the call and restore the lines to the idle state.

3.2.1 *Dialing Connection Program*

A report from the supervisory line scan program that a line has requested service is the input to the dialing connection program. To serve the request, there are several things that this program must do. First, it must find a block of temporary memory in which to store data regarding

the calling line and the number that the customer will be dialing. Then it must acquire some information about the line. Does the customer have a rotary dial telephone or a TOUCH-TONE telephone? Is the line an individual, two-party, or four-party line? A PBX trunk? A coin line? Is it a disabled customer who is not able to dial? Has the line been denied service for some reason?

When the answers to the above questions have been obtained, the program knows what it must do. It must select an idle customer digit receiver of a type which is compatible with telephones on the calling line. It must cause a network connection to be made from the calling line to the selected digit receiver. After the connection has been established and the line cutoff contacts opened, it must cause the digit receiver to apply its power cross detection circuitry to the line and read a scan point to determine the results of the test. Next, it must cause the digit receiver to remove the power cross test circuit from the line and, for a two-party line, to apply a party test circuit and then read a scan point indicating the result of this test. After all the necessary tests have been performed and passed, the program causes a relay in the digit receiver to operate and connects a supervisory relay and dial tone to the line. The transfer of supervision from the line ferrod to the supervisory relay in the digit receiver is then checked by reading another scan point. After all the above have been successfully accomplished, the program activates the dial pulse scan program with respect to the chosen digit receiver. Fig. 9 represents the functions of the dialing connection program.

It should be noted that the dialing connection program *causes* the actions mentioned to be taken in the digit receiver, and it *causes* a network connection to be made. It does *not*, by itself, perform these actions in the order stated. Instead, it calls upon the services of the network control program to find an idle path from the line to the digit receiver and to load the network controller addresses and instructions in a peripheral order buffer (POB). The dialing connection program then calls on the circuit control program to load the desired relay and scan actions in the POB. After the buffer loading is complete, the POB execucution program removes the instructions from the buffer one at a time until it is emptied. The POB execution program then reports back to the dialing connection program that the job was done successfully. If trouble developed along the way, such as a cross to a foreign potential or a failure to transfer supervision, the rest of the actions would have been skipped and a failure report would have been returned to the dialing connection program.

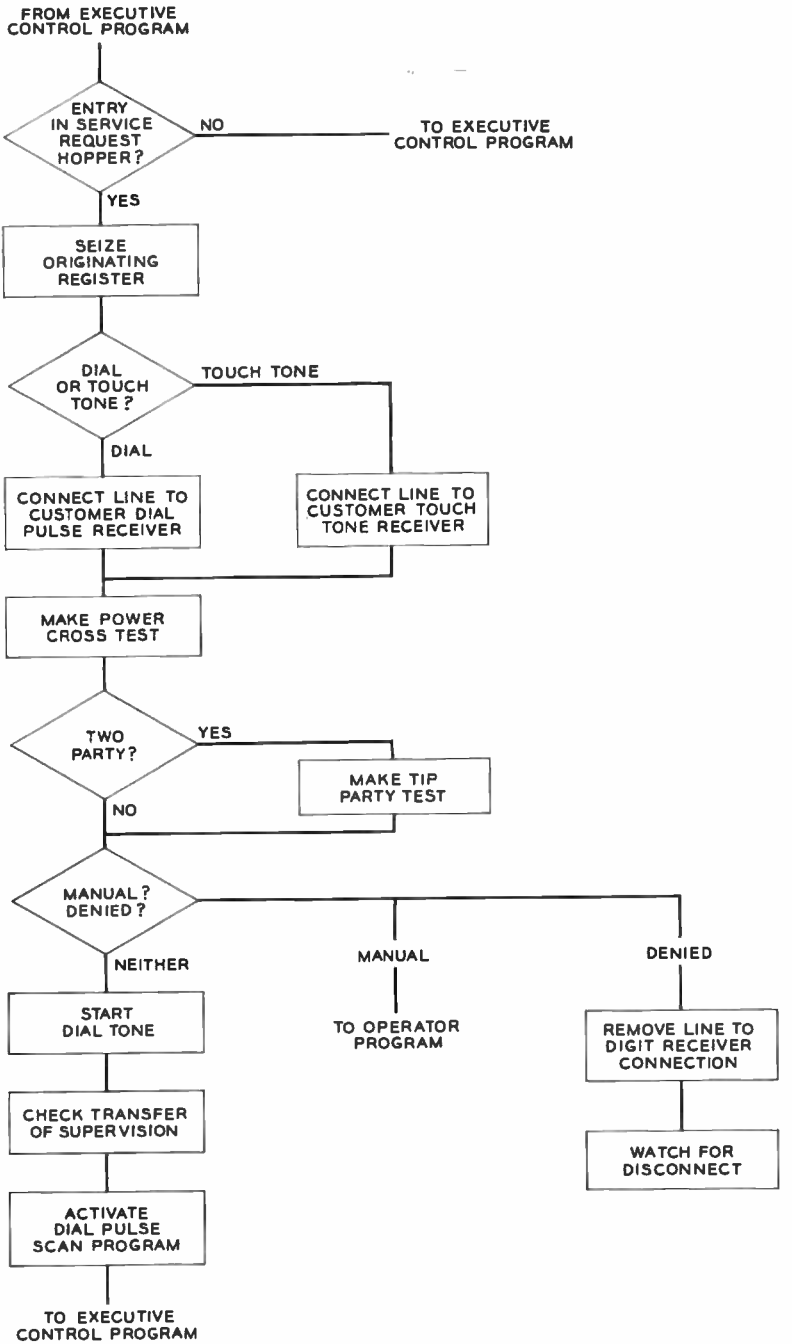


Fig. 9 — Functional flow chart of dialing connection program.

3.2.2 *Digit Analysis Program*

The responsibility of the dialing connection program ends with a successful report, and control of the call is passed to the digit analysis program. This program is responsible for recording, counting, and interpreting the customer's digits as they are dialed, determining the routing of the call if a valid number is dialed, determining whether a called number is busy or idle, and determining what the disposition of a call should be if it cannot be completed.

The dial pulse detection or the TOUCH-TONE digit detection input-output programs provide information to the digit analysis program by reporting to it when a digit is received, when an abandonment is detected, when the first dial pulse of the first digit is received, or when a permanent signal or partial dial time-out is detected. The program counts and stores the digits as they are reported to it in the temporary memory space (register) reserved by the dialing connection program. As the digits are received, some are merely counted and stored. For others, an analysis of the digits which have been received is made to determine what course to follow. For instance, when the first digit is dialed, the program determines whether the digit is zero (ten dial pulses). If so, dialing is finished and the call is directed to an operator. Control of the call then passes to a program which controls connections to switchboard operator's trunks. Any other digit (except the digit one, which is not a valid first digit of any area or office code) is stored, and a digit counter is incremented by one. Upon receipt of the third digit, an analysis of the first three digits is made. If the service code, 411, has been dialed, it is known that a connection to an information operator is desired. Dialing is finished, and control is passed to a program which will perform the desired action. Other three-digit codes are treated in a similar manner. If the first three digits are an office code in the same numbering-plan area, four more digits are expected. If the first three digits are those of a foreign area code, it is known that seven more digits must be received in order to be a valid number. The routing of the call can often be determined at this time. The office or area code determines whether it is an intraoffice call or an outgoing call. If the call is outgoing, it may be possible to determine the first-choice trunk group. However, some codes require six digits to be dialed before a route can be chosen.

Additional digits beyond the third are merely stored and counted until seven or ten have been received, depending upon whether a home numbering-plan office code or a foreign numbering-plan area code has been dialed. If the call is outgoing, the digit analysis program selects an

outgoing trunk to the distant office, selects a digit transmitter of the proper type, causes a connection to be established between them, and then passes control to an outpulsing program which will seize the trunk, make the necessary tests, and transmit the called number to the distant office.

On a call destination within the No. 1 ESS, the digit analysis program acquires some information regarding the class of the called line, finds out whether the line is busy or idle and, if idle, passes control to the ringing and answer detection program.

The digit analysis program also finds out whether a charge is to be made for the call. If so, it notifies a program which records the pertinent information.

If, during the dialing of the call, the customer hangs up and abandons the call, the digit analysis program causes the network connection to be removed, idles the digit receiver, and releases all temporary memory.

Permanent signal and partial dial reports are merely passed on to a program designed especially to take care of these conditions. Fig. 10 is a functional flow chart of the digit analysis program.

3.2.3 *Ringing and Answer Detection Program*

The ringing and answer detection program, as its name implies, controls all system actions on intraoffice or incoming calls from completion of dialing until answer. Its basic job is to connect ringing to the called line, connect audible ringing tone to the calling line or trunk, establish a talking connection if the call is answered, and remove all connections if the call is abandoned before answer.

The network locations of the lines and the type of ringing to be applied to the called line are needed to set up the ringing connection. The information regarding the calling and called lines (on an intraoffice call) is passed to the ringing and answer detection program by the digit analysis program. With this information, the ringing and answer detection program calls upon the services of the network control program, asking it to: (1) find idle circuits connected to the active phases of ringing and audible ringing (it was mentioned earlier that three groups of regular ringing trunks and audible ringing tone trunks are furnished, one for each phase of ringing to provide immediate ringing; the ringing and answer detection program keeps informed of which ringing phase is active); (2) find an idle path from the calling line to the selected audible ringing tone circuit; (3) find an idle path from the called line to the ringing circuit; (4) reserve a path from the calling line to the called line;

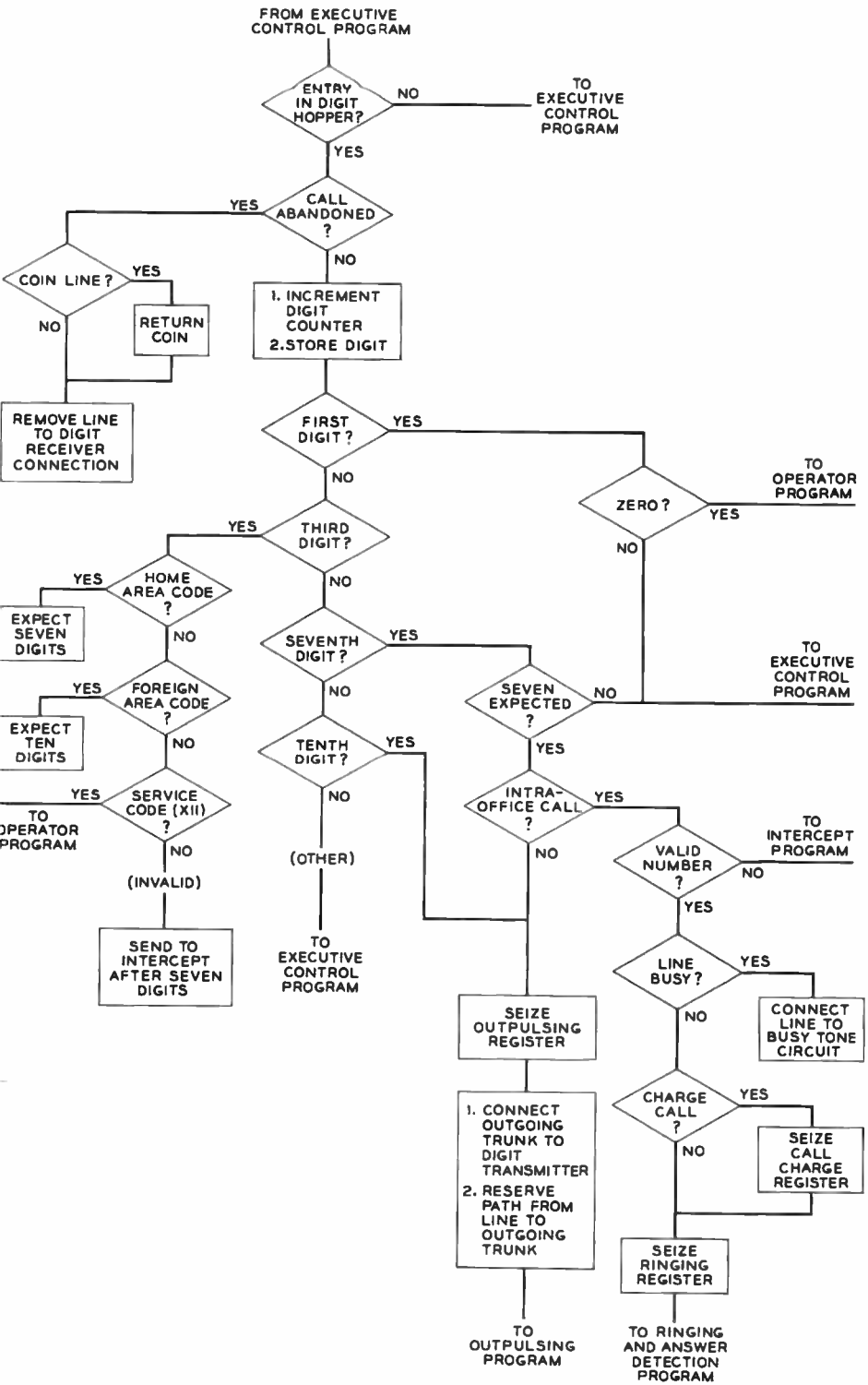


Fig. 10 — Flow chart of digit analysis program.

and (5) load the instructions for making the two connections into the peripheral order buffer.

The ringing and answer detection program next calls upon the services of the circuit control program to load further instructions in the POB which will cause the ringing circuit to perform a power cross test, a pretrip test, and a continuity test. The circuit control program also loads instructions to operate relays in the circuits which will apply twenty-cycle ringing voltage to the called line and audible ringing tone to the calling line in addition to checking that transfer of supervision has taken place at the audible tone circuit.

The job of the ringing portion of the program is almost complete at this time. All that remains is to activate the POB, wait until the POB execution program reports its success in completing the list of instructions, and activate the input-output programs which scan the ring trip ferrods and the audible ringing tone circuit supervisory ferrods.

An answer by the called party causes the ring trip relay to operate and saturate the ring trip ferrod. The answer is detected by the ring trip scan program and is reported to the ringing and answer detection program. Upon receipt of the report, the ringing and answer detection program again calls upon the services of the network control program and the circuit control program to release any operated relays in the ringing and audible circuits, to idle the two network paths, to set up a new path from each line to a junctor circuit, and to check that the supervision of each line is transferred to the junctor circuit.

If a charge is to be made on the call, the ringing and answer detection program reports the answer to the call charge program, which will record the time of answer.

After the ringing and answer detection program activates the input-output program which scans the junctor supervisory ferrods for disconnect, it finishes its responsibility for advancing the progress of the call. Fig. 11 is a flow chart of the functions performed by the ringing and answer detection program.

3.2.4 *Disconnect Program*

Unless a special service call or a coin call is in progress, no other call control program is called into play until one of the customers hangs up. At this time, on an intraoffice call, the junctor supervisory scan reports the event to the disconnect program. Hit timing will already have been performed by a program associated with the junctor supervisory scan program.

The functions of the disconnect program are: to detect flashing for

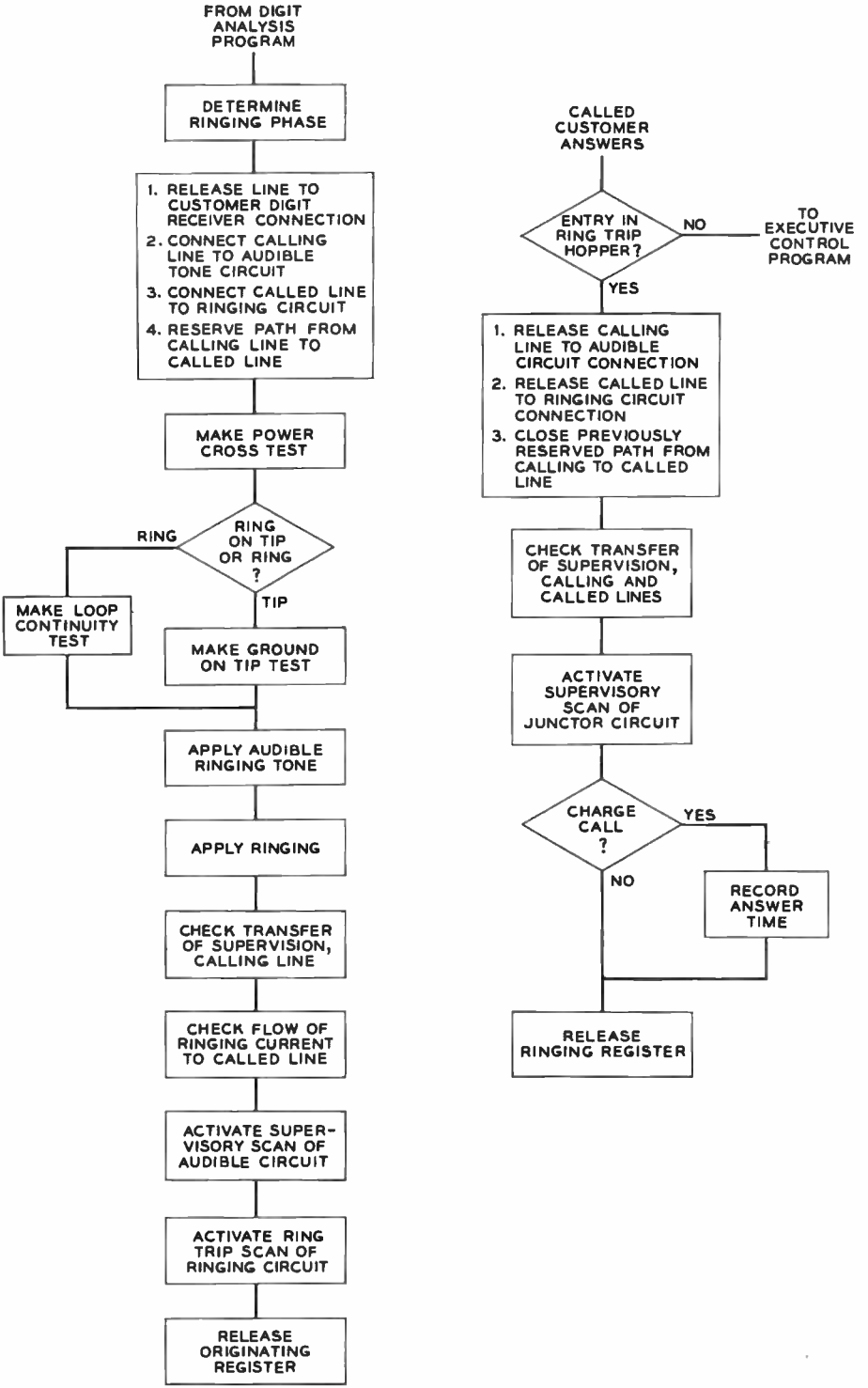


Fig. 11 — Ringing and answer detection program flow chart.

special services from those lines permitted to flash; to provide calling line control of the call (yet not permit it to keep a called line permanently tied up); to signal disconnect to a distant office over an incoming or outgoing trunk; to remove a talking connection at disconnect; to restore to idle any lines or trunks involved in the call; and to call in other programs to handle special conditions.

To determine the treatment for a disconnecting line, the program first finds out some information about the line. Is it the calling or called party? Is the line a coin line? Does it have any special services? Has the other end already disconnected? On an interoffice call, is it the line side or trunk side that is disconnecting? An incoming or an outgoing trunk? Is a charge record being made? Different actions are called for, depending on the answers to these questions.

On an intraoffice call, if the disconnect is from a *calling* line with no special services, the program knows that it does not have to perform flash timing. Because calling line control is provided, it is known that the call is over. The connection is not immediately removed, however, unless the *called* line has disconnected also. The program waits a reasonable length of time to permit the other party to hang up so that the line still off-hook does not appear to the system as a false request for service. When it detects the disconnect, it then calls upon the services of the network control and circuit control service programs to remove the network connection, to idle the junctor circuits, and to restore and check the line ferrods of the two lines.

The people involved in the conversation, however, may not perform the actions in the order given above. The called party may not hang up his phone within a reasonable time (10 to 12 seconds), or the calling customer may initiate another call very shortly after hanging up and before the called party replaces his receiver. In either of these cases, the network connection is removed and the lines are idled.

A called customer, upon disconnecting first, is permitted 10 to 12 seconds during which he may pick up his telephone and still find his original connection existing — if the calling customer has not hung up in the meantime.

The disconnect program calls in a coin control program when it knows that a coin line disconnects, in order that a coin-collect or coin-return action may be performed. A functional flow chart of the disconnect program is shown in Fig. 12.

3.2.5 *Other Call Processing Programs*

To handle call types other than the simple intraoffice call, other call control programs exist. Each is designed to take care of a particular

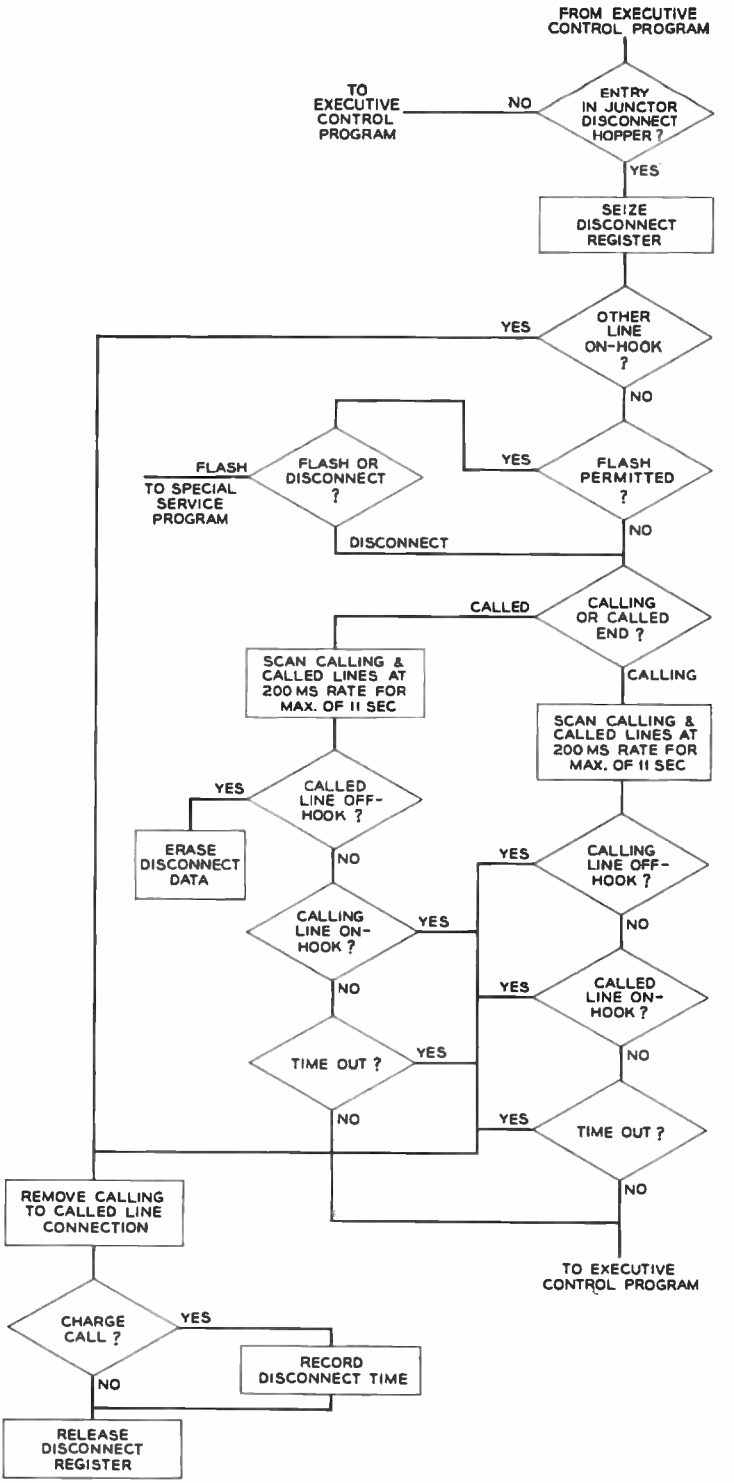


Fig. 12 — Program flow chart for disconnect of an intraoffice call.

stage of a call's progress or a particular type of call. A partial listing and a brief description of each follows:

3.2.5.1 *Outpulsing Program.* A call terminating in another central office requires that the called number be transmitted to that office in a form which it is prepared to accept. Several forms of signaling have developed over the years, and No. 1 ESS must be prepared to perform any of them. They are: dial pulsing, revertive pulsing, multifrequency pulsing, and panel call indicator pulsing. Since each is quite different from the others, a different program is designed for each.

To make an outpulsing program universally applicable to a number of call situations, the call procedure is designed so that a network connection between a digit transmitter and an outgoing trunk will already have been established. The outpulsing program then causes the outgoing trunk and the transmitter to be set in the proper states to test the continuity and polarity of the pair of leads to the distant office, to send a seizure signal, to detect a start pulsing signal, and then to cause the called number to be transmitted. It does this with the cooperation of input-output programs that are capable of generating dial pulses, multifrequency pulses, etc. The outpulsing program gives the dial pulse generating program (for instance) a digit. The dial pulse generating program forms the necessary number of dial pulses, measures an interdigital interval, and reports that a digit has been transmitted. The process is repeated until all required digits have been sent. Prefixing or deletion of digits is predetermined by the digit analysis program

3.2.5.2 *Operator Programs.* The actions required for calls to switchboard operators are different from those for other call types. As soon as a call is made to an operator, control is passed to the operator program. Calls to assistance and toll operators provide for joint holding, and these operators are permitted to ring back to a busy or an idle line. The operators also collect and return coins. In addition to the assistance and toll operators, there are information operators, business office operators, intercept operators, and repair service operators, all with their own functions and signaling arrangements.

3.2.5.3 *Permanent Signal and Partial Dial Program.* Those call attempts which become permanent signals or partial dials are handled by a program designed to switch the call first to an announcement requesting that the customer hang up and then to a distinctive receiver offhook tone for a timed interval. If these actions do not succeed in removing the condition, the program connects the line to an operator to see whether she can provide assistance, and then to the master control center for a maintenance man to test the line conductors (he determines whether a trouble condition exists). The permanent signal and partial dial pro-

gram guides the call through the various steps, performing the timing functions and requesting the needed network connections and relay operations in the circuits which are involved.

3.2.5.4 *Traffic Measurement Programs.* To determine the traffic levels in a No. 1 ESS office, a number of traffic measurements are made. Traffic engineers make use of peg counts, overflow counts, and usage counts to tell whether the number of circuits of various types or the number of trunks in various routes should be increased or decreased to carry the offered traffic.

A separate program can continuously generate service requests to measure and count the number of attempts on which excessive dial tone delays are experienced. The results of the traffic measurements are printed by a teletypewriter printer on a regular schedule.

3.3 *Service Routines*

Most call control programs use a number of service routines while controlling the progress of a call. Examples of service routine usage are: to request a change in a network configuration, to request the operation or release of a relay in a trunk or service circuit, and to obtain translation information. These service routines are used not only by the call control programs but also by the maintenance and diagnostic programs.

Because the service routines serve many clients under varying conditions, rules are established which must be obeyed whenever the service routines are used. The presentation of certain data in a certain format will cause a particular action to be performed, or particular translation information to be obtained. In use, the client sets up the necessary data in a prescribed manner and then passes control to the service routine. When the service routine finishes its requested action, it returns control to the client with data useful to the client in a predetermined location.

3.3.1 *Network Control Program*

The primary functions of the network control program are to hunt for idle network paths, to administer the network map and path memory, and to load instructions in POB's, which will be used to close network paths.

In the process of performing these functions, the network control program is provided with the ability to find an idle trunk in a group, to make a second trial if all the paths to the first selected trunk are busy, and to consult the translation program to find an alternate route if all trunks are found busy in the first-choice route.

Since the record of the busy or idle condition of all links in the switching network is kept in temporary memory (called the network link map), the network control program can reserve a path from one terminal to another for expected use at a later stage of a call. Similarly, the information regarding a connection established in the network is kept in temporary memory associated with network terminals (called path memory). The network control program records pertinent information about a path at the time a connection is made or removed.

3.3.2 *Circuit Control Program*

When a call control program determines that a change of state in a trunk circuit or service circuit is required to make a test or cut through a talking pair, the call control program calls upon the services of the circuit control program. It need only inform the circuit control program of the type of circuit to be used and the function to be performed. The circuit control program will then load the POB with the signal distributor operations necessary to implement the change and any scanner actions needed to check that the operation was performed successfully.

3.3.3 *Translation Program*[6]

Translations from line equipment number to calling line class and directory number, from office code to routing information and charge class, and from directory number to line location and class, are needed in the No. 1 ESS just as they are in other common control systems. Instead of this information being obtained by wired cross-connection fields, it is contained in tables stored in memory. The translation program gains access to the translation tables.

3.3.4 *Coin Control Program*

The collection and return of coins in coin phones is performed by a small number of coin control circuits in No. 1 ESS. A program which specializes in performing the functions required for these actions may be called in by any other call control program. In order for another call control program to use the coin control program, it must first establish a connection from the coin line to the coin control circuit and inform the coin control program whether a collection or a return is to be made.

The principal programs used in processing calls in No. 1 ESS are shown in Fig. 13. The connecting lines indicate transfer of information or of control.

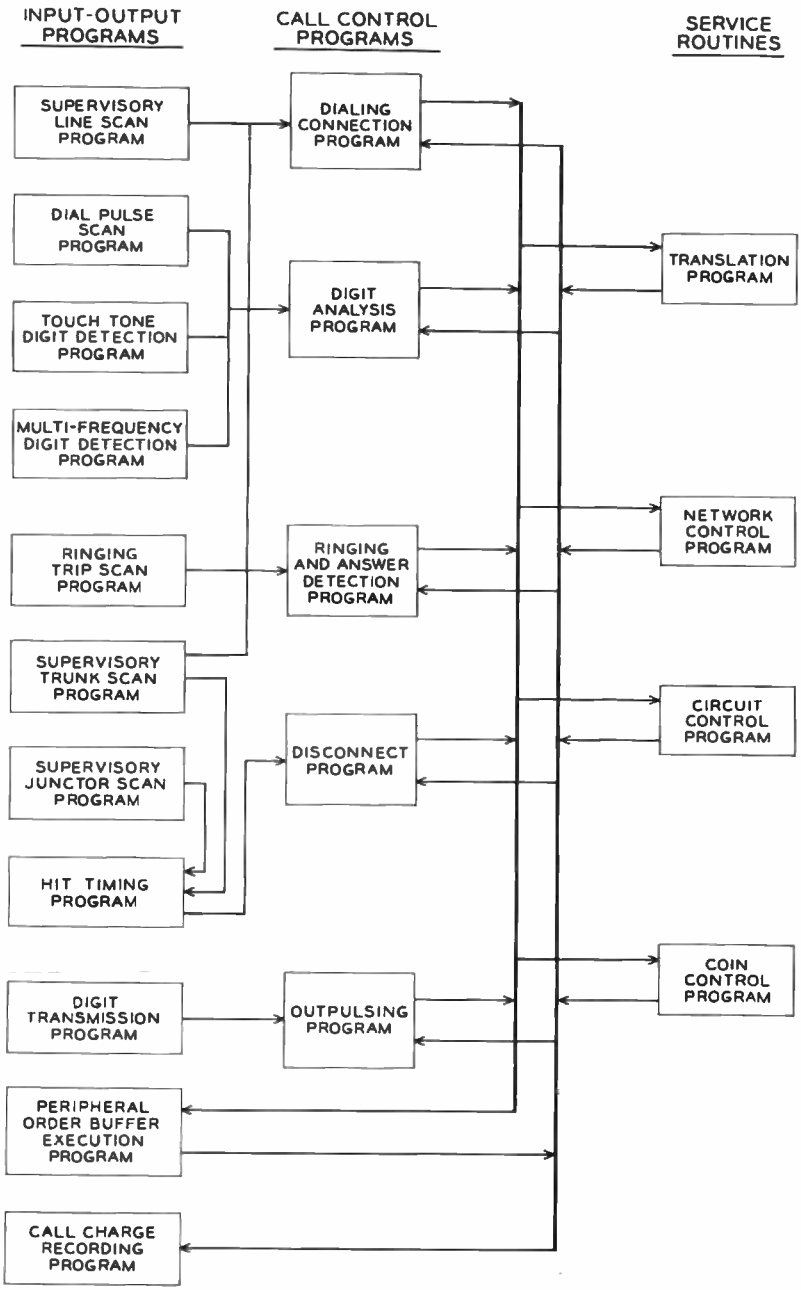


Fig. 13 — Programs used in processing calls.

IV. TEMPORARY MEMORY

Associated with the input-output programs, the call control programs, and the service routines, there are always data which change with the activity on a line or trunk, or with the progress of a call. These changeable data are kept in a temporary memory called the call store. Just as functional blocks of programs may be classified as being associated with system inputs and outputs, with call processing, or with service routines, so blocks of temporary memory may be associated with the same functions. The proper recording of information in temporary memory provides the means by which the various parts of a call are linked together and continuity of the call is maintained.

4.1 *Input-Output Oriented Memory*

Temporary memory is associated with each scan point of a line, junctor or trunk scanner in order to keep a record of the supervisory state of the facility connected to it. This memory is used primarily by the supervisory programs in detecting requests for service, disconnects, and answers. Since only service requests can be detected at a line ferrod (it is disconnected by cutoff contacts when a digit receiver is attached and not restored until disconnect), one line supervisory memory bit is associated with each line ferrod. The supervisory line scan program simultaneously reads the ferrod and the line supervisory memory, compares them and, if the line supervisory memory reads idle while the ferrod shows an off-hook condition, deduces that the customer has lifted his receiver. The scan program reports this to the dialing connection program. In the process, the line supervisory memory is marked to indicate that further readings of the bit should be ignored. In this sense, the line supervisory memory serves as a memory of the previous scanner reading and also as a busy or idle indicator for the line.

Each junctor circuit has two ferrods which supervise the two parties during talking. The customers can disconnect from the talking state only when connected to a junctor circuit; thus only one bit of temporary memory is needed for each junctor scan point. The junctor supervisory memory serves a purpose similar to the line supervisory memory in indicating the previous scanner reading.

Trunk supervisory points are multifunctional in that originations, answers, and disconnects are signaled to the system by them. Different trunk circuits and service circuits that have varying numbers of scan points appear on the same scanner in a pattern that changes from office to office. For these reasons, each scan point of a trunk scanner is assigned

two bits of temporary memory. These two bits are used to indicate: (1) that the facility is idle and may originate a request for service; (2) that the facility is in a talking state and should be monitored for disconnect; or (3) that the scan point may be changing for other reasons, such as signaling, and that the supervisory trunk scan should ignore any changes that it sees.

The digit receivers and digit transmitters each have associated temporary memory, which may be thought of as an extension of the circuits themselves. For instance, a relay in the dial pulse receiver is capable of following dial pulses. The relay activates the same scan point on each dial pulse. The dial pulse scan program, in conjunction with the temporary memory associated with the receiver, keeps a count of the pulses as they are received. Then, at the end of an interdigital time-out interval, the digit is reported to the digit analysis program, which counts and stores the digit in temporary memory. This is analogous in crossbar systems to a pulsing relay operating a set of counting relays and transfer of the pulse count at interdigital timeout to a set of register relays, steered by a digit counting circuit. Provisions for permanent signal and partial dial timing are also made, both in the No. 1 ESS and in crossbar systems.

A number of calls are being dialed into, or pulsed out of, the system at any one time. Therefore, there must be a way to keep a particular digit receiver and its temporary memory associated with a particular call. A call control register serves as the controlling register for the call. Its address is stored in the dialing register associated with the digit receiver serving the call. The dialing register is often referred to as a junior register.

Similar registers, differing slightly in use and format, are associated with TOUCH-TONE receivers, multifrequency transmitters and receivers, revertive pulse transmitters and receivers, and panel call indicator transmitters.

Another register closely associated with input programs is the hit timing register, which assists in timing disconnects to insure that hits do not cause false disconnects. This register also assists in performing the functions of timing for flashes and called party disconnects, as a service for the disconnect program.

4.2 *Hoppers*

Because the number of input-output actions is very large in wire centers served by No. 1 ESS, and because the system must meet very

tight timing requirements for certain of its input-output programs, it is necessary to limit the continuous length of processing time devoted to any given input. When a given input is detected that requires the attention of a call control program, the input program reports its finding in an area of call store called a hopper. Each input-output program is assigned one or more hoppers in which it stores information. A call control program, scheduled at a later time by the system's executive program, removes the information from the hopper and uses it to advance the call with which the information is associated. This permits the high-priority input-output program to concentrate on interrogating as many lines, trunks, digit receivers, or transmitters in as short a time as possible. The program has only to load some data in a hopper and continue, rather than interrupt its primary input-output function to perform more complex and less urgent tasks.

Since the data reported by the input-output programs are associated with a particular call, the identity of the call is stored in the hopper along with the data. The line supervisory scan program, for instance, reports to the dialing connection program through the service request hopper. The information stored in the hopper is the line equipment number, the only identification available at this stage of a call.

The dial pulse detection program, however, reports to the digit analysis program through three different hoppers, each reporting a different event: the dial pulse digit hopper reports digit counts and call abandonments; the remove dial tone hopper reports the receipt of the first digit; the permanent signal and partial dial hopper reports the named events. In each case, the identifying datum is the originating register address.

A partial list of hoppers, the names of which are self-explanatory, follows:

line service request hopper
dial pulse digit hopper
TOUCH-TONE digit hopper
multifrequency digit hopper
revertive pulse digit hopper
remove dial tone hopper
permanent signal, partial dial hopper
ringing trip hopper
junctor disconnect hopper
trunk disconnect hopper
peripheral order buffer execution result hopper
request outpulsing digit hopper
request teletypewriter character hopper.

4.3 *Output Buffers*

The peripheral order buffer execution program has been described as one of the principal means by which No. 1 ESS controls the switching network and various trunk, junctor, and service circuits. The peripheral order buffer is the temporary memory area used to list the network controller, signal distributor, central pulse distributor, and scanner actions necessary to accomplish a desired result. The number of these areas is variable and dependent upon the expected traffic load in the office.

In addition to the list of instructions to be carried out, the POB contains a cross reference to the call control register for which it is performing a service and to one or more program addresses to which it returns to report success or failure.

Similar in character and in operation to the POB are the buffers provided for transmitting teletypewriter messages and for recording call charge information on magnetic tape. The characters for a teletypewriter message are placed sequentially in a teletypewriter buffer and then are removed and converted to teletypewriter code, one character at a time, at a rate compatible with the transmitter. Call charge information is placed on magnetic tape in blocks of 500 characters. A buffer to accommodate this function is furnished. When a block of 500 characters is filled, an unloading program transfers the data to tape.

The output buffers for digits to be outpulsed to a distant office are those registers which are associated with digit transmitters and which are used by the input-output programs.

4.4 *Call Control Registers*

Call-processing registers are blocks of temporary memory used by call control programs to store data during a particular stage of the progress of a call. The information needed by different functions is not necessarily the same in content or amount; therefore, call control registers associated with different functions will differ in size.

All call control registers have a standard format. The first four words (a word consists of 23 bits) of any call control register are used for specific storage purposes. The first word of a call register contains the identity of the register and call state; this word is referred to as the state identifier. The second, or queue, word is used to insert the register on a waiting or a timing list. The third word is called the link word, used to hold another call control register's address if more than one register is associated with the call. The fourth, or scan, word contains the address of a scan register

if the call control program has requested scanning of some particular point or points. The next group of words in any call control register is set aside for network path memory. This storage area is referred to as the path memory annex and may differ in length from one type of call control register to another. The storage area following the path memory annex area is called the data area of the register, and its size is dependent on the data needed by the associated call control programs.

A brief description of the major call control registers used in the No. 1 ESS follows.

The originating register is used by the dialing connection and digit analysis programs. The path memory annex consists of two words and contains the path memory necessary for the connection from the originating line to the digit receiver. The data area of the register must hold the address and the class information of the originating line, the dialed directory number, the digit count, and other program control information.

The ringing register is used as a storage medium by the ringing and answer detection program. It is held for the period from the completion of dialing through the establishment of the talking connection. The path memory annex of the ringing register consists of seven words and contains the path memory necessary for the connections from the originating line to an audible tone trunk, from the terminating line to the ringing trunk, and the reserved path from the originating line to the terminating line. The data area contains class information of the terminating line, a condensed version of the originating line class, and program control information.

At the completion of dialing, if the call is outgoing, the originating register is changed to an outpulsing register. The digit analysis program does this by changing the state identifier of the register. The program also seizes a block of temporary memory and links this to the newly designated outpulsing register. The address of this block of temporary memory is stored in the first word of the path memory annex of the outpulsing register. The network path memory of (1) the connected path from the connections from the originating line to the digit receiver, (2) the connected path from the transmitter to the outgoing trunk, and (3) the reserved path from the line to the outgoing trunk is stored in this five-word temporary memory block. The data area of the outpulsing register contains the originating class information, dialed directory number, a route index specifying the first-choice outgoing route, and program control data.

The disconnect register is seized upon recognition of a disconnect and

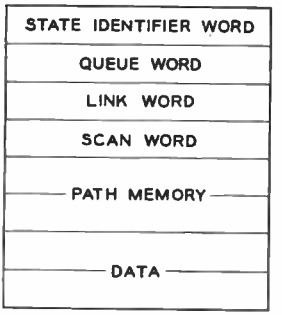


Fig. 14 — Typical call control register format.

is used by the disconnect program for timing and scanning. The path memory annex of the disconnect register consists of three words and contains the path information for a line-to-line, or line-to-trunk, connection. The data portion contains program control information.

The linking of call control registers is a one-way circular link. That is, if call control registers A, B, and C are associated with a call at a particular time, then register A contains register B's address in its link word; register B contains C's address; and register C contains A's address. Should a call program controlling this call decide to release register B, then the linkage is updated such that register A contains register C's address and register C contains A's address. Fig. 14 shows a typical call control register layout.

4.5 *Service Routine Registers*

A wide variety of registers exist to store data for use by service routine programs. The call charge register, peripheral order buffer, and coin control register are examples of service routine registers. The size and layout of these registers vary markedly.

The call charge record register has a layout identical to that of a call control register. The path memory annex of the call charge register is three words in length. The linking between a call charge register and a call control register is implemented in the same manner as the linking between call control registers.

The coin control register and peripheral order buffer contain primarily call control data and do not have the standard call control register layout, nor are they linked to a call control register in the previously described manner. They are used exclusively by the associated service routine program.

4.6 *Network Memory*

A portion of call store is reserved for the purpose of recording the busy-idle condition of the network links and junctors. This memory block is commonly referred to as the network map and is used by the network program in establishing or reserving network connections. In addition to the network map, a word of call store memory is assigned for every junctor terminal appearance on a line link network and for every trunk terminal appearance on a trunk link network. There is a direct relationship between a terminal and the address of its associated word. These words for junctor and trunk terminals are commonly referred to as path memory for lines and path memory for trunks, respectively.

The information in the path memory for trunks, together with the trunk network number, uniquely specifies a junctor's terminal and the network links used in the trunk link network. Translation and conversion programs facilitate the derivation of the address associated with the junctor terminal in the line link network. The network number of the line junctor terminal, together with the contents of the path memory for lines, uniquely specifies the line and the links used in the line link network.

When no call control register is associated with a line-to-trunk connection, the path memory for lines contains the line equipment number of the No. 1 ESS customer and some control information specifying originating or terminating line, special treatment, coin line, and flash permission. The path memory for trunks contains a junctor network number and control information specifying originating or terminating terminal, and flash permission for the No. 1 ESS customer.

When no call register is needed on a line-to-line connection, each line junctor terminal word contains a customer's line equipment identification and control information. Translation programs facilitate the derivation of the address of either junctor terminal, given the other end.

When a call control register is associated with a connection, the address of the register is placed in the word, which usually holds the path memory for trunks on a line-to-trunk connection. On a line-to-line connection, the address of the register is placed in the words which normally hold the path memory for lines. In both cases, the displaced path memory is stored in the call control register in the area called the path memory annex.

4.7 *Recent-Change Register*

Since the translation tables are stored in the program store, which is not changeable by the program during normal operation, an area of

temporary memory is set aside to provide an alternate location for the day-to-day administration of line number changes, new connections, disconnected lines, routing changes, etc. When this recent-change area is filled, the data are transferred to the program store's twistor cards by replacing them with new cards written in the program store card writer, a part of the master control center.

The translation program searches the recent-change register for a requested translation before going to the program store's tables, because there is a possibility that a change might have been made on the data.

Fig. 15 shows the principal blocks of temporary memory used for call processing and the flow of information from one to another.

V. PROCESSING AN INTRAOFFICE CALL

5.1 *Detection of Origination*

Fig. 16 demonstrates the role played by the program, the memory, and the equipment involved when a request for service is initiated. The supervisory line scan program examines the line ferrod sensor condition, with its associated line supervisory memory, approximately ten times per second. Upon detecting a line ferrod sensor in a saturated state and its line supervisory memory in an idle state, the program recognizes an origination. The program enters the line equipment number (LEN) in the line service request hopper.

5.2 *Connection of Line to Digit Receiver*

The first action of the dialing connection program, as illustrated in Fig. 17, is to examine the line service request hopper for an entry. Upon detecting an entry, the dialing connection program seizes an originating register and stores the customer's line equipment number in it (from the line service request hopper). After storing the line equipment number, the dialing connection program transfers control to the translation program, giving it the calling customer's line equipment number. The translation program returns with the originating line class information, consisting of: (1) TOUCH-TONE or dial pulse receiver required; (2) class of line (manual, individual, party, coin); (3) type of service (flat rate, message rate). The dialing connection program stores the originating class information in the originating register, then seizes and initializes a peripheral order buffer (POB). The dialing connection program next transfers control to the network control program, giving it the customer's line equipment number, the originating register's address, and a code for the type of receiver.

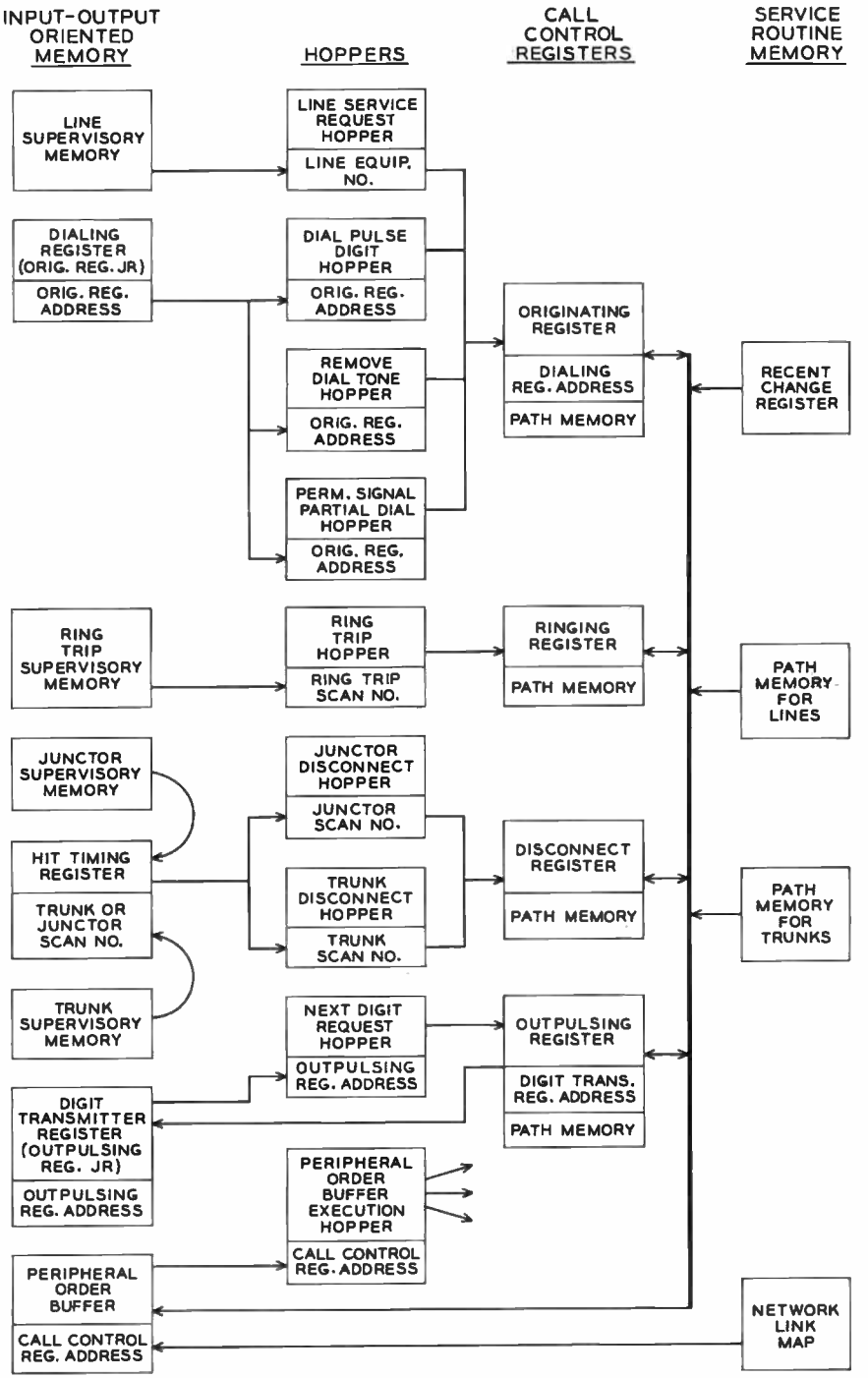


Fig. 15 — Temporary memory used for processing calls.

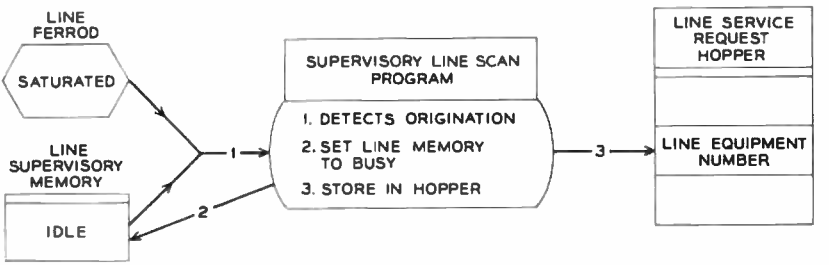


Fig. 16 — Detection of origination.

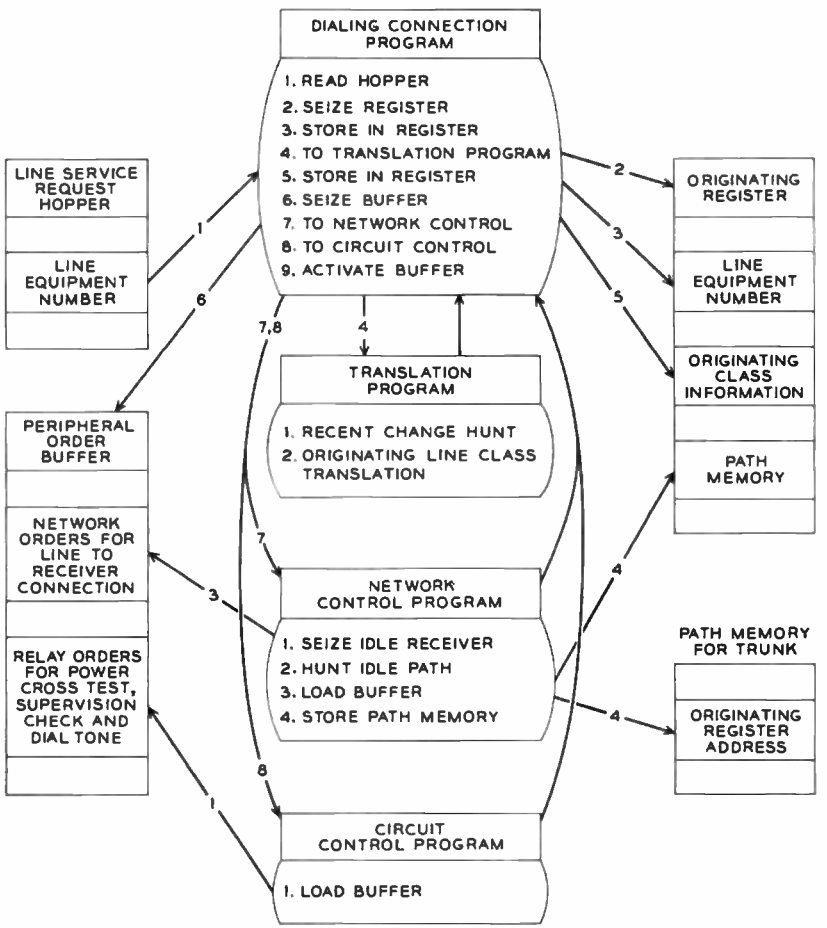


Fig. 17 — Initial actions for connection of line-to-digit receiver.

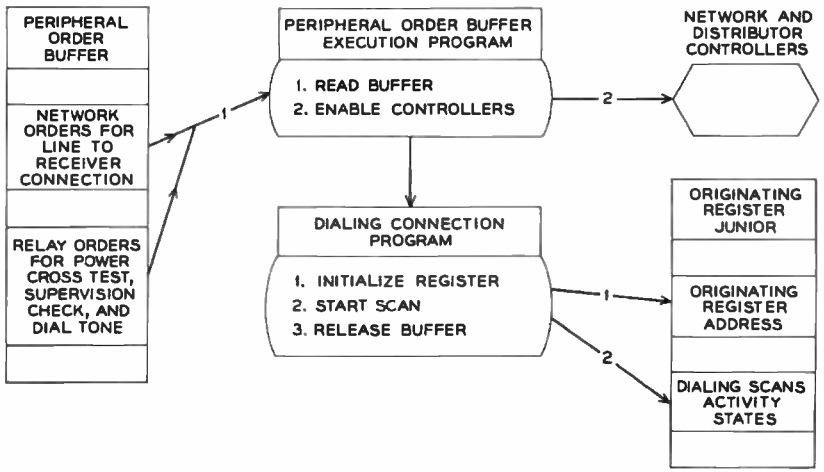


Fig. 18 — Final actions for connection of line-to-digit receiver.

The network control program loads the POB with instructions to establish the network connection between the customer's line and the receiver's trunk network number. The circuit control program loads the relay and scan operations required to perform the power cross test, transfer of supervision check, and application of dial tone. After completion of the loading, program control is returned to the dialing connection program which activates the POB.

As shown in Fig. 18, the POB execution program causes the instructions in the POB to be carried out. The successful execution of these instructions establishes a configuration as shown in Fig. 19; the originating line is connected to the digit receiver by a path through the line link and trunk link networks. Dial tone is applied to the originating

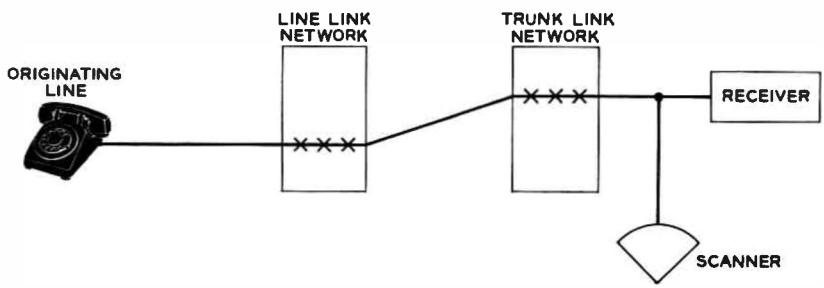


Fig. 19 — Receiver connection.

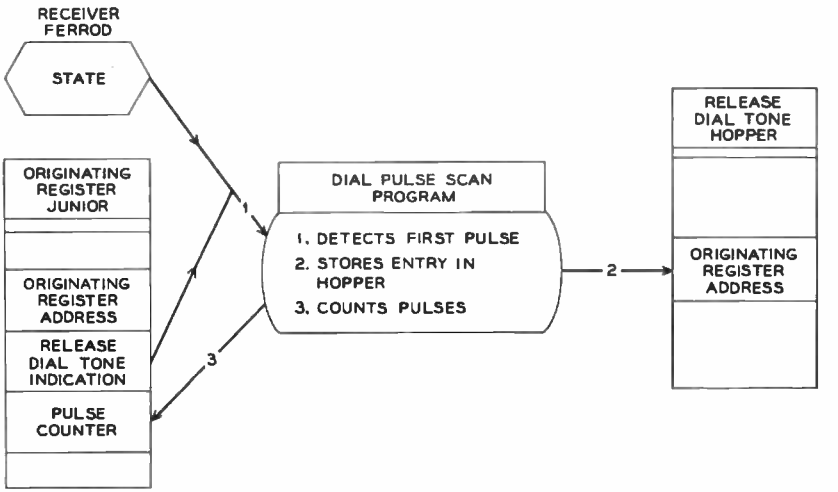


Fig. 20 — Detection of first pulse.

line, and supervision for abandonment is performed at the digit receiver. Upon successful execution of the instructions in the POB (as shown in Fig. 18), control returns to the dialing connection program. The dialing connection program releases the POB, stores the originating register address in the originating register junior, and sets indications in the originating register junior which initiate scanning for dial pulses, abandons, interdigital, permanent signal, and partial dial timing.

5.3 *Digit Analysis*

5.3.1 *Release of Dial Tone*

As shown in Fig. 20, the dial pulse detection program examines the ferrod associated with the pulsing relay of the receiver, and the originating register junior, to count dial pulses and to recognize the first pulse of the first digit. When the first pulse of the first digit is detected, the dial pulse scan program stores the originating address in the remove dial tone hopper and increments the pulse counter.

The digit analysis program (as shown in Fig. 21) regularly examines the remove dial tone hopper. On detecting an entry, the digit analysis program seizes and initializes a POB. The digit analysis program then transfers control to the circuit control program, giving it the receiver's trunk network number. The circuit control program loads the necessary

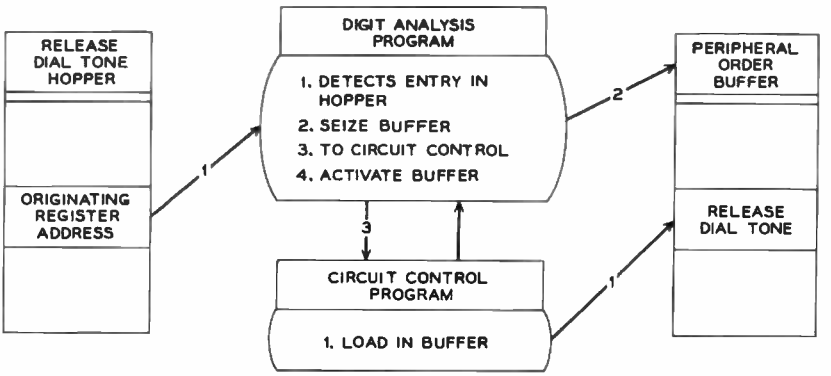


Fig. 21 — Initial actions for the release of dial tone.

instructions in the POB to release dial tone and then returns to the digit analysis program, which then activates the POB.

Upon successful execution of the POB (as shown in Fig. 22), control returns to the digit analysis program. This program marks a dial tone released indication in the originating register and releases the POB.

5.3.2 *Reception of Digits*

As shown in Fig. 23, if the receiver ferrod remains in a saturated state for a period of 120 to 240 milliseconds, the abandon-interdigital timing

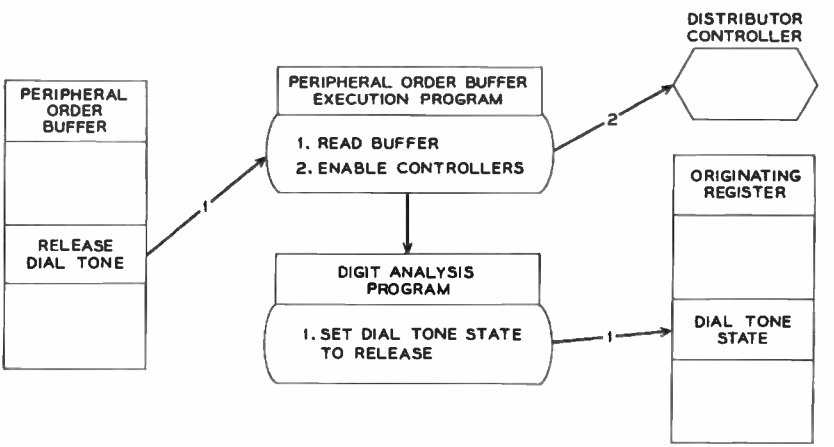


Fig. 22 — Final actions for the release of dial tone.

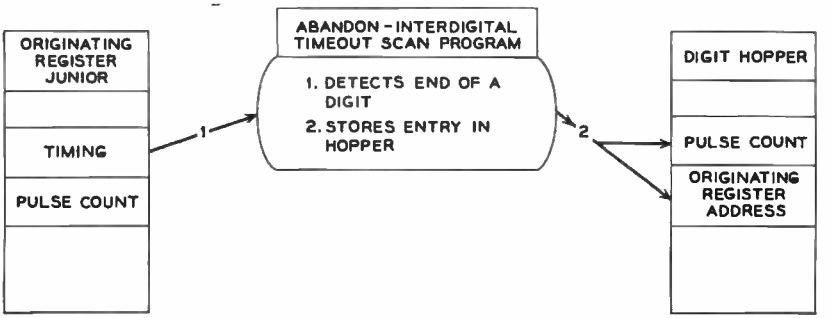


Fig. 23 — Reception of digits.

program determines that this is the end of a digit. The program interrogates the originating register junior for the pulse count and the originating register address and stores both items in the dial pulse digit hopper.

The digit analysis program (as shown in Fig. 24), scheduled by the executive control program, detects the entry in the dial pulse digit hopper, stores the pulse count in the originating register, and increments a digit counter. When the digit analysis program recognizes that the third digit has been received, it delivers the first three digits to the translation program. The translation program returns the following three-digit class information to the digit analysis program: (1) special service code dialed; (2) invalid code dialed; (3) interoffice code dialed,

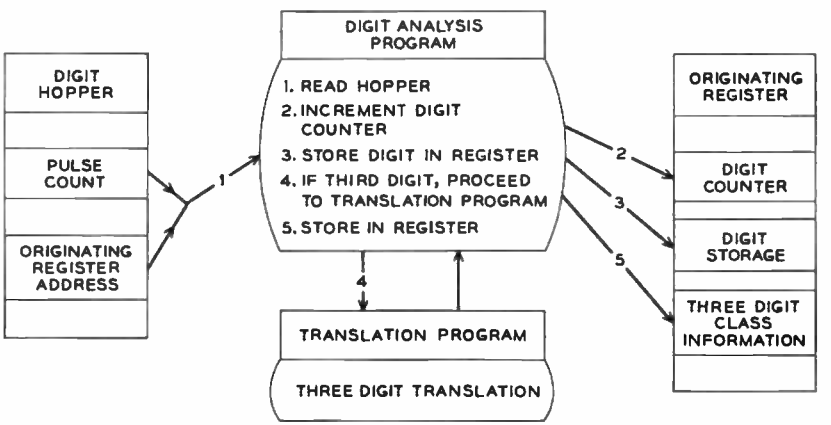


Fig. 24 — Analysis of third digit.

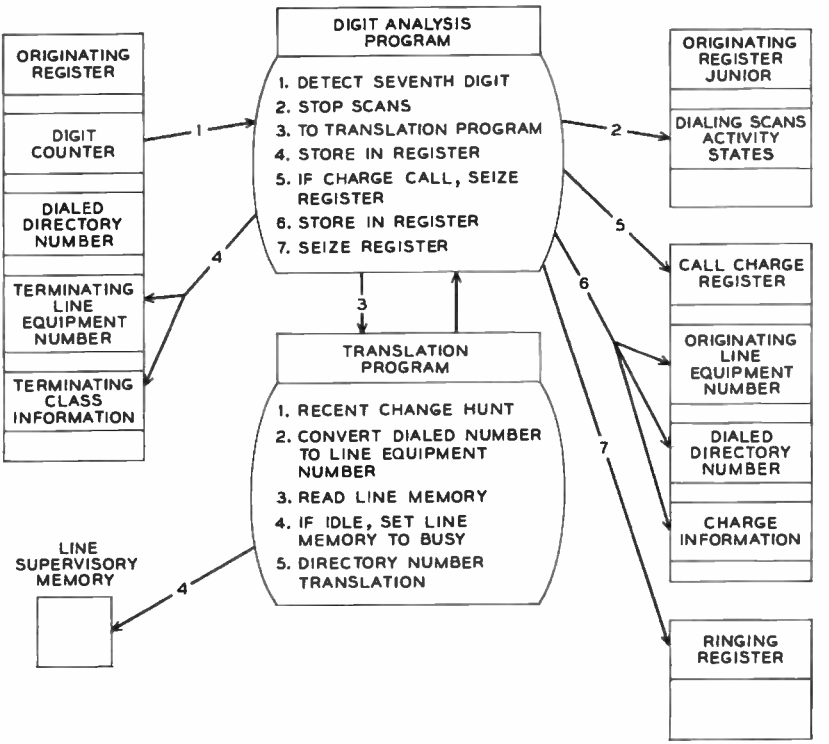


Fig. 25 — Analysis after end of dialing.

expect seven digits; (4) interoffice code dialed, expect ten digits; (5) intraoffice code dialed; and (6) charge call or free call.

5.3.3 *End of Dialing*

On the intraoffice call of this example, when the digit analysis program determines that the entry in the digit hopper is the seventh digit (as shown in Fig. 25), it shuts off the scan programs that detect abandon-interdigital time-outs, dial pulses, permanent signals, and partial dials by properly marking the originating register junior. The digit analysis program transfers to the translation program and requests a directory number translation. The translation program converts the directory number to a terminating line equipment number and terminating class. The translation program examines the terminating line's supervisory memory to see whether the line is busy. If idle, it marks the line super-

visory memory busy and then extracts the directory number translation. The directory number class information consists of: (1) busy line found, (2) invalid number, (3) idle line found, (4) busy but special treatment, (5) temporary transfer activated, and (6) trunk group found. The translation program places this information in its buffer memory so that it can be passed to the digit analysis program, which in turn stores this information in the originating register. The digit analysis program determines whether it is a chargeable call. If chargeable, a call charge register is seized, initialized, and linked to the originating register. The digit analysis program stores in the call charge register the originating line equipment number, charge information, and the dialed directory number. The digit analysis program then seizes and initializes a ringing register, links the ringing register with the originating register and call charge register, and then transfers control to the ringing and answer detection program.

5.4 *Ringing and Answer Detection*

5.4.1 *Establishing the Ringing Connection*

The ringing portion of the ringing end answer detection program (as shown in Fig. 26) seizes and initializes a POB and transfers to the network control program after giving it the originating register and ringing register addresses. The network control program idles the receiver, seizes idle ringing and audible ringing tone circuits, hunts an idle path from originating line to the audible tone circuit, hunts an idle path from the terminating line to the ringing circuit, reserves a talking path between the originating line and the terminating line, and loads the POB with instructions to establish two connections. The first connection is from the originating line to the audible ringing tone circuit. The second is from the terminating line to the ringing circuit. It also stores the path memory specifying the above network configuration in the ringing register path memory annex. The network control program returns control to the ringing program, which requests the circuit control program to load the POB with instructions to control relays and scan actions in the two circuits. On return, the ringing program releases the originating register and activates the POB.

As shown in Fig. 27, after the POB execution program successfully executes the instructions, control returns to the ringing program. The final actions in establishing a ringing connection are to set the audible tone circuit supervisory memory in a state such that the trunk super-

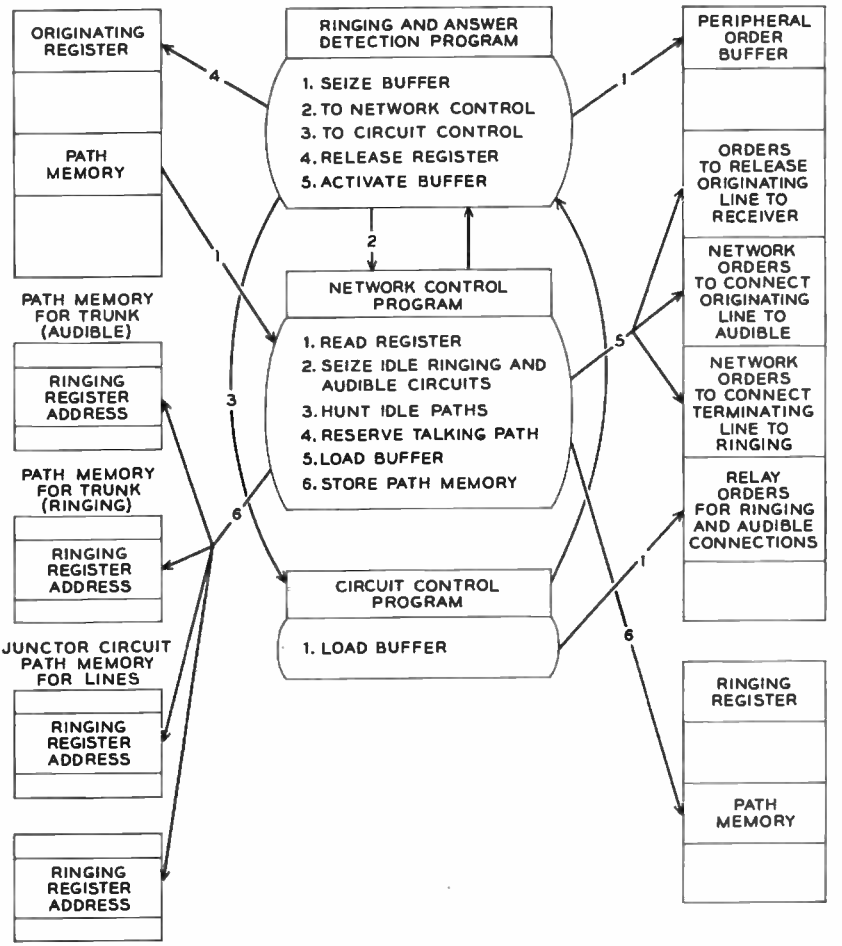


Fig. 26 — Initial actions for the ringing and audible connection.

visory scan program will detect a disconnect and the ringing circuit supervisory memory in a state such that the ringing trip scan will detect an answer by the called customer.

The successful execution of the POB actions in setting up ringing establishes the configuration shown in Fig. 28. The originating line is connected to an audible ringing tone circuit through the line link and trunk link networks; the terminating line is connected to the ringing circuit by a connection through the line link and trunk link networks; and a path through the line link network to a junctor circuit is reserved between the originating and terminating line. Supervision of the origi-

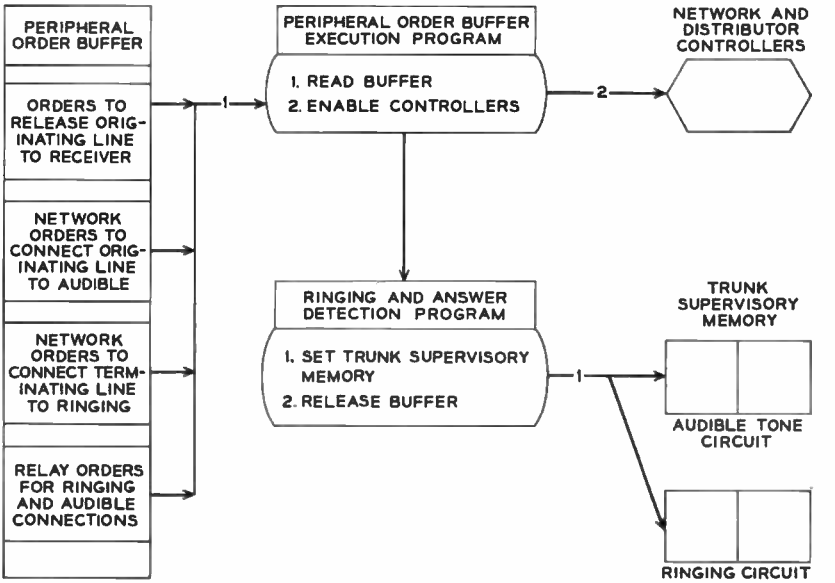


Fig. 27 — Final actions for the ringing and audible connection.

nating line is maintained at the audible ringing tone circuit, and the terminating line is supervised at the ringing circuit.

5.4.2 *Answer Detection*

The ring trip scan program (as shown in Fig. 29) examines the ring trip ferrod and the ringing circuit supervisory memory ten times per second. If the ring trip scan program detects the ring trip ferrod in an

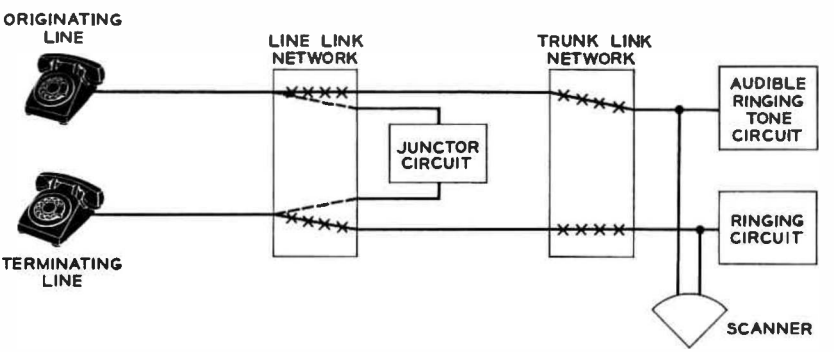


Fig. 28 — Ringing and audible connection.

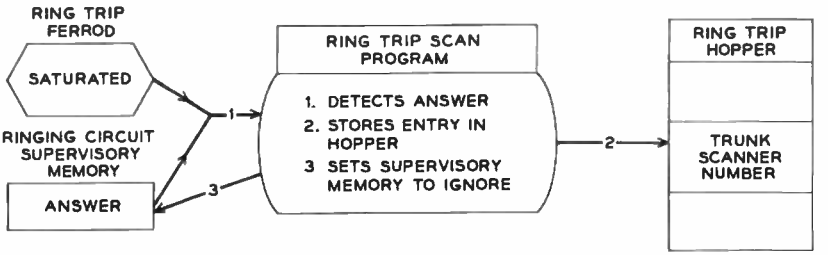


Fig. 29 — Detection of answer.

off-hook state and the ringing supervisory memory in an active state, it recognizes this condition as an answer. The ring trip scan program sets the ringing circuit supervisory memory to the ignore state and loads the trunk scanner number in the ring trip hopper. The answer detection part of the ringing and answer detection program (as shown in Fig. 30), when scheduled by the executive control program, examines the ring trip hopper. Upon detecting an entry, the answer detection program delivers the trunk scanner number to the translation program, which translates the trunk scanner number to its trunk network number and to the address of its associated path memory word. This information is then made available to the answer detection program. It seizes and initializes a POB and sets the audible trunk supervisory memory to the "ignore" state. It then transfers to the network control program, giving it the addresses of the ringing and call charge registers.

The network control program loads the POB with the instructions necessary to release the connection between the originating line and the audible ringing tone circuit, to release the connection between the terminating line and the ringing circuit, and to establish a talking path between the originating and terminating lines. The network control program also (1) idles the ringing and audible tone circuits (2), loads information associated with the talking path in the call charge register's path memory annex, and (3) places the address of this call charge register in the junctor's path memory words. The network control program then returns to the answer detection program, which requests the circuit control program to load the necessary relay and scan instructions to close the talking path. Then the answer detection program activates the POB.

As shown in Fig. 31, after the instructions stored in the POB are carried out, the answer detection program sets both sides of the junctor supervisory memory to busy, releases the POB, releases the ringing

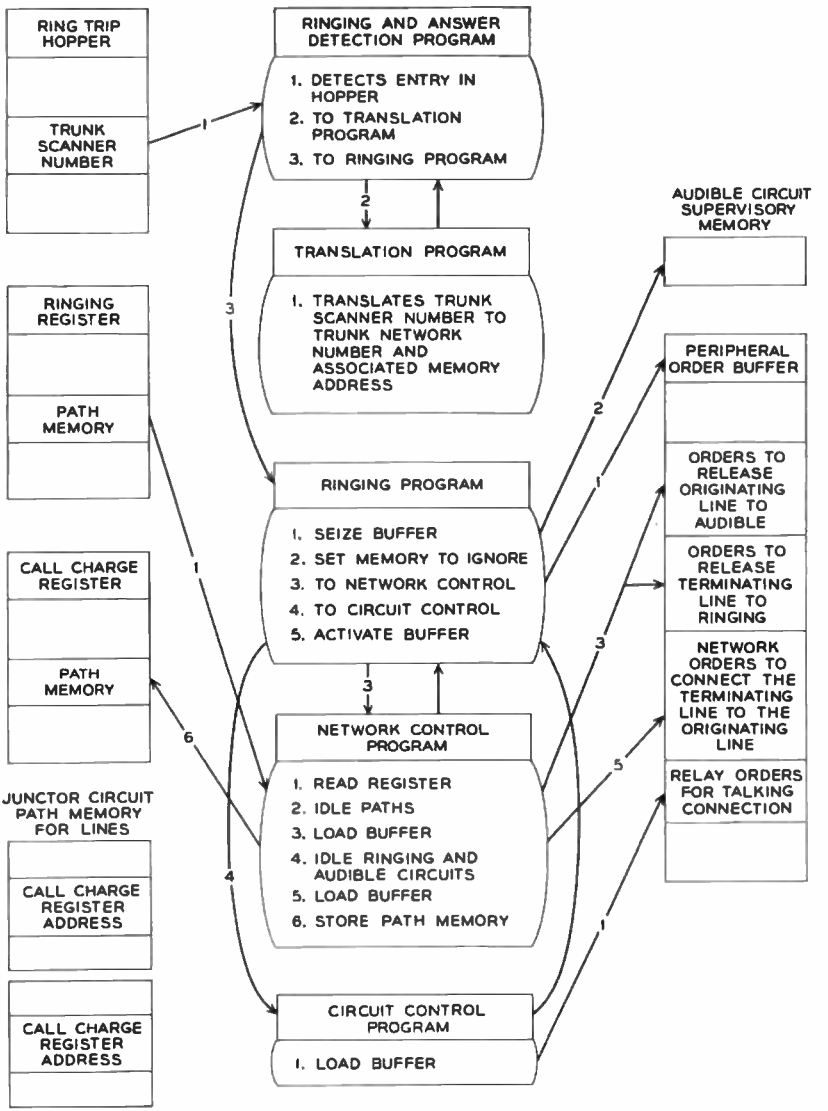


Fig. 30 — Initial actions for the talking connection.

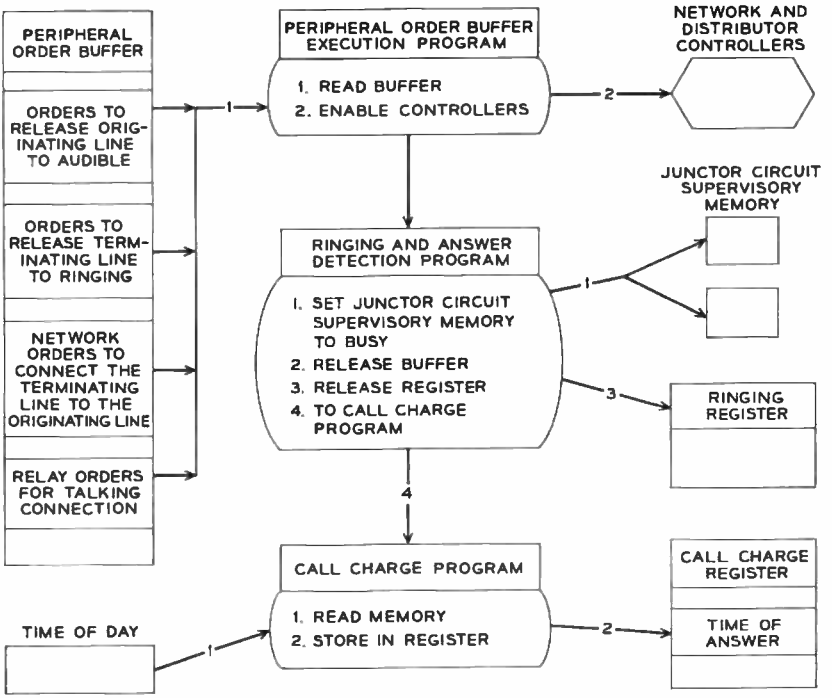


Fig. 31 — Final actions for the talking connection.

register, and transfers to the call charge program. The call charge program enters the time of answer (after reading the time of day) in the call charge register.

The talking connection between originating and terminating lines is established (as shown in Fig. 32) through the line link network. Super-

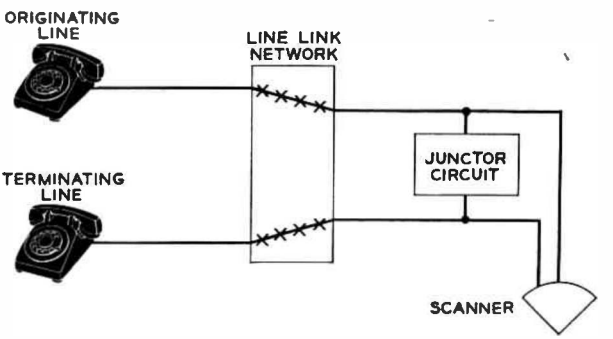


Fig. 32 — Talking connection.

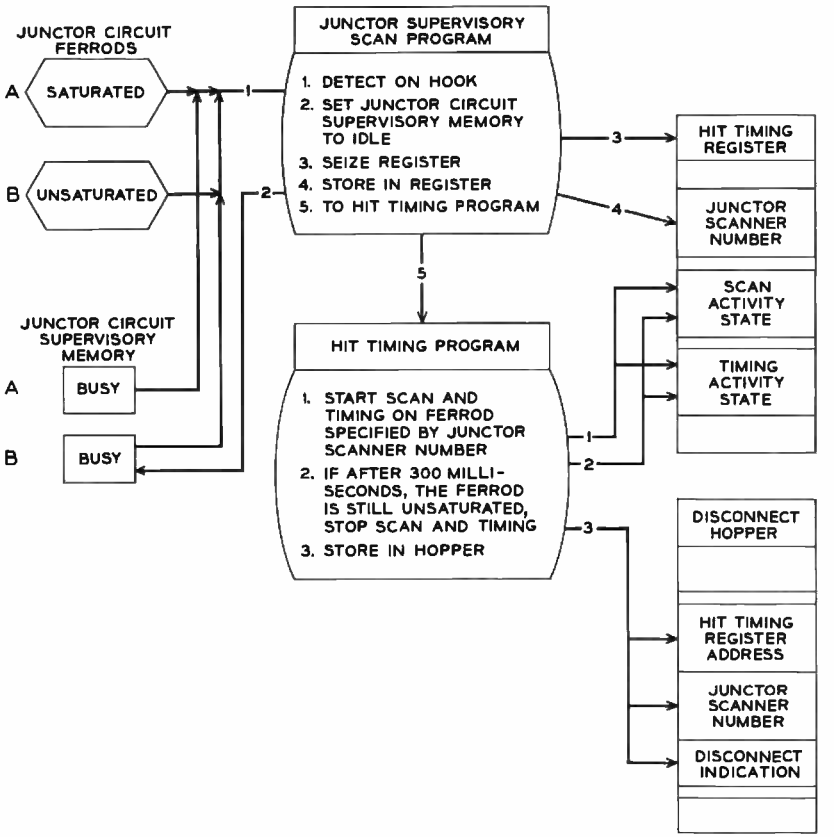


Fig. 33 — Detection of on-hook and hit timing.

vision of the originating and terminating lines takes place at the junctor circuit. The call charge register is in control of the call up to the time of disconnect.

5.5 *Disconnect*

Ten times per second the junctor supervisory scan program (as shown in Fig. 33) examines the state of the junctor ferrod and the junctor supervisory memory. If the junctor ferrod is unsaturated and the junctor supervisory memory (side B, Fig. 33) is in the busy state, the junctor supervisory scan program sets the junctor supervisory memory (side B) in the idle state. The junctor supervisory scan program then seizes and initializes a hit-timing register by placing the junctor scanner

number in the hit-timing register to start a directed scan of the junctor ferrod.

Three hundred milliseconds after the initial entry is stored in the hit-timing register, the hit-timing program reads the ferrod specified by the scanner number, and, if the ferrod still indicates on-hook, it recognizes a disconnect. The hit-timing program enters the junctor scanner number, the hit-timing register address, and a disconnect indication in the junctor disconnect hopper.

Later, the disconnect program (as shown in Fig. 34) removes the entry from the junctor disconnect hopper and transfers to the translation program after giving it the junctor scanner number. The transla-

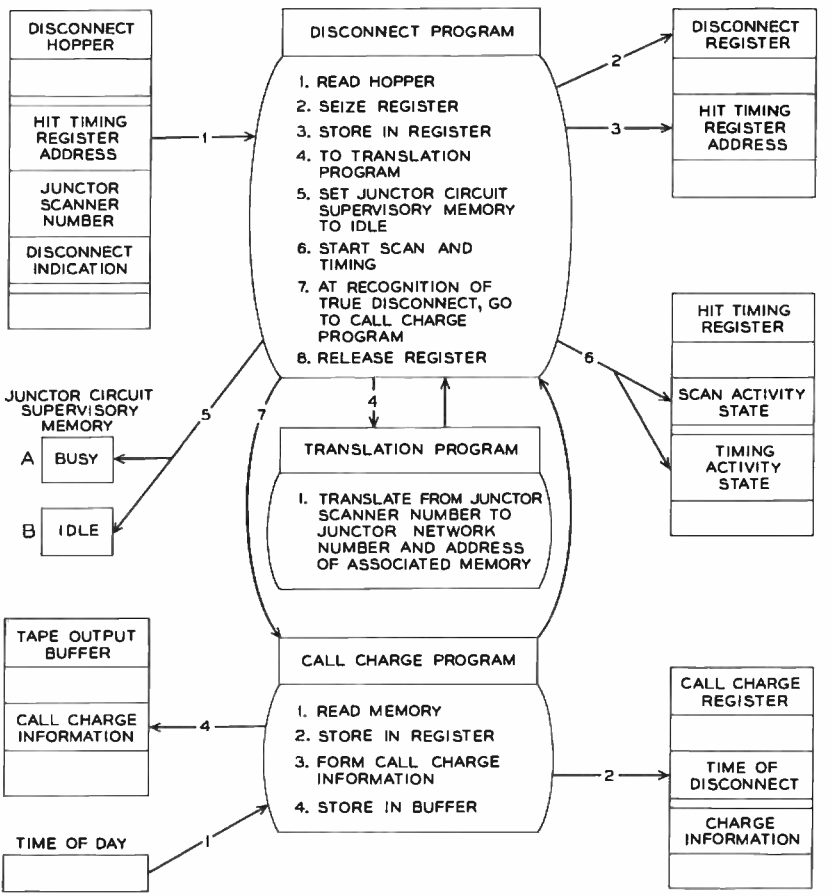


Fig. 34 — Detection of disconnect and charging actions.

tion program translates the junctor scanner number to the network numbers of the junctor terminals and obtains the addresses of the associated path memory. The translation program then returns these to the junctor disconnect program. The disconnect program seizes a disconnect register, stores in it the junctor network numbers and the hit-timing register address, sets the junctor supervisory memory (side A, Fig. 34) to the idle state, and links a disconnect register with the call charge register. The call charge then records disconnect time in the call charge register. From the call charge register the disconnect program determines that the originating customer has disconnected. The disconnect program also initiates directed scans of the both junctor circuit ferrods. If the called customer disconnects or the calling customer reoriginates before 10 to 12 seconds have elapsed, the disconnect program stops the directed scans, releases the hit-timing register and transfers to the call charge program after giving it the call charge register address and a disconnect indication.

The call charge program loads the charge information into the tape output buffer. The call charge program returns control to the disconnect program. As shown in Fig. 35, the disconnect program seizes and initial-

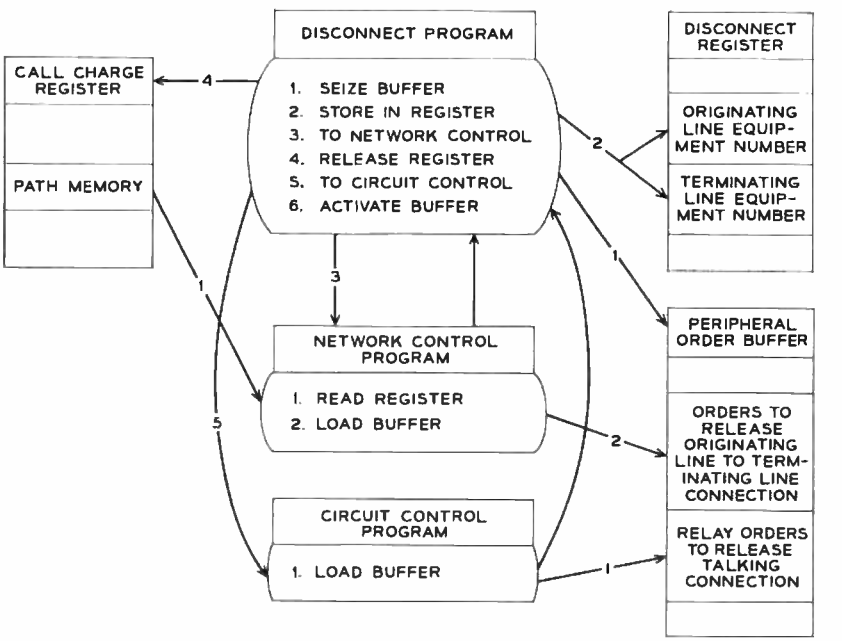


Fig. 35 — Initial disconnect actions.

izes a POB and transfers control to the network control program, giving it the call charge register address. The network control program loads the POB with the network actions necessary for restoration of the line ferrod. The network control program then returns to the disconnect program, which transfers to the circuit control program for the loading of the necessary relay actions. On return, the disconnect program releases the call charge register and activates the POB.

As shown in Fig. 36, after the relay orders stored in the POB are carried out, the disconnect program releases the POB and delivers to the translation program the originating and terminating line equipment numbers. The translation program converts the line equipment numbers to their line memory addresses and makes these available to the disconnect program. The junctor disconnect program sets the originating and terminating line supervisory memory to the idle state and releases the disconnect register.

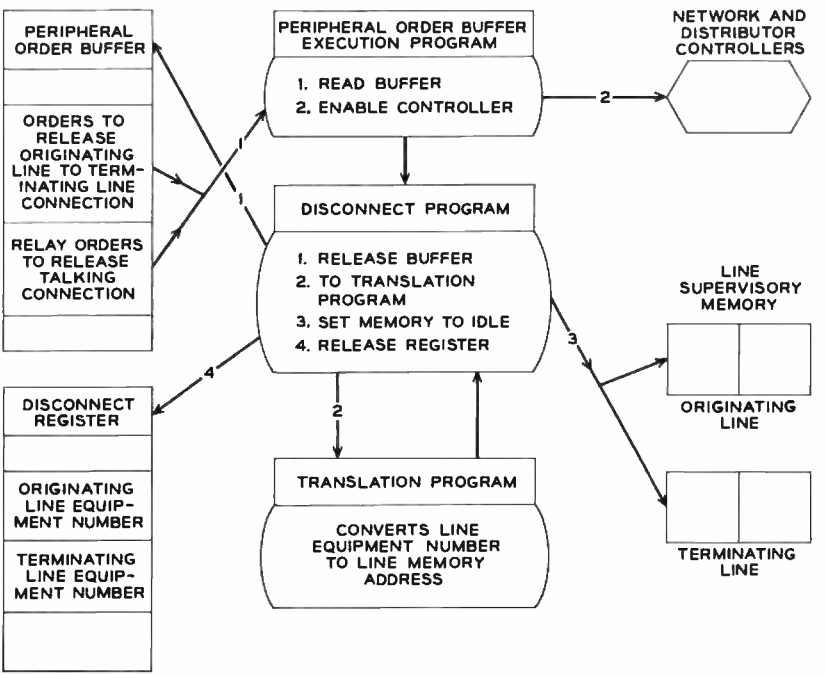


Fig. 36 — Final disconnect actions.

VI. CONCLUSION

The No. 1 ESS solves the problem of interconnecting telephone customers by centralizing the decision-making and the memory required to process telephone calls in an electronic data processor. As a result, trunk and service circuits have been greatly simplified.

The foregoing pages have described the manner in which the program, aided by circuits and by temporary memory, processes telephone calls. A simple intraoffice call has been used to illustrate the procedure followed in processing a particular type of call. However, the No. 1 ESS must offer many other services; hence programs must be provided to process other types of calls. The basic framework described here is supplemented in order to offer the full range of modern telephone services.

Although the variety of telephone services and equipment results in a large program to control the system, the use of a stored program offers an economical means to accomplish many present switching tasks and a flexible means for accomplishing the numerous future switching functions.

VII. ACKNOWLEDGMENT

Many of our colleagues have contributed to the planning necessary to implement call processing in the No. 1 ESS. Many of these contributions are discussed in greater detail in the other papers included in this issue of the Bell System Technical Journal. In addition, the authors acknowledge the work of numerous members of the systems engineering, program system design, and programming organizations concerned with the No. 1 ESS.

REFERENCES

1. Keister, W., Ketchledge, R. W., and Vaughan, H. E., No. 1 ESS System Organization and Objectives, B.S.T.J., this issue, p. 1831.
2. Biddulph, R., Budlong, A. H., Casterline, R. C., Funk, D. L., and Goeller, L. F., Line, Trunk, Junctor, and Service Circuits in No. 1 ESS, B.S.T.J., this issue, p. 2321.
3. Feiner, A., and Hayward, W. S., No. 1 ESS Switching Network Plan, B.S.T.J., this issue, p. 2193.
4. Freimanis, L., Guercio, A. M., and May, H. F., No. 1 ESS Scanner, Signal Distributor, and Central Pulse Distributor, B.S.T.J., this issue, p. 2255.
5. Harr, J. A., Hoover, Mrs. E. S., and Smith, R. B., Organization of the No. 1 ESS Stored Program, B.S.T.J., this issue, p. 1923.
6. Ulrich, W., and Vellenzer, Mrs. H. M., Translations in the No. 1 Electronic Switching System, B.S.T.J., this issue, p. 2533.

Translations in the No. 1 Electronic Switching System

By W. ULRICH and MRS. H. M. VELLENZER

(Manuscript received January 13, 1964)

Translations are the means of storing and retrieving office and customer information in a No. 1 electronic switching system installation. The translation scheme must be sufficiently general to handle special telephone services in addition to the items stored as cross connections in present telephone systems. The categories of translation information, the techniques of storage, the means for specifying translation data, and means for making changes therein, are described in this article.

I. INTRODUCTION

The information recorded in the storage or memory units of a No. 1 ESS office consists of four parts:

(1) the transient information about telephone calls in progress and the present state of all lines and trunks in the office,

(2) a program for controlling all system operations that is basically identical in all offices,

(3) a parameter table in the program store (semipermanent storage unit) containing certain information which varies from office to office and which changes only when major additions are made to an office, and

(4) translation information, which includes the bulk of the information that varies from office to office, some of which changes from day to day.

This article is concerned with the translation information and the portion of the program that is used to fetch and administer this information.

The following are the basic categories of translation information, described for the moment in simple form.

(1) Originating line translation: this translation indicates any special treatment of an originating subscriber, and specifies the directory number to be charged for his calls.

(2) Terminating translation: this translation specifies which terminating line should be connected to a calling customer or trunk when a directory number is dialed by the customer or received from an incoming trunk.

(3) Trunk translations: in the No. 1 ESS, a trunk distributing frame allows a trunk circuit in any frame to be connected to any position on the No. 1 ESS switching network. This requires a flexible trunk equipment location to trunk network position translation, as well as the inverse. In addition, the nature and complexity of miscellaneous trunk circuits and the fact that these trunk circuits are served by general-purpose (master) scanners (to detect conditions within the circuit), and central pulse distributors and signal distributors (to change relay states within the circuit) require a translation to let the ESS program know which of these are connected to a particular trunk circuit.

In addition, it is necessary to know which trunks belong to which trunk groups, since a trunk is usually seized because it is an available member of a desired trunk group.

(4) Office code translations: The first three digits of a dialed number must be interpreted to check for validity, detect intraoffice calls, and to select a route for interoffice calls. For some 10-digit calls, a 6-digit translation is necessary.

(5) Routing and charging translations: a route, as derived by the office code translation, is only a route pattern; this must be interpreted to find out which trunk group to use for an interoffice call, and how many digits to pulse forward to the distant office. In case all the trunks in this group are busy, an alternate route must be provided. If a coin zone call is made from a coin telephone, an indication must be sent to the operator so that she may quote the charge; for other charge calls, a "billing index" is recorded on an automatic message accounting (AMA) record along with other details of the call so that the customer will be charged the correct amount for the call. In addition, routing provides information as to the proper disposal of calls to vacant office codes, misdialed calls, incompletely dialed calls, and other similar situations whose treatment differs in different applications of the system.

In summary, translations make it possible for a general-purpose telephone call control program[1] to function in any central office, by providing in a standard form information specific to this office concerning directory numbers, office codes, line and trunk equipments, and routing and charging procedures.

The nature of the information provided by the translations requires that translation data be changeable. Translation data are stored on

twistor magnet cards[2] in the program store. Current changes are received as messages from a teletypewriter (used as a major input to the system) and are then stored in the variable memory or call store; thereafter, they are periodically transcribed into the program store, using the program store card writer available as part of the master control center of every office.

The translation problem may be broken up into five parts.

(1) What translations must be performed? This problem will be discussed in terms of the input-output requirements of the ESS translation programs.

(2) In what format are translation data stored in the ESS? The method of storing translation data dictates the translation program, which interprets the translation data derived from the input quantity in such a way as to give the full required output to the telephone operational program.

(3) How are the original translation data specified by a telephone operating company? In this article we will describe some of the forms which have been developed to simplify the problem of specifying translation data. It is important that even the most complex translation items be easily specified.

(4) How are changes in translation data introduced into the system? The ESS program which accepts change data from a teletypewriter, converts these data into a form usable in the system and stores it in the variable memory or call store will be described.

(5) How do we transcribe revised and/or additional translation data into the semipermanent memory or program store from the call store? The process of adding, revising, and deleting translation information in the program store will be described.

II. THE CHARACTERISTICS OF THE TRANSLATION PROGRAM

The translation program (see Fig. 1) is a collection of related subroutines. It is requested by a call control program whenever the latter requires information about a specific item, called the "input parameter" to the translation program. These input parameters are stored in a central control register at the time the translation program is entered. Within one category, such as line translations, there exist a number of input situations, usually specified by entering the translation program at a different entrance point. Whenever the translation program has finished its work, it transfers back to the requesting program. In accomplishing its task, the translation program records its output information

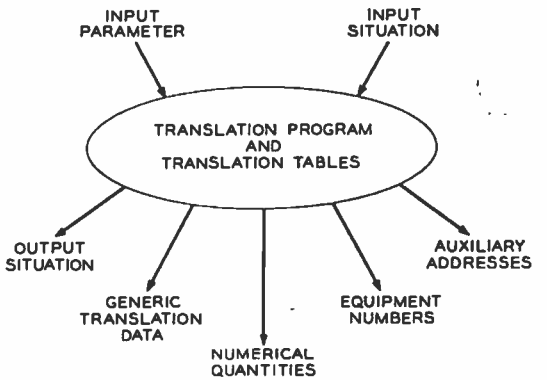


Fig. 1 — Input-output description of translations.

in central control registers and in fixed locations of the call store. If it recognizes an unusual situation, it modifies its return point in the program that requested the translation. The outputs are of five kinds:

(1) Special output situations are the result of finding a special condition as a result of making the translation. For example, if we are making a directory number translation and we find the directory number is unassigned, we wish to indicate that the call should be routed to an intercept operator; we indicate this and other special output situations by modifying the program address to which we return after completing the translation program.

(2) Equipment numbers are the binary addresses required to select a line or trunk network appearance, to operate a relay using a central pulse distributor or signal distributor, or to read a line or trunk scan point.

(3) Numerical quantities such as directory numbers, trunk group numbers, billing indices, and route pattern numbers have the same number of bits and format in all offices; the same quantity may eventually lead to different actions in different offices.

(4) Generic data are nonnumeric output information having the property that a particular binary configuration has the same meaning in all offices. For example, part of the output data from a line equipment translation made in response to a service request is the information concerning whether this is a two-party line and whether it has a TOUCH-TONE subset; to provide this information, a particular group of output bits exists which can be interpreted in the same manner by the call control program in any office.

(5) Auxiliary addresses are program store or call store addresses at which translation information may be found. Sometimes it is more convenient to give as an output the address at which information may be found than it is to give the information. For example, if part of the translation is a list, such as an abbreviated dialing list, it would be impractical to copy the entire list at the time an originating translation is made. Instead, the auxiliary address provides the means for finding the right item in this list at the proper time in the call. The information stored in the block specified by an auxiliary address may consist of generic data, equipment numbers, numerical quantities and auxiliary addresses.

The above is a summary of the characteristics of the translation program in terms of inputs and outputs. The program itself is mainly a series of table look-ups, simple data conversions, and checks for special auxiliary data. Because translations are made so frequently, it is important that the translation program consume minimum time. Because of the volume of translation data, it is important that it be densely and efficiently packed. Because of the large number of required variations in operation and because of the unknown requirements of the future, it is important that the translation scheme be flexible and have ample room for growth of features and services.

The translation program must work with a mixture of data in the call and program stores. The call store contains translation changes representing items of data, such as the class of new customers recently connected to the system, which must be up-to-date but which have not yet been transcribed into the program store. The program store contains the bulk of translation data, including, unfortunately, whatever information has been rendered obsolete by the new information in the call store. The translation program must provide the up-to-date data to the telephone program.

Numerical quantities as defined previously exist in the system; these pose a problem for the program. We must ensure that numerical quantities are never directly examined but are either recorded without examination or are merely stored to act as inputs for a subsequent translation. This makes it possible for all translation outputs to be handled in a standard way in all offices. For example, one part of the line class of service is a numerical quantity representing specialized call routing for this class of service. This quantity may be different for the same general class of service — e.g., PBX toll diversion beyond two message units — in different offices. (The types of classes of service required are too diverse to make the number of a particular class stand-

ard in all offices without incurring a heavy economic penalty in all offices.) However, this presents no problem to the office program because it never examines this part of the class of service; it merely stores it, later uses it as an input parameter for the routing translation, and then receives modified routing data. The translation *program* is insensitive to the value of numerical quantities.

III. INPUT-OUTPUT DESCRIPTION OF TRANSLATIONS

3.1 *Line Translations*

The input parameter for line translations is the line equipment number, which is the network appearance of the line in question. The chief outputs are the directory number (a numerical quantity) and the class of service, a combination of a numerical quantity and generic data.

The generic class data consist of a 6-bit major class word, plus a group of bits representing the presence or absence of some particular service or feature, such as a TOUCH-TONE subset, abbreviated dialing, call transfer, dial add-on, etc. (Space exists for many more bits than have yet been assigned to features or services.) The major class represents mutually exclusive aspects of the class of service, such as denied, unassigned, two-party, manual originating, multiline hunting group, or coin. The code for a major class is standard in all offices.

The numerical part of the class of service is a 10-bit number representing the specialized charging and routing or CHART (CHArge and RouTe) class. As previously mentioned, this number is used as an input parameter for routing translations.

Because two-party lines require more complex translation data (two originating classes, two directory numbers per line), an auxiliary address is part of the translation output of a two-party line translation. This auxiliary address is later used along with the party indication as an input parameter to a special translation program for obtaining the originating class of service of the particular party, as contrasted with the particular line.

In the discussion on multiline hunting groups (MHG)* in the directory

* We use the term MHG to refer to a method of selecting an idle line from a group in the central office, as distinguished from the term PBX, which refers to a type of equipment on the customer's premises. In general, an MHG is connected to a PBX on the customer's premises, but some of the smaller PBX's are treated at the central office as a series completion group. Series completion is handled at the central office by attempting sequentially to connect to a series of directory numbers. The main attribute of an MHG is that one directory number may be associated with many lines.

number translation below, it will be seen that each line in the MHG has a special busy-idle bit in the call store, grouped within a series of such bits associated with the MHG. If an MHG line originates, the translation program automatically busies the special busy-idle bit. (The service request detection program[1] has already marked the regular busy-idle bit busy, but that program has no way of knowing that the requesting line was in an MHG or, if so, which line within which MHG.) Thus, an MHG line can be treated very much like a regular line by the service request processing program.

If a customer has abbreviated dialing and dials an abbreviated code, the translation must provide the directory number corresponding to that code.

If a customer is connected to certain auxiliary equipment, such as an answering service, a special sleeve lead* condition must be created in an auxiliary line circuit to control this equipment. The sleeve lead represents the busy-idle state of the line. When the customer goes off-hook, this sleeve lead must be grounded by operating a relay. Since the conventional No. 1 ESS line circuit does not provide such a relay, nor the means for controlling one, a special relay in the auxiliary line circuit controlled by a general-purpose output of a signal distributor is used for applying the ground to the sleeve lead. The translation must provide the indication that such an auxiliary line circuit must be controlled (this is one of the bits of generic class data) and must provide the equipment number of the signal distributor point used for controlling this circuit.

A 3-bit disconnect guide is retained throughout the call. Two of the bits refer to specific situations, coin and add-on privilege. The latter requires timing to distinguish between a flash and a disconnect, since a flash is a signal to request a dialing receiver so that another telephone may be dialed and added to the present connection. The third bit is general-purpose and indicates that some special action is required at disconnect time, the special action to be indicated by the translation. Included in this category are lines with sleeve lead circuits (idle must be restored by releasing the relay) and multiline hunting group lines (MHG busy-idle bit must be restored).

A number of special output situations also exist in line translations. The output situation problem is handled as follows:

---

* In electromechanical systems, the sleeve lead is a third wire of the connection within the office. The auxiliary equipment can be connected to this point. Every effort was made to make the No. 1 ESS compatible with present customer station equipment.

Suppose there are four special output situations in addition to the normal output situation. The program requesting the translation then places four transfer instructions to the programs corresponding to the four special situations, immediately followed by the program for handling the normal case. Special output situations are discovered in the course of making the translations; the requesting program must be prepared to encounter these situations. The translation program will transfer back to the return address (J)* for the first special situation, to (J) + 1 for the second situation, (J) + 2 for the third situation, (J) + 3 for the fourth situation and (J) + 4 for the normal case. (This avoids the execution time of an extra transfer in the normal case.) This scheme permits several programs to request a particular translation and to have a standard method of handling special output situations that are signaled by the translation program.

The special output situations are as follows:

(1) Unassigned line originates.

(2) Line from master control center originates. Dialing from this line has completely different meaning than dialing from a customer's line.

(3) Line marked in trouble originates. This may well be the signal that the trouble has been cleared and that the line is now also available for terminating calls. The temporary translation routing calls for this number to an operator or announcement will have to be cleared if the subscriber actually dials.

3.2 *Directory Number Translations*

The input parameter for directory number translations is a directory number. A directory number translation is made only when it is already known that the call terminates in the local central office. The directory number is in one of three versions:

(1) Normalized office code plus four binary-coded decimal digits. A *normalized office code* is a 5-bit number representing a particular 3-digit office code of the many that may be handled by one ESS; each of these 3-digit codes corresponds to a different normalized office code. The normalized office code is obtained from the office code translation made when a local subscriber dials a full 7-digit number, or when an incoming trunk sends a full 7-digit number.

(2) Four or five binary coded decimal digits plus the class of an incoming trunk. These must be interpreted to derive the implied normalized office code for this trunk. (Communication of directory numbers

* The J register[3] is used to store the return address. It is set up via the J option at the time of the transfer to the translation program.

among central offices is frequently accomplished by sending only the last four or five digits.)

(3) Standard ESS format, consisting of a 10-bit quantity representing the binary version of the hundreds, tens and units digits, and a 7-bit quantity representing the *number group number*, the identification of the group of 1000 directory numbers to which this directory number belongs. This format might have been derived from an abbreviated dialing translation or from a call transfer translation.

The object of the directory number translation is to find the equipment location of a line associated with the called number and to provide any special information necessary to complete a call to such a line. The directory number translation program checks the busy-idle bit associated with the called line and, if it is idle, marks it busy. If the line associated with the called directory number is busy and has the series completion feature, the translation program will try the directory number to which the called line series completes. If the called directory number has the call transfer service in effect, calls will be transferred to a different directory number; if the latter is not in the local office, the translator output is simply the directory number to which the call is to be transferred.

For any call that can be completed in the local office, the directory number translator returns the line equipment number and the terminating class of either the called subscriber or the line to which his call has been switched because of series completion or transfer. The terminating class information consists entirely of generic data. It consists of a 6-bit major class word having the same meaning and coding as the originating major class word, an indication of the type of ringing signal to be applied, and an additional group of bits representing the presence or absence of particular terminating features and equipment, such as ground start, series completion, and call tracing.

If the directory number is that of a multiline hunting group, the directory number translation hunts for an idle line in the group. The equipment number is then that of the idle line, while the class is that of all the lines in the group. The translation program marks the idle line busy. Subsequent terminating call actions are similar to those for calls terminating to individual lines.

If the called line has an auxiliary line circuit, then the address of the signal distributor point for operating a relay to apply ground to the sleeve lead is provided by the translation.

In addition to the normal case of a line found and available, four special output situations exist:

(1) called line busy

(2) called directory number unassigned; route to intercept

(3) called directory number is associated with a trunk group. This will happen for official numbers that are associated with operator trunk functions.

(4) called directory number has transferred its calls to a number outside this central office. The new directory number is provided as the output of the translation.

3.3 *Office Code, Routing, and Charging Translations*

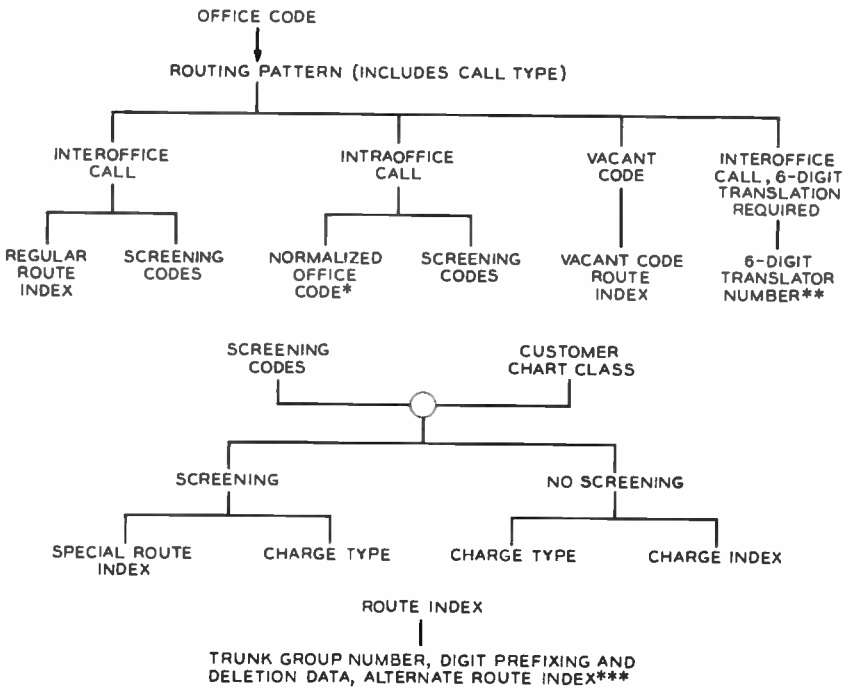
The office code, routing, and charging translations are probably the most complicated of the translations which occur in the ESS. From an input-output point of view, an office code is given and full routing and charging information is supplied. Internally, a number of translations take place before these final results are given.

A special attribute of office code translations is that the translation depends not only on which number was dialed, but on who dialed it. The chart class of the originating customer will have an effect on the routing and charging. For example, a customer whose toll calls are denied will clearly have a different route when he tries to dial a toll call than a regular individual service customer. Similarly, a customer on a four-party line, who, when making a toll call, must be routed via a centralized automatic message accounting office in order to have his directory number requested by an operator, may well be routed differently than an individual or two-party customer making the same toll call. While these unusual routings are fortunately the exceptions, it does mean that fundamentally the translation consists of finding the appropriate data entry in a two-dimensional, rather than one-dimensional, matrix.

The situations described above could, of course, be handled by program means alone. For example, it would be possible to check if the customer has toll denial, and if so, to see whether this is a toll call, or to check whether the customer is on a four-party line, and if he is, check to see if this is a toll call. But such a scheme would lack flexibility. It would not be prepared for unexpected situations. The more generalized approach uses less real time and is able to handle unexpected situations provided the latter do not exceed the limitations of the overall plan.

As mentioned previously, that aspect of a customer's class of service which affects the routing of his calls is the chart class. A chart class is a 10-bit number; thus the number of chart classes is limited to 1024.

The office code, routing and charging translations are shown diagrammatically in Fig. 2. The memory layouts of office code translators are



*USED FOR SUBSEQUENT DIRECTORY NUMBER TRANSLATION

**USED FOR A SUBSEQUENT TRANSLATION; IN THIS TRANSLATION DIGITS 4, 5 AND 6 WILL BE USED AS AN INDEX IN THE SPECIFIED TRANSLATOR TABLE

***USED IF ALL TRUNKS IN THE SPECIFIED GROUP ARE BUSY; THIS ALTERNATE ROUTE INDEX IS THEN TRANSLATED IN THE SAME WAY AS THE ORIGINAL ROUTE INDEX, AND MAY LEAD TO FURTHER ALTERNATE ROUTING

Fig. 2 — Office code, routing and charging translations.

discussed more fully in Section IV, and are shown in Fig. 7. Associated with each office code is a routing pattern. The route pattern first identifies the category of the call, i.e., vacant code, intraoffice, 7-digit interoffice, 10-digit interoffice, 3-digit call, etc. For interoffice codes, this routing pattern gives a regular route index and a series of 15 screening codes corresponding to 15* divisions of the chart classes, or charts. A route index is a number which implies a trunk group plus appropriate digit deletion and digit prefixing information, plus another route index for alternate routing in case the first trunk group is busy. By proper linkage of route indexes, it is possible to create any desired pattern of

* The number 15 is an engineering compromise between a larger number, which would cost extra memory per route pattern, and a smaller number which might excessively limit the screening flexibility.

alternate routing among different trunk groups used for different destinations. A regular route index is used for this call provided no screening takes place. If screening takes place, a special route index will be found and substituted for the regular route index.

Associated with each chart class is a list of 32 or 64 special route indexes or charge indexes.* The route pattern will contain a screening code for each chart. The translation program will then take the screening code associated with the particular chart of which this subscriber's chart class is a member, and if that screening code is n, will read the nth word of the chart class table. This nth word will either provide a substitute or special route index, or it will provide a charge index to be transcribed on the AMA tape. In either case, a charge type is also found. A charge type indicates such items as whether the call is free, whether a detailed AMA entry (including both the calling and called subscriber's directory number) or bulk AMA entry (including only the calling subscriber's number) will be made, whether the entry must include the length of time of the call or whether an entry may be made as soon as the called subscriber answers.

The office code translation must also provide information as to whether overlap outpulsing is used on the route associated with this call. If overlap outpulsing is to be used, it means that pulsing to the distant office must start after the subscriber dials his hundreds digit, i.e., before he has finished dialing. This means that the connection for outpulsing must be set up before the normal time.

The office code translation program must also take into account the question of whether a subscriber dialed a prefix that he was not supposed to dial, or failed to dial a necessary prefix. The standard prefixes of the future in the Bell System are a 1 for station-paid toll calls and a 0 for person-to-person and special calls. However, not all customers will have the same rules for dialing a prefix 1; for example, dial TWX customers may not have to dial a 1 for toll calls, whereas regular customers will have to dial this 1. We must route the call to an appropriate announcement if the 1 is omitted in dialing by a regular customer.

For an intraoffice call, the office code translation must provide the normalized office code of this call. This is a necessary input for subsequent directory number translations (see above). For interoffice calls, if a 6-digit translation is required, the office code translation must provide the number of the table containing this particular 6-digit translation. Subsequently, a translation will be performed using this table, with digits 4, 5 and 6 as the index within the table, since digits 1, 2 and 3 have

* The following description is further clarified in Section IV and Fig. 9.

already been used in the first translation to select the table. (Prefix 1 or 0 is not counted as a digit in the above discussion.)

3.4 *Trunk Translations*

Fig. 3 gives a diagram of the required trunk circuit translations. There is a significant difference in the required translations between universal trunks and miscellaneous trunk and service circuits. With a universal trunk circuit, a trunk scanner number implies a trunk signal distributor number; universal trunk circuits do not have any associated

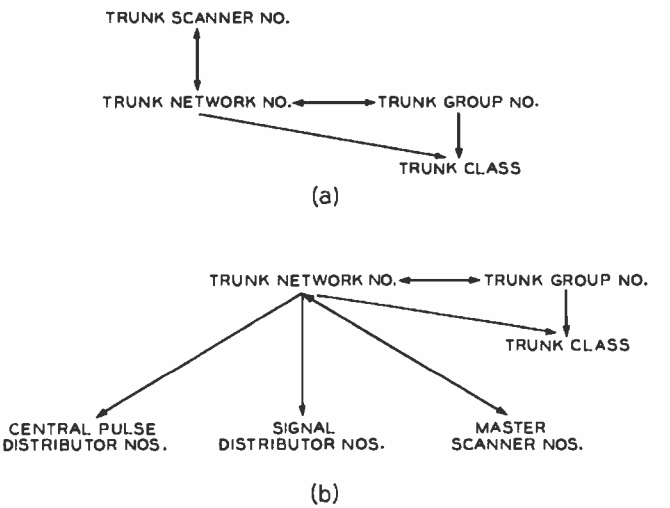


Fig. 3 — Trunk translations.

central pulse distributor points, since one of the requirements of a universal trunk is that it contain only signal distributor controlled relays. Translations must be made from trunk scanner number to trunk network number and vice versa.* The trunk scanner number is the source of information concerning a seizure or disconnect. The trunk network number must be obtained in order to set up a connection to this trunk circuit. A translation from trunk network number to the trunk scanner and signal distributor numbers is required in order to operate the trunk circuit relays when we have seized a particular trunk on the basis of a

* A trunk distributing frame is interposed between the trunk switching frames and the trunk frames, so that any trunk equipment may be connected to any trunk network appearance.

hunt based on network numbers. A translation from trunk network number to trunk group number is required, since the administration of available trunks within a group is made on the basis of the trunk group number. We must be able to find the class of any trunk, and this is done most conveniently by having a translation from the trunk group number to class and from the trunk network number to class.

For miscellaneous trunk circuits the same basic translations are necessary with the following additions: since the master scanner is used for miscellaneous trunk circuits, it is necessary to substitute a master scanner to trunk network number and reverse translation for the original trunk scanner to trunk network number translation. Because the miscellaneous trunks and service circuits are controlled from signal distributor and central pulse distributor points which have no relation to the master scanner number, a translation is required to find the signal distributor and central pulse distributor numbers necessary for controlling a particular trunk circuit.

The trunk class is in a standard 4-word array. The trunk class words are:

(1) common and outgoing information
(2) incoming information
(3) special operator options
(4) trunk circuit program index.

The fourth word contains the trunk circuit program index. This is an indication to the program of the type of trunk circuit; each type of trunk has a different index. The third translation word contains special operator options indicating such items as the type of coin control action to be taken with this trunk. The second word contains incoming information, including the number of digits to be received, the type of incoming pulsing, and whether a start dial signal is expected; for the case of trunks with 4-digit incoming pulsing, a normalized office code is required to steer the call to the proper office code if the particular ESS installation handles more than one office code. The first translation word contains common and outgoing options such as the type of supervision on the trunk, the type of pulsing, the type of trunk circuit (incoming, outgoing or two-way) and details on the form of the outpulsing (for example, start dial signals on dial pulse outpulsing).

Presumably, the trunk class information for any particular trunk takes much less than four words. However, since the number of different trunk classes in any office is relatively small, the gain in having a standard program which will always know what information to expect in a particular bit is greater than the loss of having an unnecessarily

large area of memory devoted to the trunk class detailed information. For a particular trunk group, a code is stored which is expanded into the 4-word array; the 4-word block is stored in memory only once if two trunk groups have the same class.

Other translations associated with trunks include the hunt for an idle trunk. In the case of the directory number translation, it is convenient for the translation program to make the busy check because the translation program has all necessary equipment numbers already generated. It is equally convenient for translations to make the hunt for an idle trunk and to make such a trunk busy, to hunt for an outgoing transmitter associated with a particular trunk, to hunt for an idle terminal in a conference trunk, and to restore trunks in memory to the idle state after a disconnect or after the discovery of a blockage in the network.

Table I summarizes the universal trunk translations required for an incoming call. The call is detected by means of a seizure signal recognized by a trunk scanner. The trunk scanner address must be translated to find the trunk network number, so that a receiver may be connected to the incoming trunk. In addition, the class of the trunk must be derived, so that the proper receiver may be connected. The trunk class is derived by translation from the trunk network number. The trunk class also indicates how many digits are expected over this trunk. Finally, the trunk class is also needed to give the call processing program which controls the trunk circuit relays the necessary information as to the type of trunk involved. This information is given in the form of a circuit program index, an identifying number unique to each particular type of trunk circuit configuration. Finally, the trunk network number is used to permit the call processing programs to mark the appropriate path memory for all paths to be associated with this trunk during this call.

Incoming calls are not normally screened, because screening is per-

TABLE I — INCOMING CALL TRUNK TRANSLATIONS

Quantity	Function
Trunk scanner no.	seizure or disconnect signal discovered at this scan point
↓	
Trunk network no.	used for network path hunt and network memory changes
↓	
Trunk class code	compact version of trunk class, used to find detailed trunk class information
↓	
Trunk class	used to determine the type of incoming receiver needed, incoming pulsing details, type of trunk circuit

formed at the originating office. The translation corresponding to an office code translation is merely one to find out which intraoffice code has been called so that the proper directory number translation may be made.

Disconnect is recognized by the scanner at the trunk circuit. It is again necessary to make a translation from the trunk scanner number to the trunk network number so that the disconnect signal may be associated with the proper path information; the trunk class must be found so that the trunk release operations may be controlled.

Table II summarizes the universal trunk translations necessary for an outgoing call. The trunk was initially seized because it was a member of the trunk group indicated by the route index which was found by the office code translation. For outgoing calls, the route index indicates the number of digits to be pulsed plus any digit prefixing information. The class of the trunk is necessary in this case primarily to select the type of transmitter to be used in connection with this outgoing trunk and to provide the circuit program index necessary to control this trunk.

In order to control a trunk circuit and in order to mark the trunk busy it is necessary to know the scanner number associated with the trunk; for this purpose, a trunk network number to trunk scanner number translation is necessary.

TABLE II — OUTGOING CALL TRUNK TRANSLATIONS

Quantity	Function
Office code	Dialed by customer
↓	
Route index	
↓	
Trunk group	
↓	
Trunk class	Used for selecting outpulsing transmitter, and indicating type of trunk circuit
Network number of idle trunk	Used for network path hunt and network memory changes
↓	
Trunk scanner number	Used for making scan point memory busy
⋮	
At disconnect time:	
Trunk scanner number	Disconnect detected at this scan point
↓	
Trunk network number	Used for making network memory changes at disconnect time
↓	
Trunk group number	Used for updating list of idle trunks in group
↓	
Trunk class code	Compact version of trunk class, used to find detailed trunk class information
↓	
Trunk class	Used to determine the type of trunk circuit

A disconnect may be initiated by either subscriber at the trunk circuit. A translation from trunk scanner number to trunk network number is necessary at this time to find the proper network path information, plus class.

3.5 *Miscellaneous Translations*

A number of other translations exist, not associated with the basic telephone operation. Some of these are for maintenance purposes. Alarm indications are connected to the master scanner; when such a scan point becomes active, a translation must be made to discover the meaning of this particular alarm indication. If a particular unit is being diagnosed for faulty operation, a list of the scan points for examining the maintenance outputs of the unit and a list of signal distributor and central pulse distributor points for applying test signals to that unit must be provided.

Translations must also be provided for the details of traffic counts peculiar to a specific office, including indications of which traffic counters are to be printed out on a teletypewriter.

In general, translations must provide information on any item that a telephone company may change on a day-to-day or long-term basis.

IV. MEMORY LAYOUT

A major aspect of the translation problem is that of storing the large volume of the translation data in memory. The design of the memory layout was influenced by three requirements:

(a) The data can be stored in either program store (PS) or call store (CS).

(b) The data must be densely packed in order to conserve memory space.

(c) The output of a translation must consist of generic data, equipment numbers, and numerical quantities.

Because of requirement (a), a 23-bit word was chosen as the basic translation word (TW). Its physical location is either a 23-bit CS word, the right part (23 bits) of a 37-bit* PS word, or the left parts (14 bits each) of two consecutive PS words. In the latter case, the first of the two words contains the 14 least significant, the second one the 9 most significant bits of the translation word.

An exception from the rule of the 23-bit translation word is found in abbreviated dial and transfer lists. It was found that 23 bits were inadequate, but 28 bits were adequate, for most entries in such a list. A

* Of the 44 bits in each program store word, 7 are check bits; hence only 37 useful bits of data are available in each word.

list word therefore consists of 14 bits; two such words form a list entry. Such lists are preferably stored in the upper or left bits of program store words. In the call store, such list words are stored in two 23-bit words.

Translation data exist in the PS as a collection of tables. The set of tables devoted to each type of input parameter is called a "translator." Corresponding to the various input parameter types, there are line equipment number translators, directory number translators, 3-digit code translators, trunk network number translators, etc.

Translators are composed of subtranslators, each subtranslator corresponding to a growth unit of the central office. For instance, there is a subtranslator per line switch frame, per trunk switch frame, per number group (i.e., block of 1000 directory numbers), etc.

Subtranslators are joined together to form a translator in the following way:

The binary representation of the input parameter is divided into two parts, the *subtranslator selector* identifying the unit and the *index* identifying the item within the unit. A head table contains the addresses of all subtranslators. The head table exists for the ultimate size expected for the particular central office, but only as many subtranslators are provided as have corresponding units, and new ones are added as the number of units increases (see Fig. 4).

A subtranslator is a table which consists of one translation word (TW) per index. This word, the primary translation word, contains either the complete data connected with the input parameter, or if one word is not sufficient, a reference address to an auxiliary block where the data are stored in auxiliary translation words. To make the recognition of auxiliary addresses possible, the complete data cannot start with three leading zeros; a word whose three leading bits are zeros is then interpreted as an auxiliary address.

The requirement for conserving memory space suggests the technique of using abbreviated codes for generic data. The call control programs which use translation need directly usable information. Therefore, generic data consist of separate groups of bits for each item described. The number of bits must be large enough to allow for all possible values of the data, whether they occur in a particular office or not. For instance, the type of outpulsing used on outgoing trunks is described in 3 bits to allow for all possible types of transmitters, although there might be only one type, multifrequency transmitters, in actual use. By listing the values of the generic data which occur frequently in a particular office and assigning consecutive numbers to them, one arrives at an abbreviated code for the actually occurring data combinations in a particular office. The detailed version of this data is therefore stored

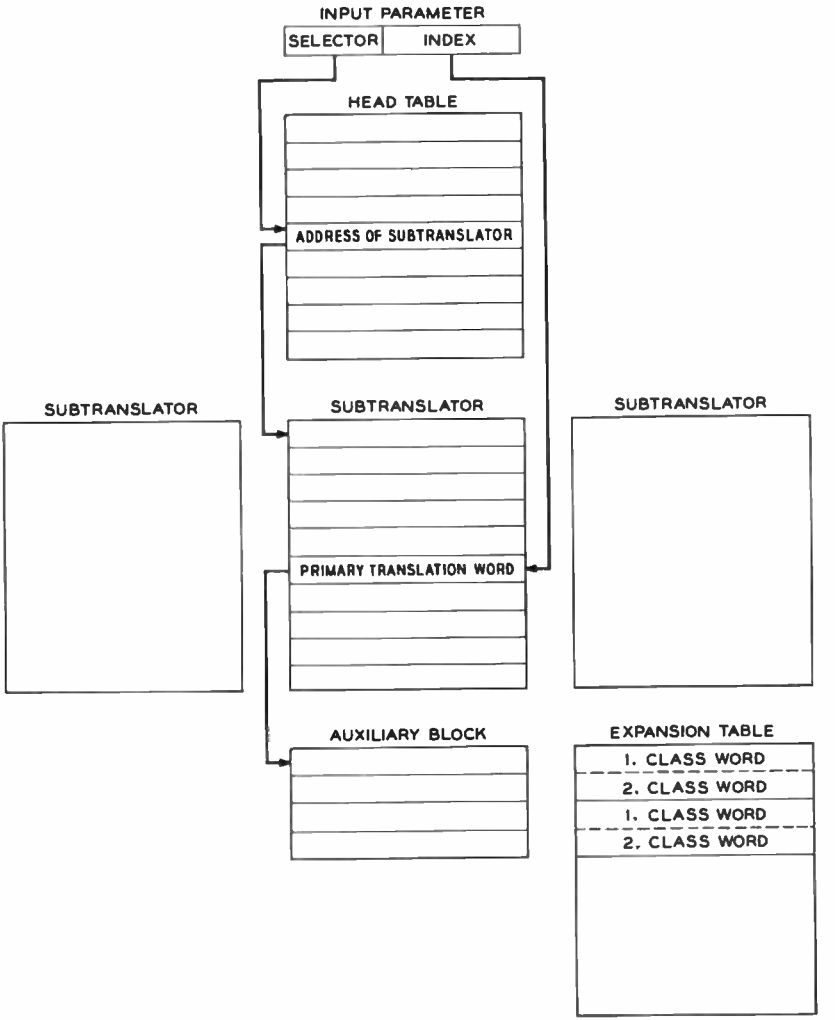


Fig. 4 — Pictorial description of a translator.

once in an expansion table; everywhere else the abbreviated code is stored. Since the call control programs are interested in generic data, the translation program expands the abbreviated code (which is not generic) before delivering it to the call control program.

Examples for the use of abbreviated codes are the line class of service, the trunk class, and the route index:

(a) Class of service includes equipment information, special service

information, and routing and charging information. In the detailed form, it occupies two translation words. For a reason which will soon become apparent, the size of the abbreviated codes is limited to 6 bits, which makes it possible to use them for the 56* classes of service most frequently occurring in a particular office. Consequently, for many lines the primary translation word is sufficient to hold the originating information, i.e., the directory number (17 bits) and the abbreviated code (6 bits). Lines for whose originating class of service no abbreviated code exists need at least four words of storage, one for the auxiliary address, one for the directory number, and two for the class of service.

(b) The trunk class describes equipment and use of a trunk or service circuit. When spelled out in detail in the trunk class expansion table, the data occupy four translation words. However, the abbreviated trunk class code appearing in the trunk network number translator and the trunk group number translator is only 8 bits long, allowing for 256 different classes of trunks in a particular office.

(c) A route index is an 11-bit abbreviated code for a particular routing describing the first-choice trunk group, the alternate route and special treatment. The detailed route information, taking up two translation words each, is stored in the route index expansion table.

To sununarize: a translator consists of a head table, one or more subtranslators, auxiliary blocks, and possibly expansion tables (see Fig. 4).

A detailed description of the line equipment number translator is given here as an example (see Fig. 5). It has four types of primary TW's:

(a) The TW contains an auxiliary address (complex class of service).

(b) The TW contains an abbreviated class code and a directory number (simple class of service).

(c) The TW contains the abbreviated code for MHG lines, the MHG number and the terminal number (i.e., the position in the multiline hunting list).

(d) The TW is zero (unassigned line).

The class expansion table contains two class words for each abbreviated code. They provide for the following categories:

(a) major class: a code for the mutually exclusive aspects of the class of service, as individual line, coin line, multiline hunting group, etc.

(b) feature class: a group of bits representing the presence or absence of some equipment and service features as TOUCH-TONE dialing, ground start, abbreviated dialing, variable or preset transfer, etc.

* 56 instead of 64 since codes 0 to 7 are not usable. Their binary representation, starting with 000, conflicts with the code for an auxiliary address.

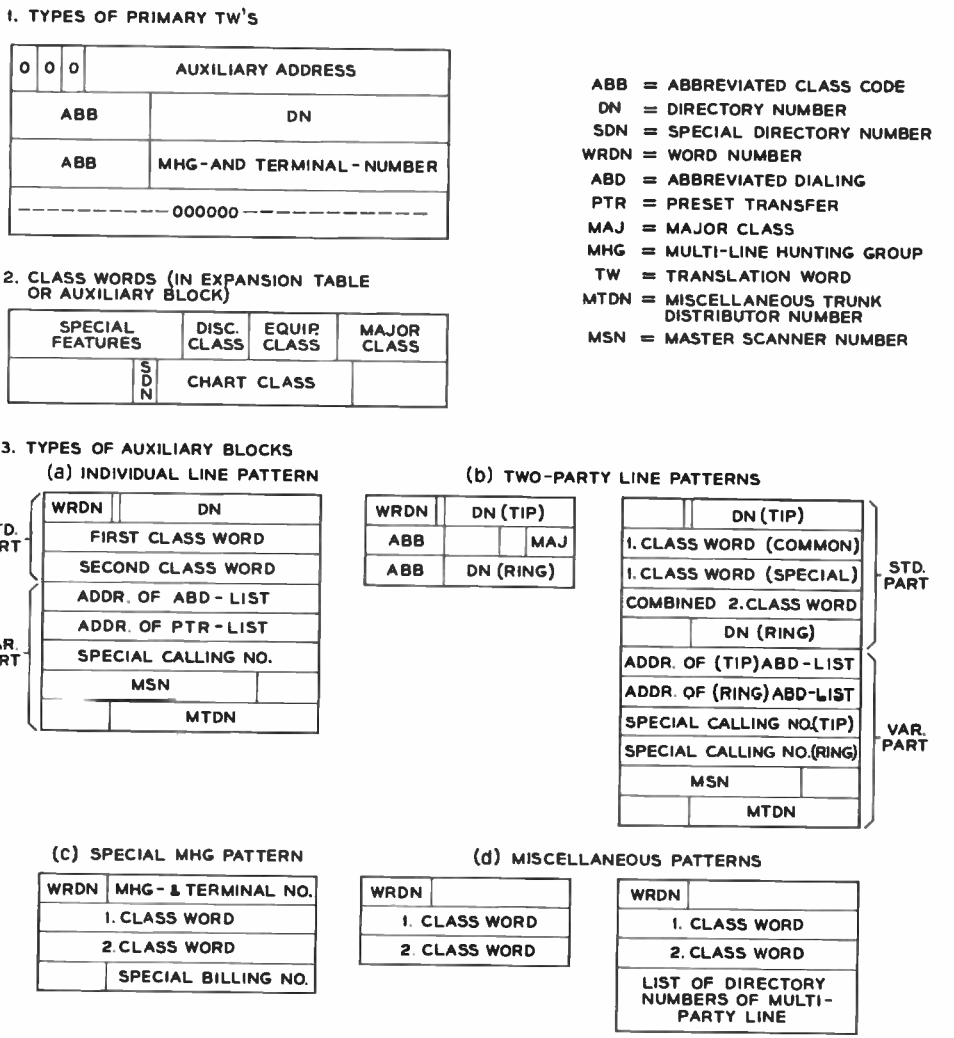


Fig. 5 — Data stored in line equipment number translator.

(c) chart class: a code representing the charging and routing directions

(d) disconnect class: a group of bits indicating the action to be taken at disconnect time.

The auxiliary blocks appear in several patterns:

(a) Individual line pattern: this pattern contains as the standard part

the directory number and the two class words (which provide the same categories as those in the expansion table). The following variable part may be entirely or partially missing depending on the class of service. It contains successively the address of an abbreviated dial list, the address of a transfer list, a calling number which differs from the billing number, and sleeve lead auxiliary line circuit data.

(b) Two-party line pattern: this comes in an abbreviated and in an expanded form. In the abbreviated form, the pattern contains abbreviated class codes and directory numbers for both parties and, for reasons of expediency, the common major class and equipment features. In the expanded form, it contains the detailed data for both parties. The common equipment features and disconnect class are stored only once. The variable part of the pattern contains, if they exist, the addresses of the abbreviated dial lists both for tip and for ring party, special calling numbers for both parties, and the common sleeve lead data.

(c) MHG pattern: If a line from a multiline hunting group is not billed to the main directory number, or if its ground start feature differs from that of the majority of lines in the group, an auxiliary block is required which contains the MHG number, terminal number, two class words, and billing number (either special or common).

(d) Miscellaneous patterns exist for multiparty lines and the line from the master control center.

(e) Multiline hunting groups have their own miscellaneous translator. Its head table contains for each group, identified by its PBX number, the address of a common block. The common block contains the main directory number, two originating and one terminating class words, and the hunting list as a standard part. A variable part may contain the address of an abbreviated dial list, and/or sleeve lead data for an electromechanical overflow counter.

The office code translations require a number of special techniques in order to include the screening facility previously described. The translation memory layouts are shown in Fig. 6. The first step in making an office code translation is to find a route pattern number associated with each office code. This is done by looking in a table of a thousand office codes to find a route pattern number. Assume that the route pattern for a given office code is 41. Via the route pattern address table, this route pattern number is expanded into the address of the corresponding route pattern information. The route pattern information is a 5-word block. The first word contains a route index (the standard route index) and the call type information. The next 4 words contain 15 screening codes corresponding to the 15 charts or divisions of chart classes.

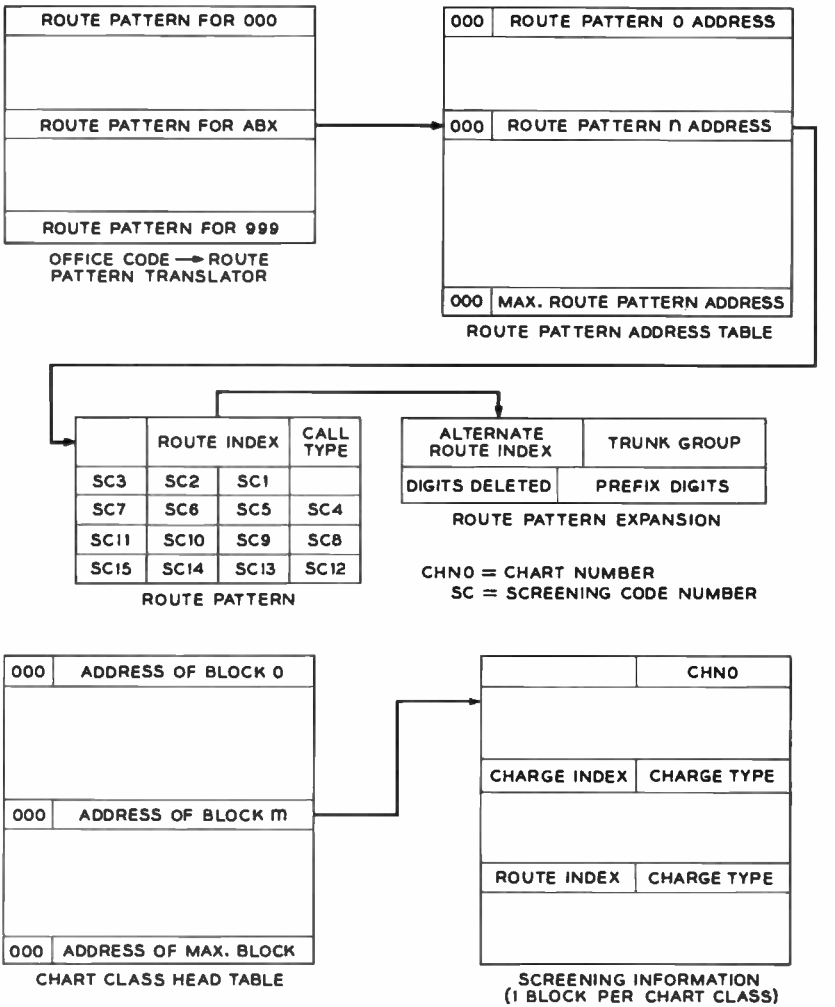


Fig. 6 — Memory layouts for office code translations.

Next, the originating customer's chart class must be used to find screening and charging information. If this chart class is M, the mth word of the chart class head table indicates the address of the block of screening information. A preliminary word in this block gives the chart number (1–15) corresponding to the block. If the chart number is, for example, 11, then SC11 in the route pattern would indicate the word in the screening information block which is applicable to this call.

If SC11 = 23, word number 23 of the screening information block is the desired screening word. The contents of this screening word are the charge type and either the billing index (if no screening is invoked) or a special route index (if screening is invoked).

The route index is separately expanded into 2 words of data giving the first-choice trunk group, an alternate route index in case this trunk group is busy, and an indication of how many digits are to be deleted and/or which digits are to be prefixed for outpulsing over the trunk group indicated by this route index.

Other cases, such as vacant codes, intraoffice codes, and codes requiring six-digit translations, are handled by variations of this basic technique. For example, if six-digit translation is required, the expansion on the route pattern indicates that an auxiliary three-digit translation is required and that this translation is stored at a table starting with a given address.

Translation data which have recently been changed are stored in the call store until they have been transcribed into the program store. While in the call store, such entries are referred to as recent changes (RC's).

RC's are stored in the same form as the data will later have in the program store, i.e., as primary and auxiliary TW's. However, since they do not appear in the context of a subtranslator, their association with a particular input parameter must be established. This is done by storing with them their primary translation address or TAG, i.e., the address of the subtranslator location associated with the primary TW.

An RC entry therefore consists of a primary part and possibly an auxiliary block. The primary part occupies two call store words, an RC register. The first word contains the TAG and the status bits, and the second word contains the primary TW.

The four states of the status bits correspond to the four possible states of an RC:

11 — temporary, i.e., do not incorporate RC into PS;

10 — permanent, i.e., incorporate RC into PS;

01 — delayed, i.e., RC is not yet active; and

00 — deleted, i.e., inactive or no RC.

A temporary recent change is used in connection with certain services which require a change of translation information that is not meant to be permanent. For example, the record that a subscriber wants calls to his telephone number to be transferred temporarily to another telephone number is not meant to be entered in the permanent translation record of the system. The record should be used in making the terminating translation whenever the subscriber is called; it must temporarily

override the permanent translation information. However, the permanent translation information must not be lost, because it contains all aspects of his normal service which are reinstated when the temporary change of translation is deleted. Temporary translation changes are used in connection with temporary transfers, or with an indication that a line is temporarily out of service because of trouble.

The primary RC's are stored in the primary RC area in ascending order of their TAG's. RC's with the same TAG but with different status (and data) may exist simultaneously. If they do, they are arranged in the order: temporary, permanent, delayed, and deleted.

If an RC requires auxiliary data, it is stored temporarily in the auxiliary RC area and the primary TW is referenced to this temporary auxiliary block.

RC's must be available for translation. Therefore, before translation data are read from the program store, a search is made through the RC area for possible superseding information. For some translations, such as trunk and 3-digit code translations, the search may be bypassed if an RC indicator, a call store bit dedicated to this function, indicates that no RC exists at this time for the translator.

The search through the primary RC area is performed as a so-called "binary hunt."

Let us assume that the RC area contains exactly $2^n - 1$ entries, as shown in Fig. 7. Therefore, there is a central register which divides the area into a lower and an upper half. Since the RC's in the area are ordered according to their TAG's, comparing the primary translation address of the item for which the translation is about to be made with the TAG in the central register renders a three-way decision: an RC for the item is found in the central register, an RC may exist in the lower half, or an RC may exist in the upper half. If no RC is found in the central register, the search is continued in either the lower or the upper half, which is again an interval of the size $2^m - 1$. The process is continued either until an RC is found in a central register or until the interval is reduced to size $2^1 - 1 = 1$. If no RC is found within n steps, then it has been proven that no RC for the item to be translated exists.

The restriction that the size of the RC area be exactly $2^n - 1$ registers is unnecessary. If the size is m, the area is considered to consist of two overlapping sections of size $2^n - 1$. After an initial decision as to which section applies, the hunt can proceed exactly as described above. Again, after $n + 1$ steps, it is proven that no RC for the item to be translated exists.

Considering the fact that in most cases no RC will be found, a modifi-

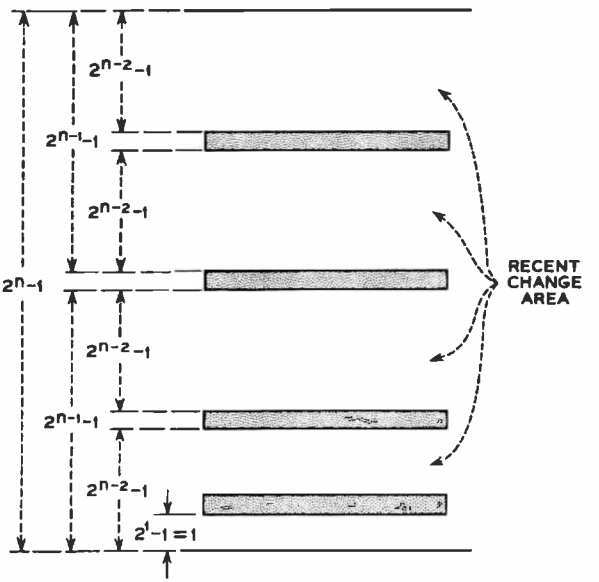


Fig. 7 — Recent change hunt.

cation to accelerate the hunt was designed by substituting a time saving two-way decision for the three-way decision: if an RC exists at all, it may be in the central register or the lower half of the interval, or it may be in the upper half. If an RC exists in some central register, it is considered to be in the lower half; the hunt is then continued, and from then on the result of the decision will always be the upper division, with the final result that the hunt ends in the register just below the one with the found RC. By examining as a final step the register just above the "end" register, either the RC is found or its nonexistence determined.

Auxiliary data are always obtained at the address stored in the primary TW. The translation program does not care whether it is a temporary or a permanent auxiliary address.

V. INITIAL PREPARATION OF TRANSLATION DATA

Fig. 8 shows how translation data enter the system. Originally, the telephone company fills out forms from which punched cards are derived. These punched cards are processed by a general-purpose computer program and are then placed on twistor cards which are inserted into the ESS program store. Subsequent information is inserted by means of

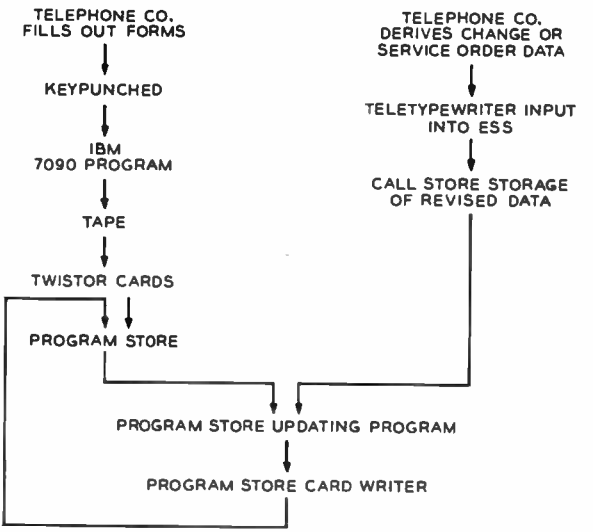


Fig. 8 — Method of placing translation data in program store.

a teletypewriter message typed into the system; the system then stores the translation information corresponding to this information in the RC area of the call store. During the actual running of the system, the call store is always checked for any updating information of the program store translation information, so that the translation program delivers the most current translation information. Periodically, the old records from the program store and the recent changes from the call store are processed, and a new set of program store cards are created using the program store card writer. The change messages are then deleted from the call store, and the associated memory is again made available for new change messages.

A set of forms for deriving the original translation data was created for use by operating telephone companies. In creating these forms, an attempt was made to simplify the task of filling in the high-runner information and to make the complicated data conversions a part of the general-purpose computer program for taking the contents of these forms and creating the information to go on the program store.

Fig. 9 shows the major form used for line information. The form is organized by directory numbers and is headed by an office code and hundreds indication. These forms were designed to simplify keypunching. The small numbers shown on these forms indicate the column of an IBM card into which the appropriate information is to be entered. The

ESS 1100
11-63

DIRECTORY NUMBER RECORD CONVERSION
NO. 1 ESS
ESS UNIT _____
OFFICE CODE _____
HUNDRED

DIRECTORY NUMBER	LINE EQUIPMENT NUMBER						RING COMB		CLASS INFORMATION						SUP INFO REF		REMARKS	OLD DIRECTORY NO.	OLD CABLE PAIR	NEW CABLE	
									CLASS												
	NET	FRAME	BAY	CONC	SWITCH	LEVEL	MULT	CODE	IFB				USOC	EQUIPMENT	FEATURES	PAGE	LINE				CABLE	PAIR
49	00	0	0	0	0	00	-	-	✓					—	—	—	—					
48	15	7	2	7	3	15		1					2FR	0111	0372	—	—	HIGHEST LEVEL IN 4:1 CONC OFFICE				
47	15	7	1	7	7	03		2					2FR	0111	0372	—	—	" " " 2:1 "		"		
46	HG			N	ø	00	1	7										ENTRY FOR AN MHG				
45																						
44																						

Fig. 9 — Form used for line information.

office code and hundreds information, being common to an entire page of a form, is placed in the left columns of the card so that it can be simply duplicated in going from card to card. It is followed in the sequence of columns by the last two digits of the directory number and the line equipment number. This line equipment number is written in the standard format of line equipment numbers in the ESS, i.e., a 2-digit network number, a 1-digit frame number, a 1-digit bay number within the frame, a 1-digit concentrator number, a 1-digit switch number and a 2-digit level number. All the digits of the line equipment number must be filled in unless the directory number is associated with more than one line equipment number. This is the case for a multiline hunting group. If a particular directory number is associated with such a group, the person filling in the directory number record fills in the letters MHG in the network and frame columns, and the number of the hunting group in subsequent columns.

Next comes a pair of columns for ringing combinations. For all individual lines nothing is filled in. For two-party lines, the number 1 or 2 is filled in, and for multiparty lines an M is written in the first column and the code is written in the second column.

Next comes a group of columns for the class information of the customer. This class is summarized in a uniform service order code (USOC) and two 4-digit numbers summarizing the special equipment and features associated with this subscriber. The five most common USOC codes in a particular office will be written in as headings for the first five columns, and a check mark will then indicate if a subscriber's class of service is one of these. If not, the 3-digit USOC designation is written in the last column of the class heading. The equipment and features columns consist of two 4-digit octal numbers which specify special equipment and

special features associated with this line. Each of the octal numbers summarizes a maximum of three binary conditions. It should therefore be quite easy to remember the significance of each of these octal characters, especially those pertaining to features that are very common. For a subscriber who has no special equipment or features, it is sufficient to have the equipment and features columns blank.

If additional digital information is required, this information is found in a supplementary reference form, and the basic directory number record merely points to the page and line on this form at which this supplementary information is to be found. Supplementary information includes such items as sleeve lead circuit scanner and signal distributor addresses, abbreviated dialing lists, fixed transfer lists and series completion lists.

This directory number record is straightforward, especially for lines with simple features and equipment information. For the lines which have more complicated information, a somewhat higher degree of knowledge of the system is required. For example, it is necessary to know that when a line has a sleeve lead as part of its auxiliary line equipment, it is necessary to fill in supplementary information; this supplementary information should be the master scanner and signal distributor number of the sleeve lead circuit.

For multiline hunting groups, the form shown in Fig. 10 is used. Each terminal in a multiline hunting group is identified by the group number and the group terminal number. It is, of course, necessary to specify the line equipment of each terminal. The main directory number of the MHG must be shown, and if a particular terminal is to be reached on a nonhunting basis using another directory number, this must also be specified. The make-busy arrangements must be specified, and the

ESS 1105
11-63

MULTI-LINE HUNTING GROUP RECORD
NO. 1 ESS

ESS UNIT _____ PAGE ____ OF ____

LINE	GROUP NUMBER	GROUP TERMINAL NUMBER	LINE EQP NO HUNTING ORDER						DIRECTORY NUMBER	NON-HUNTING DIRECTORY NUMBER	MAKE BUSY		CLASS INFO			CHART CLASS		SUP INFO REF	
			NET	FRAME	BAY	CONC	SWITCH	LEVEL			TYPE	KEY NO	USOC	EQP	FEA	CHART	COLUMN	PAGE	LINE
00																			
01																			
02																			
03																			
04																			
05																			

Fig. 10 — Form used for entering multiline hunting group information.

USOC code, equipment, and features may have to be specified separately for different terminals if not every line in the hunting group has identical treatment. In the case of MHG's, a single USOC code may specify a number of different toll diversion treatments, so that it is necessary to indicate the specific chart class. (Normally, a chart class is implied by the USOC code.) Supplementary information may be required for at least some of the terminals in the hunting group if, for example, they have sleeve leads.

The trunk forms are relatively straightforward and will not be described in detail here. They present the information that is necessary for making the translations indicated in Section III.

The specification of the 3-digit translations is considerably more complicated. Five different forms must be filled out. Furthermore, there is considerable interrelationship among these five forms.

The first and simplest of these forms is the basic 3-digit translation form in which, for every 3-digit code, a rate and route pattern, simply a 4-digit number, is specified (see Fig. 11). All office codes having the same rate and route pattern must have the property that all classes of service in the particular office are routed and charged in an identical manner.

A similar form is filled out for each 6-digit translator.

Next, a rate and route pattern record must be filled out (see Fig. 12). This pattern gives a regular route index, a call type, an indication of whether overlap outpulsing can be used with the regular route index, and a series of 15 screening codes. If this is an intraoffice call route pattern, the normalized office code is substituted for the regular route index. Each intraoffice office code must have a separate route pattern.

The screening codes that are written in the 15 double columns must have corresponding entries on the form shown in Fig. 13. This form has

ESS 1300-8
11-63

THREE DIGIT TRANSLATIONS
NO. 1 ESS

ESS UNIT _____
BASE RATE AREA _____

PAGE ____ OF ____.

1ST DIGIT						1ST DIGIT					
2ND & 3RD DIGITS	RATE & ROUTE PAT.	REMARKS	2ND & 3RD DIGITS	RATE & ROUTE PAT.	REMARKS	2ND & 3RD DIGITS	RATE & ROUTE PAT	REMARKS	2ND & 3RD DIGITS	RATE & ROUTE PAT.	REMARKS
00			50			00			50		
01	0009		51			01	0006		51		
02	0002		52	0008		02	0007		52		
03	0003		53			03	0000		53		
04	0004		54			04	0001		54		
05	0005		55			05			55		

Fig. 11 — Three-digit translation form.

ESS 1305
11-63

RATE & ROUTE PATTERN RECORD
NO. 1 ESS

ESS UNIT _____

RATE & ROUTE PATTERN 1ST & 2ND DIGITS

R&R PAT. 3RD & 4TH DIG	RATE & ROUTE CHART SCREENING CODE 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	REG. CALL TYPE	REGULAR ROUTE INDEX	FAT NO	OVERLAP OP	REMARKS
00			01													10	0100			
01			01													10	0101			
02			02													10	0102			
03			03													10	0103			
04			04													10	0104			
05			05													10	0105			
06			06													10	0106			
07			06													10	0107			
08			00													07	0108		Y	
09																		01		

Fig. 12 — Rate and route pattern record.

two columns devoted to each chart class. Associated with each screening code is a charge index and sometimes a special route index.

To illustrate the problem of filling out these forms, let us consider only that aspect of the forms dealing with the translation necessary for wide area telephone service (WATS). We will consider here only interstate WATS. This service permits all subscribers to dial calls outside their state within certain zones on bands. Six different bands are provided, and a customer is able to reach all bands up to the farthest band that he is allowed to reach. For example, a customer in New York subscribing to band 6 will be able to call anyone in the continental U. S. outside New York State, whereas a customer having only band 1 service can call only a few of the surrounding states. The office code form has been filled out for a few typical office codes. These office codes include 9 numbering plan area (NPA) codes (codes indicating a 10-digit call to an area outside the subscriber's 7-digit area) and one conventional 7-digit code. Different route patterns have been assigned to each of these codes, and the route patterns are expanded on the rate and route pattern record. Route patterns 0 through 7 are associated with call type 10 (a 10-digit call). Whereas route pattern 8 is associated with a 7-digit code, route pattern 9 is associated with a 6-digit foreign area translator. Since routing will be done on the basis of the 6 digits dialed, no route index is specified. Nine different regular route indexes are shown for the nine route patterns. In addition, a series of screening codes for chart 3 has been assigned. Chart 3 is the chart assumed to be used in the local office for the WATS classes of service.

In Fig. 13, columns are labeled by the appropriate class of service.

ESS 1304-A
11-63

RATE AND ROUTE CHART 03
TITLE
NO. 1 ESS
ESS UNIT

PAGE 1 OF —

SCREENING CODE	SUBSCRIBERS CLASS OF SERVICE																	
	COLUMN 0001						COLUMN 0002						COLUMN 0003				COLUMN 0004					
	WATS 1M						WATS 2M						WATS 3M				WATS 4M					
	SPEC ROUTE INDEX	CHG	ACC I	ACC O	SP CALL TYPE	REM	SPEC ROUTE INDEX	CHG	ACC I	ACC O	SP CALL TYPE	REM	SPEC ROUTE INDEX	CHG	ACC I	ACC O	SP CALL TYPE	REM	SPEC ROUTE INDEX	CHG	ACC I	ACC O	SP CALL TYPE	REM
00	0081	000					0081	000					0081	000					0081	000				
01		015						015						015						015				
02	0081	000						015						015						015				
03	0081	000					0081	000						015						015				
04	0081	000					0081	000					0081	000						015				
05	0081	000					0081	000					0081	000					0081	000				
06	0081	000					0081	000					0081	000					0081	000				
07																								

(a)

ESS 1304-A
11-63

RATE AND ROUTE CHART 03
TITLE
NO. 1 ESS
ESS UNIT

PAGE 2 OF —

SCREENING CODE	SUBSCRIBERS CLASS OF SERVICE																	
	COLUMN 0005						COLUMN 0006						COLUMN 0007				COLUMN 0008					
	WATS 5M						WATS 6M						WATS IF									
	SPEC ROUTE INDEX	CHG	ACC I	ACC O	SP CALL TYPE	REM	SPEC ROUTE INDEX	CHG	ACC I	ACC O	SP CALL TYPE	REM	SPEC ROUTE INDEX	CHG	ACC I	ACC O	SP CALL TYPE	REM	SPEC ROUTE INDEX	CHG	ACC I	ACC O	SP CALL TYPE	REM
00	0081	000					0081	000					0081	000										
01		015						015						015										
02		015						015					0081	000										
03		015						015					0081	000										
04		015						015					0081	000										
05		015						015					0081	000										
06	0081	000						015					0081	000										
07																								

(b)

Fig. 13 — Rate and route chart.

Two types of WATS classes of service exist, measured-time and full-time. Customers who have WATS measured-time service are charged a base rate for a certain number of hours of calling time, and are charged for overtime on a time basis. Customers who have WATS full-time service are allowed to call for an indefinite amount of time for the basic rate. Six of the seven screening codes refer to the six bands of WATS: the seventh refers to intrastate numbers. All interstate WATS customers

are denied intrastate calls and are routed to a special announcement indicated by special route index 0081. Customers who are allowed to complete their calls because the calls are within their allotted bands are routed via the regular route. The charge index for measured-time WATS customers is assumed to be 15 and for full-time WATS customers is assumed to be 16. (The charge index numbers are for this particular office.) If, on the other hand, a customer is denied the right to make this call because the call is outside his band, there is no charge, a condition indicated by charge index 0. In examining the rate and route chart it can be seen that customers with WATS 1 service are allowed to call any codes within screening code 1, customers with WATS band 2 service within screening codes 1 and 2, . . . , and WATS band 6 service customers are allowed to dial any office codes with screening codes 1–6. For example, any office code having a route pattern which has a screening code 4 on chart 3 would be denied to the customer with WATS 1M, 2M and 3M, and would be permitted to customers with class of service WATS 4M, 5M and 6M.

The chief difference between the WATS 1F and WATS 1M customer is a different charge index for those calls that are not denied. This charge index then implies what type of charge record will be made of this call. (Presumably, in this case, the operating company is interested in taking data even though no charge will be made on each individual call.)

The translation forms have a dual use. They are used for creating the initial translation information and they become part of the office records. When subsequent changes are made they are made on these forms. A number of other forms, similar to the forms described, have also been created for maintaining office records. However, the above records contain most of the information necessary for generating the original line, trunk, and office code translation information. These forms are then punched on IBM cards and go through an extensive sorting, checking and data conversion program. This program has been written for an IBM 7090 computer and is approximately 15,000 words long. To compile the data for a 5000-line office should require approximately one hour of running time.

VI. Process of Accepting Recent Changes

The bulk of all RC's are introduced into the No. 1 ESS through service orders sent over the service order teletypewriter. A service order is the form by which the operating company informs everyone concerned of the pertinent facts of pending changes on customers'

lines and telephone numbers. The message is expressed in telephone company language and may contain, from the viewpoint of ESS, both significant and superfluous information. (For example, the fact that a pink phone should be installed is not pertinent translation data in ESS.)

Fig. 14 shows a copy of a service order with its various fields. The first three fields on the first line are intended to hold the service order number, the activation code, and the service order type. The activation code indicates whether an RC should become effective immediately or

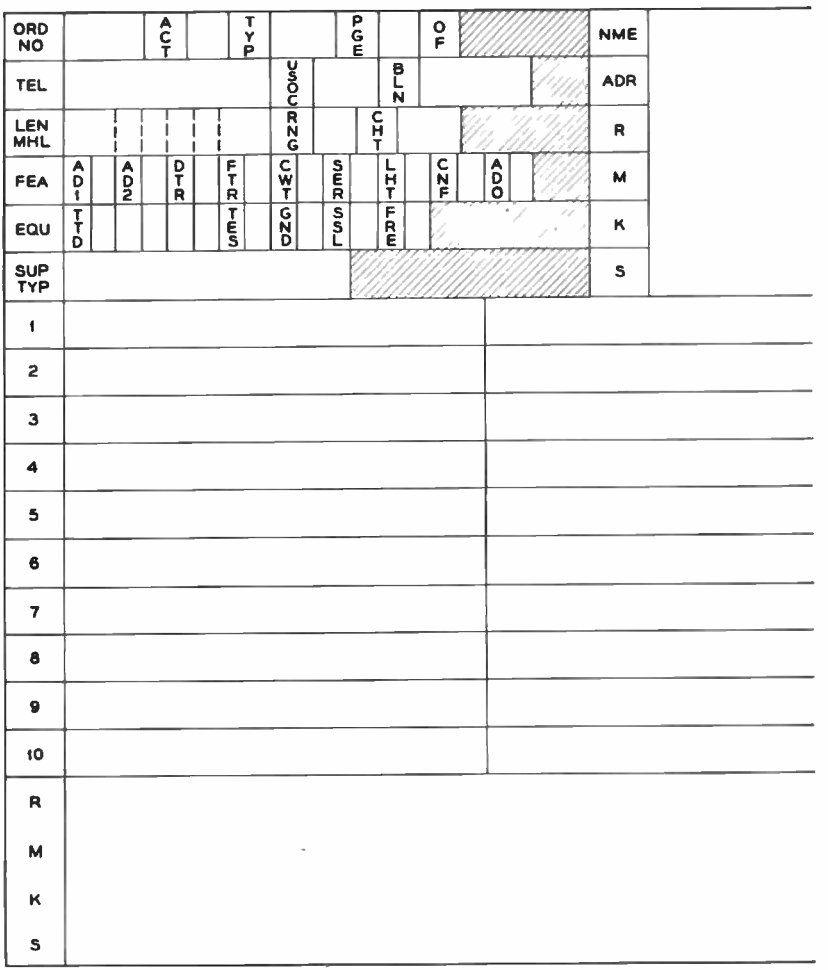


Fig. 14 — Service order format.

be delayed. The service order type (in order, out order, change directory number, change class of service, etc.) indicates how to process the following data.

The fields on the second line are reserved to hold directory number, unified service order code (USOC), and billing number, if it differs from the calling directory number. The third line has the fields for the line equipment number, or the MHG number if the service order refers to a multiline hunting group, the type of ring, and the chart class. The next two lines have the fields for all service and equipment features. A check in one of them indicates the existence of the feature in question. Then follows a field for the type of supplementary data. If this type is not blank, the supplementary data, such as an abbreviated dial or transfer list, or sleeve lead data, or a multiline hunting list, follow on the lines underneath.

When an RC is introduced via the teletypewriter, the data are selected from the appropriate fields as a function of the service order type, and digested. If the format is not correct, the RC is rejected. Otherwise, the code in the field is translated from telephone company language into the cipher used inside ESS. For example, unified service order code (USOC) 1FR (one party flat residential) is translated into the codes for major class = individual line, and chart class = flat rate; line equipment number 02425212 is translated into its binary form 00101001101101100. The translated item is then stored in the proper place of the auxiliary block of memory in the call store which is seized at the start of the assembly of any new RC. The proper position within this block is determined by a pattern decided upon by an examination of early data. If the pattern allows a variable-length block of memory, the largest size is selected initially and revised downward as more information is absorbed.

In this way, the RC is gradually built up in the auxiliary area regardless of whether the block will finally be retained or not. At the end of the assembly, a compressibility check is made. If the class of service allows an abbreviated code, the auxiliary block is abandoned, and only a primary entry with complete data remains. If the data cannot be compressed, the primary entry is made up with the reference to the auxiliary block already prepared. In any case, the primary RC is entered into an RC buffer register.

The RC buffer register is an intermediate storage location in which the primary RC's are stored before being inserted in the ordered RC area. This buffer storage is necessary so that the actual insertion can take place whenever spare time is available in the system.

Supplementary data, such as abbreviated dialing or fixed transfer lists which are not part of an auxiliary block, are also temporarily assembled in the auxiliary area. For each of the items in the lists, a primary RC entry will then be made as room in the RC buffer allows.

Insertion consists of the following steps: first, the point of insertion in the ordered list is determined. Then a search is made for an inactive RC register in the neighborhood of the point of insertion. If such a "hole" is found, all entries between the point of insertion and the hole are moved to create a "hole" at the point of insertion. Finally, the primary RC from the buffer is inserted there.

The RC buffer is administered sequentially, with the top entry following the bottom entry, and emptied on a first-in, first-out basis.

The time spent in the insertion process is unpredictable, because it depends on the number of RC registers between the point of insertion and the first available empty space. Occasionally, this time might exceed the allowable limit permitted by the main program of the system. Therefore, the insertion sequence is temporarily interrupted if the allowable time has expired, and continued later when time is again available.

In order to minimize insertion time, an effort is made to scatter the active RC's over the entire primary area. This effect is achieved by merely changing the status bits of an RC that is to be removed to the "deleted" or inactive code. Eventually, it will be overwritten in the insertion process, since a deleted status is the indication of an available space.

No action is taken to delete auxiliary blocks. As soon as the reference to them becomes inactive in the primary RC, they are "dead," scattered between still-active blocks. A consolidation routine is required which is performed every night. This routine rearranges the scattered active blocks into a contiguous file. It is a very time-consuming routine. It runs through all primary RC's, singling out those which have a reference address to an auxiliary block in the still unarranged area, comparing the reference addresses, and in the end arriving at the RC with an auxiliary block nearest to the end of the consolidated area. This block is then moved to close the gap and its reference address is changed. The process is repeated until all "dead" spaces are squeezed out.

Most service orders request a delayed activation of the change. In this case, the primary RC or RC's for line equipment number and/or directory number are given delayed status until activation is called for by dialing the service order number from a special telephone line termination.

In order to preserve the connection between service order number and resulting RC's, an RC entry with the service order number as TAG is made, whose primary TW contains either a line equipment number, a directory number or an MHG number to identify the primary RC's which have to be activated. If more than one such identifier is needed, a reference address leads to an activation block which contains them. At activation time, the entry which contains the service order number is deleted, while entries identified by the service order number are given "permanent" status.

Changes on trunk circuits or office codes are treated in a way similar to those originated by service orders. They are normally introduced over the maintenance channel of the teletypewriter. They lack a service order number and therefore cannot be accepted on a delayed basis. Although they use telephone company terminology, their format is closer to the needs of ESS.

VII. PROGRAM STORE UPDATING

When recent changes have accumulated in the call store, they are transcribed onto the program store. The future location of primary RC's is predetermined by their TAG. The future location of auxiliary blocks is chosen when the RC is received, and the location stored as the permanent auxiliary address in a control word preceding the auxiliary block. This control word is not part of the auxiliary block and is not carried over to the program store.

It is required that any number of modules be updated at a time and also that the updating of a module may be interrupted at any time without harm.

For each card of a module about to be changed, a card image is prepared in the call store by copying into it the contents of the old PS card. Then a search is made through the RC area to find the RC data modifying the card image. The address of the 64 words in the card corresponds to 64 program store addresses. Whenever a permanent RC with a TAG equaling one of those addresses is found, the corresponding card image location is changed. If the primary TW contains anything but a temporary auxiliary address, it is transcribed as such. If the primary TW contains a temporary auxiliary address, the control word of the auxiliary block containing the future permanent auxiliary address is transcribed instead (see Fig. 15).

If the primary TW contains a temporary auxiliary address, whether

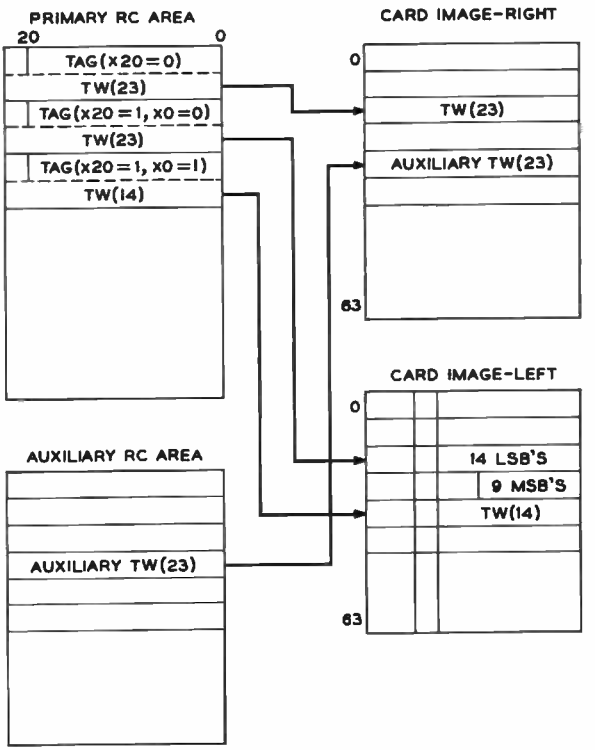


Fig. 15 — Transcription of recent changes from the RC area to the card image.

the TAG belongs to the card or not, the permanent auxiliary address is examined and the following cases are distinguished:

(a) None of the auxiliary TW's belong on the card.

(b) The auxiliary block starts on the card. It may or may not end on it.

(c) The auxiliary block ends, but does not start, on the card.

(d) Sixty-four auxiliary TW's in the middle of the auxiliary block belong on the card.

Whatever portion of the auxiliary block belongs on the card is then transcribed onto the card image.

The card images thus prepared are used to write new memory cards. After a module has been written and inserted into the PS, a limited verification takes place. An error in a changed word does not cause rejection, since the RC is still available.

After an arbitrary number of modules has been written, verified, and duplicated,* the RC area is updated; the RC data which were correctly transcribed to the PS are eliminated as follows:

All primary RC's with permanent status are examined, and the following cases are distinguished:

(a) The primary TW contains anything but a temporary auxiliary address. The primary TW in the RC area is compared with its counterpart in PS. If both are equal the RC is deleted.

(b) The primary TW contains a temporary auxiliary address. Each auxiliary TW in call store is compared with the contents of the program store words at the location of the permanent auxiliary address. If all words match, the block is considered correctly transcribed. If the primary TW in PS already contains the permanent auxiliary address, the primary RC is deleted. If it does not yet have the right reference address, the primary TW in the RC area is changed by replacing the temporary with the permanent auxiliary address. If the block was not correctly transcribed, no action is taken.

This method of updating the RC area explains why the limited verification method is permissible: if a changed word is incorrectly transcribed to the PS, no harm is done, since the RC is not deleted. It will be corrected at a later time.

This method of updating also makes the writing of modules independent of each other. For instance, if an auxiliary block overflows from one module into the next one, and if either module is revised, the RC will stay intact until the other part is done. Or, if the primary RC is in one module, the auxiliary block in another, either module may be written without consideration of the other one. Also, the writing of a module can be interrupted without any damaging effect, since no changes in the RC area are made before the three steps of writing, verifying and duplicating are completed.

Available space in the PS translation area is administered via linked lists of available space. The lists are first created in PS when the original translation data are installed. Lists are maintained for spaces of 2, 3, $\cdots$, 31 words, and one list is maintained for larger spaces, both in the right and the left halves of the PS. Each of the larger spaces contains the address of the next large space and contains its own length. As new space is needed, or as active space is relinquished, the linked lists and their headcells are updated via the recent change mechanism. If space

* The first of the two duplicate modules is written as described above. The second is merely copied from the first one.

is seized from a larger area, the length information for this area is updated.

VIII. CONCLUSION

The translation plan for No. 1 ESS has accomplished a number of goals. It has provided:

(1) compact storage of subscriber, trunk and office code data — data that are currently stored in cross connections in electromechanical systems,

(2) convenient means for handling changes in such information,

(3) facilities for providing much of the information that permits a generic program to handle a specific No. 1 ESS installation,

(4) convenient forms of input and output data for use by a generic office program, and

(5) convenient means for introducing translation information changes in a working office.

Translations have been a major tool in making a generic No. 1 ESS possible and efficient.

IX. ACKNOWLEDGMENTS

Many of our colleagues have made substantial contributions to the No. 1 ESS translation scheme. In particular, we would like to acknowledge the contributions of J. H. Carran, D. F. Peckens, D. P. Bannon and H. R. Appenzeller, who worked out the plan of associating the various equipment numbers in the ESS and who worked out the cutover forms; Mrs. E. Fong and Miss S. H. Wiley, who programmed and debugged much of the basic translation program; Miss B. Reaugh, who programmed the translation information change program; Mrs. A. C. Some, who did much of the early work of deducing the translation requirements; W. G. Feger, W. H. List, and R. R. Scotson, who are writing the program to convert the data on the cutover forms into data to be stored in translation tables; and L. S. Tuomenoksa, who did much of the work on an earlier system that was a sound starting point for the present scheme. In addition, we would like to thank our colleagues at Bell Laboratories who, in deriving system programs, discovered many of the requirements for the translation and made many valuable suggestions, and our colleagues at New Jersey Bell Telephone Co. American Telephone and Telegraph Co., and Western Electric Co. who supplied so much sound constructive criticism in the derivation of useful translation forms.

REFERENCES

1. Carbaugh, D. H., Drew, G. G., Ghiron, H., and Hoover, Mrs. E. S., No. 1 ESS Call Processing, B.S.T.J., this issue, p. 2483.
2. Ault, C. F., Gallaher, L. E., Greenwood, T. S., and Koehler, D. C., No. 1 ESS Program Store, B.S.T.J., this issue, p. 2097.
3. Harr, J. A., Taylor, F. F., and Ulrich, W., Organization of No. 1 ESS Central Processor, B.S.T.J., this issue, p. 1845.

System Testing of the No. 1 Electronic Switching System

By G. HAUGK, S. H. TSIANG and L. ZIMMERMAN

(Manuscript received January 27, 1964)

Plans for testing the hardware and software and for evaluating overall system operation of the No. 1 electronic switching system are described. Program and hardware test facilities and the early results achieved using these facilities on the first two No. 1 ESS installations are presented.

I. INTRODUCTION

Planning for system testing started at the beginning of the No. 1 electronic switching system (ESS) development.[1] Test facilities were designed concurrently with the design of the system. This paper describes the test plan being followed, the test facilities that are being used and the results that were obtained on the first two No. 1 ESS installations.

The first No. 1 ESS, located at the Holmdel Laboratories, is being used for checking system design. The second No. 1 ESS at Succasunna, N. J., is the first of a large number of systems scheduled for commercial service.

Fig. 1 broadly illustrates three sequential periods of testing. Factory tests are followed by system tests and, after cutover of an office to service, by maintenance tests.[2] Fig. 1 also depicts system evaluation, an activity which has its beginnings in the planning and design stages and carries through many issues of a new system.

Factory testing is a subject on which there are many views. Most views include some degree of device and package testing, and inspection for workmanship — such as the quality of the wired connections. Beyond that they range from continuity checks of mounting plate assemblies and major units to functional tests of groups of major units or subsystems, and on out to rather complete system testing. The major units of the Holmdel system, after very little factory testing, were shipped directly to Bell Laboratories for extensive design testing prior to use in the

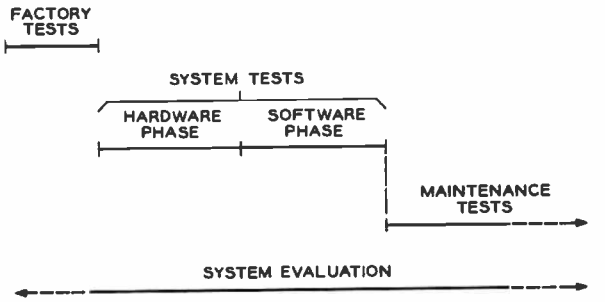


Fig. 1 — Test diagram.

system. With the availability of test specifications, the major units for the Succasunna system underwent considerable factory testing prior to shipment.

The system test interval (see Fig. 1) consists of two sequential phases, hardware and software (program). At installations beyond the early No. 1 installations the software phase will largely disappear. The debugging of programs which provide new features will be accomplished on the Holmdel system. When this point is reached, testing at the field site becomes what is known as "installation testing." This paper deals with system testing. However, much of the test planning and many of the facilities and techniques for system testing can be and are being applied to installation testing by the Western Electric Company.

II. HARDWARE TESTING

System hardware testing checks the proper functioning of all units as a system. The major units of the No. 1 ESS are the central control, program store, call store, central pulse distributor, and peripheral units. The peripheral units include the signal distributor, scanner, network, and master control center. The teletypewriter, automatic message accounting unit, memory card (program store memory) writer, and control display and test panel are all parts of the master control center.

For the first No. 1 ESS system at Holmdel, the hardware testing was intended to prove the system design both from the hardware and logic standpoints. Although most units had been individually tested, they had never worked together as a system. Many maintenance features,[2] such as those in the central control, could be tested only during system testing.

The hardware testing is done in two stages, manual and program. Manual testing comprises only a very small part of the total effort. The

hardware testing is essentially accomplished by program means. Special programs, called "X-ray programs," were designed for this purpose. The term "X-ray" is used to convey the idea of examining the internal and basic functions of the system. Two aspects of hardware testing are the detection and location of troubles. The latter is by far the more difficult. One objective in the design of the X-ray programs was to simplify the method used in hardware testing. The programs developed not only detect troubles at system speed but also help in locating them easily and quickly.

2.1 *Manual Hardware Testing*

Manual hardware testing includes all tests not performed by use of X-ray programs. Many of these tests, such as power testing and continuity checking of interunit wiring, are rudimentary in nature and perhaps not properly classified under system testing. On the other hand, some manual testing, particularly that carried out on the Holmdel system, can be considered system testing — for example, the pulsing and monitoring of the systems communication buses[3] using pulse generators and oscilloscopes to ferret out system noise problems.

In the No. 1 ESS the central control is the basic control unit and the most complex unit in the system. It governs the flow of information and coordinates the action of all other units. The program store is used to store the program of the system and data such as the translation information associated with each customer.[4]

In the plan that is being followed for No. 1 ESS testing, the goal is to bring the central control and the program store to a state where they can be operated together on X-ray programs as soon as possible in order to fully test themselves and other units of the system. With a proper installation sequence, manual testing and in fact X-ray testing, can progress while many major system units are being installed. The total hardware test period, therefore, can be shortened.

Manual testing of the central control and program store primarily consists of testing the central control's ability to send addresses to and receive instructions from the program store. In addition, central control's ability to execute a few simple instructions is checked before X-ray programs are used. At the conclusion of these manual tests only a very small fraction of the vast amount of circuitry in the central control[5] has been tested. The manual testing of the central control and program store is carried out with the aid of a central control manual tester (see Fig. 2). This test set is mobile and plugs into a central control. Using

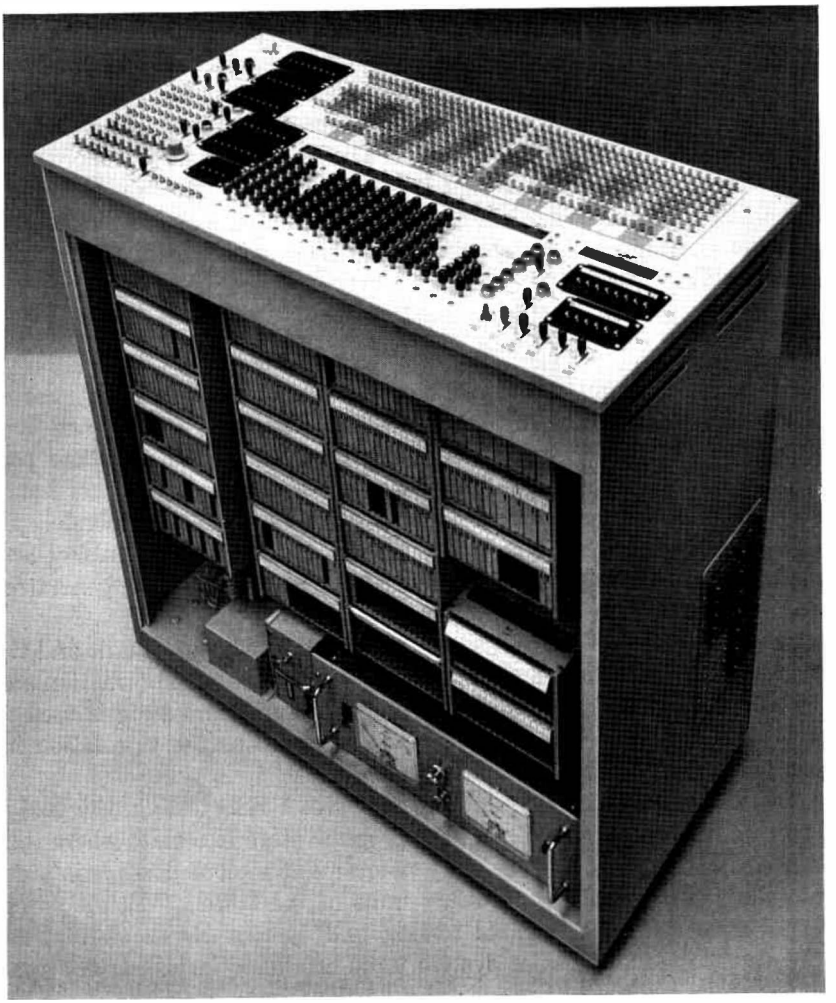


Fig. 2 — Central control manual tester.

this set one is able to

(1) insert instructions and control the execution of instructions by central control,

(2) simulate the response of any unit to central control,

(3) monitor key points in the central control,

(4) monitor outputs of the program store, call store and scanners, and

(5) generate signals to stop or interrupt[4] the system or automatically

insert program instructions when a selected program store or call store address is reached.

The last feature is used when a central control is under program control. In fact, the central control manual tester is the primary tool used with the X-ray programs.

2.2 *Programmed Hardware Testing*

The No. 1 ESS is a program-controlled machine having a high degree of control centralization. This made possible the use of programs, very powerful tools, for system testing. Also, because of the high degree of control centralization, common testing is emphasized. That is, once the central control operates with the program store, it is used to test itself and the rest of the system in a "bootstrap" manner.

The basic central control circuits are tested first. Using the basic central control functions, the central pulse distributor is tested next, and in the process additional central control functions are tested. In a similar manner, the call store and the peripheral units are added in turn. The unit to be tested is selected by switches on the central control manual tester.

The X-ray programs are designed to test at operating speed bit-by-bit and function-by-function a system which has not been previously operated. The underlying principle employed in the design of the X-ray programs is to start with simple tests to check out basic circuits, then gradually extend into other circuits using, insofar as possible, only previously tested circuits. Each test in the X-ray program checks for a known test result and is designed to check a particular circuit or function.

A generalized sequence chart for the X-ray programs is shown in Fig. 3. The programs go from test to succeeding test until a failure occurs. When this happens, a transfer is made to a failure leg. What happens in this leg is dependent upon the circuit-controlled option which has been selected by the user. These options are provided in the central control manual tester which is plugged into the system during the X-ray tests. The options are stop, record and advance, and recycle.

Stopping freezes the machine in a state as close as possible to the trouble condition. Pertinent information about the state of the system at the time the error occurred is preserved and displayed on the central control manual tester.

Record and advance is an attempt to save the data in the central control registers and continue. The data are typed out via a teletypewriter at the end of every test that fails.

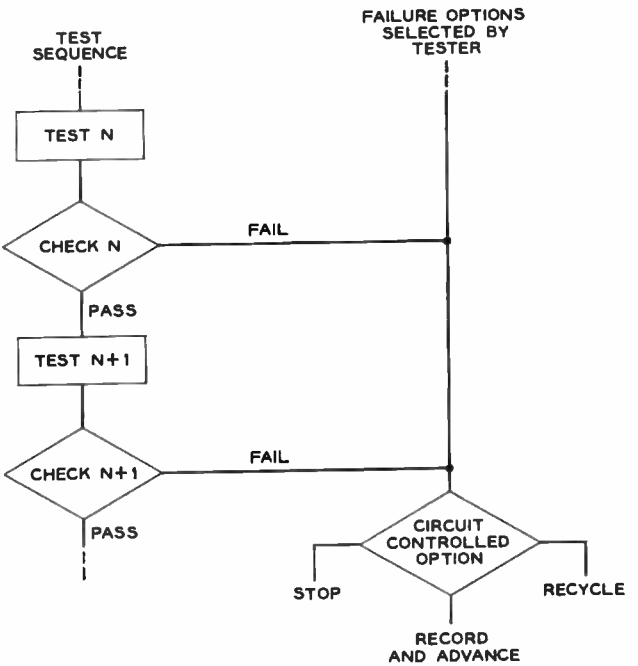


Fig. 3 — Generalized X-ray sequence.

Recycle provides a means to continually repeat the test at system speed. Therefore, any part of a circuit can be examined with an oscilloscope under dynamic conditions.

The normal mode of operation is to use the stop option to detect a trouble. With the machine stopped, lamps on the central control manual tester (J-register display)[4] are used to determine the program store address of the test which failed. For each test a comment indicating the possible source of trouble, should the test fail, is given in the program listing. Other lamp displays on the central control manual tester yield further information regarding the trouble. At this point the trouble is localized. Additional information about the trouble can be obtained by recycling the test. Using this procedure, which requires the use of the X-ray program listing, circuit schematics and the oscilloscope, a trouble is usually tracked down quickly.

The X-ray programs for most of the system units are divided into two parts, single system and duplicate system. A single-system X-ray is used to test a single unit. A duplicated system X-ray tests multiple-unit ar-

rangements and maintenance features. All X-ray tests are made with a central control-program store combination.

For reliability and maintenance purposes[2] all units, with the exception of those which affect only single customers, are duplicated. The duplicated units normally run in parallel. Each is checked against its mate by match circuits while performing identical functions. It is possible, however, with minor hardware modifications, to split the system into two independent systems. Such an arrangement, using a central control manual tester on each system, has been used during single-system X-ray testing. This method of operation accelerates single-system X-ray testing, thereby reducing the hardware testing period. For duplicated system X-ray tests, the split system arrangement is not applicable.

The X-ray programs used prior to cutover and the maintenance programs used after cutover of an office to service are similar, since they both test the same hardware. In what follows, some of the factors which caused differences are discussed.

Maintenance programs have stringent real-time and complicated interface requirements because they must share real time with call programs. The X-ray programs, used before a system is providing telephone service, are not time-shared with call programs. In addition, the X-ray programs need not share the system memories with call programs.

The primary maintenance tool used when a No. 1 ESS is giving telephone service is the dictionary.[8] A maintenance dictionary is a table relating the printouts of diagnostic test results with corresponding faulty plug-in packages. The dictionary technique is built upon the assumption that a trouble has just occurred in a working system. However, at the outset of the hardware phase of system testing, multiple faults must be assumed in the system and the dictionary technique is not applicable.

The maintenance programs have available to them all of the hardware maintenance facilities — for example, working teletypewriters and match circuits. On the other hand, during much of the hardware phase of system testing the hardware maintenance facilities are not usable as tools.

The dictionary technique makes use of the pattern of test failures which results from a series of tests applied over large amounts of circuitry. The X-ray technique makes use of the information at the first test that fails in a series of tests, each of which is applied over small amounts of circuitry.

In Table I these factors are summarized. In the last column of the table an S or C indicates whether a factor results in simplification or complication in the design of the X-ray programs.

TABLE I — TEST PROGRAM COMPARISONS

Factor	X-Ray	Maintenance	Result
Time shared with call programs	no	yes	S
Memory shared with call programs	no	yes	S
Use of auxiliary test sets and instruments	yes	no	S
Multiple faults in every unit	yes	no	C
System maintenance facilities available	no	yes	C

The assignment of central pulse distributor, scanner, and signal distributor points varies from office to office. The X-ray program obtains the addresses of these points and other items which vary from office to office by reference to a translation area,[6] provided for system programs, in the program store. Thus the X-ray program is usable in any No. 1 ESS installation.

III. SOFTWARE TESTING

The large system program for No. 1 ESS is at least as complex as the hardware used to carry out the program. The early systems are being used for design debugging of the software as well as the hardware.

3.1 *Hardware Facilities*

During system testing of No. 1 ESS, input-output, control, and monitoring facilities, which can be used for program debugging, are provided by the central control manual tester and the system's maintenance teletypewriter, automatic message accounting magnetic tape unit, and control display and test unit. However, because of the large amount of program debugging being done at the Holmdel Laboratories, additional facilities were added to the Holmdel system to provide greater speed and flexibility.

The additional input-output facilities provided at Holmdel consist of a card reader, a high-speed printer and a magnetic tape unit.

A card reader, which reads 100 cards per minute, is used as an alternative to the teletypewriter to load information into the ESS. This means of loading information is a more reliable means of repeatedly introducing information than the teletypewriter, is more flexible and is about 10 times faster than teletypewriter tape. The information is read into the system via No. 1 ESS scanners. This reader has also been provided at Succasunna for use in program debugging.

A high-speed printer, which prints 80 characters per line at 1000 lines per minute, is used as an alternative to the teletypewriter to dump information from the ESS. It is more than 100 times faster than the teletypewriter. The printer provides a practical means of obtaining large amounts of data from the system without excessive use of machine time and eliminates the delay that is encountered in obtaining printed copy if magnetic tape is used. The system communicates with the printer via the peripheral bus.[3]

The tape reader assembler and processor (TRAP) is a facility used to control the transfer of data from magnetic tape to program store twistor cards via the memory card writer. TRAP has been provided at both the Holmdel and Succasunna installations because of the frequent changes required on twistor cards during the program debugging period. In addition, the TRAP unit at Holmdel has been modified and connected to the system's peripheral bus so that its magnetic tape unit can be used in loading and unloading system information. The magnetic tape unit is about 12 times faster than the high-speed printer.

At Holmdel a program test console (console) is used in place of a central control manual tester; it provides greatly expanded monitor and control facilities. In appearance, the most striking difference between these units (see Figs. 2 and 4) results from the 2544 monitor lamps on the console as opposed to a 442-lamp display on the central control manual tester. The console simultaneously monitors 864 points in each central control. These points include almost all of the central control flip-flops. The remaining lamps are used to monitor key points in all other units of the system. The view of system status provided by these lamps is especially important in debugging maintenance programs.

The console also provides three additional displays. A cathode ray tube display of the program flow is obtained by using a digital-to-analog converter on a program store address register. Because the No. 1 ESS is controlled by a repetitive executive control program, this display provides a picture which gives immediate indication of program response to external inputs and detection of anomalies in program flow.

For call stores, in which 24-bit words are stored, there are two console displays. One makes use of program control to display the contents of any set of 31 call store locations as a 31×24 array of spots on a cathode ray tube. Another circuit, not program-controlled, uses a bank of 24 lamps to provide a continuous display of the contents of any selected call store location.

The console and central control manual tester both provide similar control of the central control clock for stopping, manually stepping, and

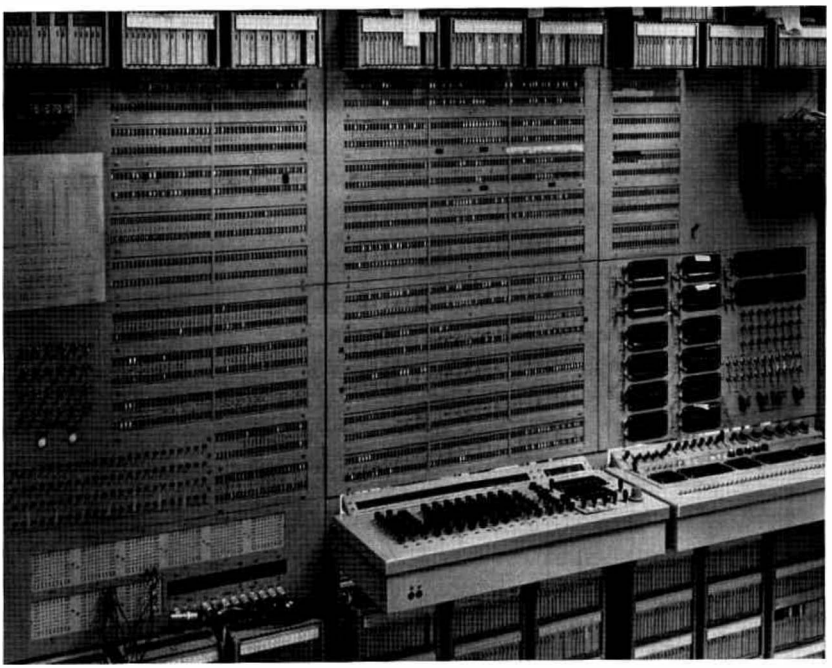


Fig. 4 — Program test console.

inserting simulated inputs. The console connects to all units of the duplicated system, while a central control manual tester can be connected to only a single central control. The console is capable of more fully controlling the duplicated system, as well as being able to control either half. In addition, the console provides manual control over many of the important maintenance features such as the inhibition of system interrupts.[4]

A simplified picture of program debugging closely parallels that of circuit debugging. Inputs are applied and outputs are checked. If trouble is suspected, internal conditions must be examined. To obtain easy access to information about any selected point in a program, flagging facilities are provided in the console and the central control manual tester. A flag is a signal generated under specified conditions which can be used to (a) stop the system, (b) light a lamp, or (c) interrupt the system. The console has provisions for generating fourteen distinct flags. The mechanisms for generating flags are:

(1) program address match — There are eight sets of switches on which program addresses can be set up. Whenever the central control

addresses the program store to an address which has been selected, a flag is generated.

(2) call store address match — There are four sets of switches used to select call store addresses in a manner similar to selecting program store addresses. In addition, reading or writing and/or a bit configuration for the data may be specified as additional conditions on the generation of flags.

(3) call store block match — There are two call store block match circuits identical to the call store address match circuits except that two sets of switches are provided with each circuit to define a range of addresses. The address condition for the flag is satisfied whenever a call store address is used which falls within a selected range.

3.2 *Programmed Facilities*

To expedite program debugging a programmed ESS utility system has been designed around the flag generators. The teletype or card reader is used to insert information into the system to define the function to be performed when a flag generator causes an interrupt. The function to be performed can be made up of any combination of the following:

(1) dump — The contents of call store or central control memory locations are written on the teletypewriter, high-speed printer, or TRAP magnetic tape.

(2) write — Information previously entered in the system (via card reader, teletypewriter or magnetic tape) is transferred into preselected call store or central control memory locations.

(3) trace — This sets up a central control mode which causes a dump to be performed at each program transfer.

Upon the completion of any of the above items, program control is transferred back to the point which was interrupted.

(4) jump — Jump specifies a program address to which control is transferred.

(5) patch — This causes the system to transfer program control to instructions which have been previously written in the call store.

The utility system outlined above was designed to be usable, with some restrictions, for program debugging using the central control manual tester. Most of the restrictions arise because the central control manual tester does not have any call store block match circuits and has only two program store and two call store address match circuits.

The program debugging facilities described were designed so that program test conditions could be submitted to the system and results obtained with a minimum expenditure of ESS machine time.

IV. OPERATIONAL TESTING

Operational testing verifies that the hardware and the system program satisfy system requirements. So as not to delay the introduction of a new system into service, this phase of testing must be started early enough to insure time to take corrective action as required.

The early systems must be subjected to extensive operational testing. These tests, among other things, must

(a) check all call features without and with traffic,
(b) verify traffic handling capability,
(c) verify traffic counts,
(d) verify AMA accuracy,
(e) check all maintenance features,
(f) evaluate maintenance dictionaries,
(g) check system capability at temperature and voltage limits, and
(h) check transmission and crosstalk characteristics.

At later installations these items will be tested, but much less extensively. In addition, installation tests include other items such as

(a) verify all translations (lines, trunks, etc.),
(b) test all lines, and
(c) test all trunk circuits.

V. EXPERIENCE

Fig. 5 presents a simplified picture of key events in the testing of the Holmdel system during the year 1963. The arrival times of the central controls (CC's) and the program stores (PS's) are shown. System testing did not really start until the receipt of program store 1. Prior to that time central control 1, which was shipped from the factory without benefit of factory testing, underwent extensive manual testing. The early manual tests were primarily of continuity, while later tests were functional, using program instructions inserted via the central control manual tester. Intervals for the wiring of major central control design changes are also indicated. The start of the use of X-ray programs, the beginning of system program debugging on one shift, and the expansion of this effort to two shifts as well as the first No. 1 ESS telephone call made using an X-ray program are also shown.

Fig. 6 presents a simplified picture of key events in the testing of the Succasunna system.

Figs. 7 and 8 show the total number of troubles cleared per month at Holmdel and Succasunna during 1963. They are broken down into three categories: wiring, circuit pack, and other troubles.

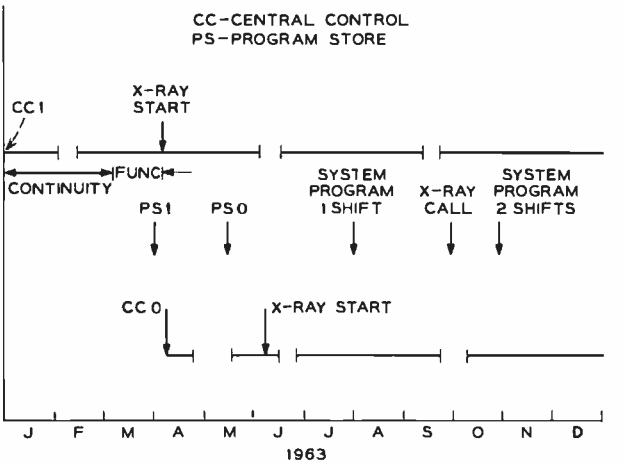


Fig. 5 — Holmdel system events.

Circuit pack troubles were running about 35 a month on the Holmdel system and somewhat higher than this at Succasunna at the end of 1963. There are approximately 11,500 circuit packs in each of the systems. Experience with the Morris ESS[7] and other electronic systems indicates that an order-of-magnitude reduction can be expected in the package failure rate as an electronic system moves from the testing and wiring change period into service.

Many of the earlier wiring troubles encountered at Holmdel were attributable to the factory. A lower proportion of the wiring troubles at

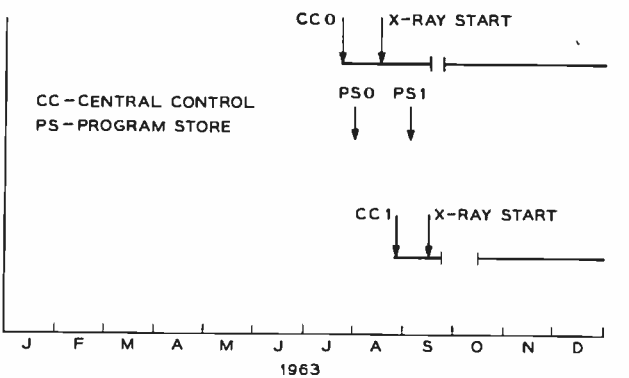


Fig. 6 — Succasunna system events.

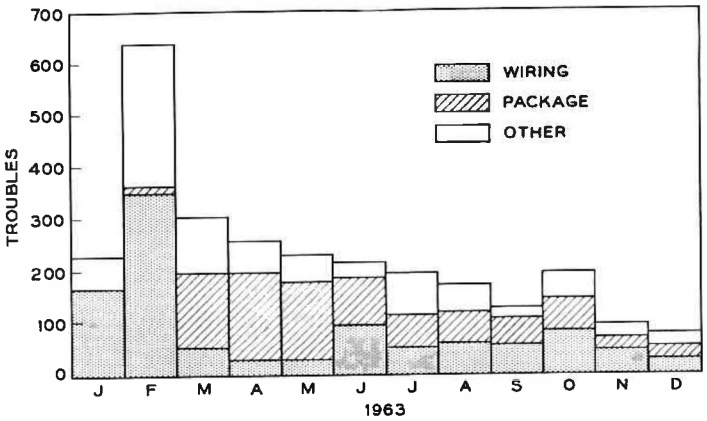


Fig. 7 — Troubles cleared per month at Holmdel.

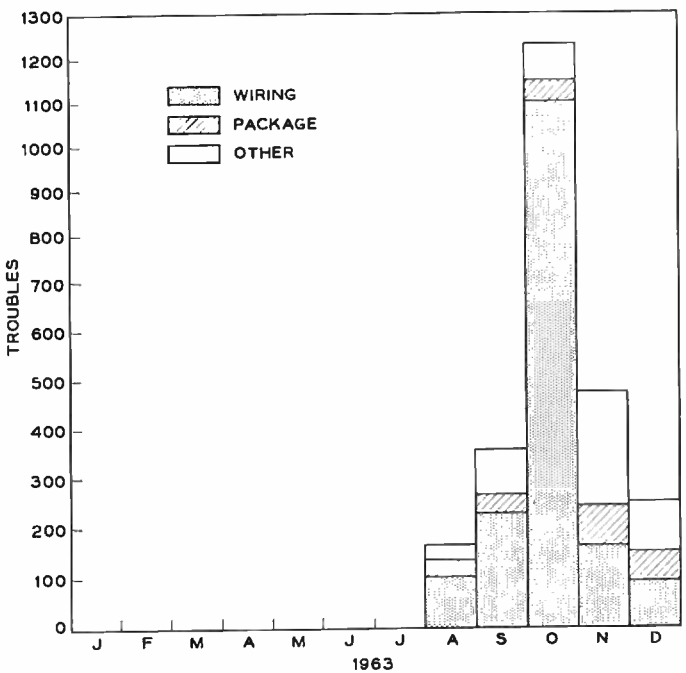


Fig. 8 — Troubles cleared per month at Succasunna.

Succasunna are attributable to the factory. They, like the later wiring troubles at Holmdel, are more frequently chargeable to installation and design change wiring.

Included in the category of other troubles are logic and circuit design problems uncovered in system testing. This source has been a major contributor to design changes necessary to realize an operational system.

The principal sources of program problems are programming error, clerical error, requirement change and program improvement. Fig. 9 shows the number of program problems encountered per month on the Holmdel system during 1963. These were found and corrected while debugging X-ray and system programs. Since Holmdel is the first No. 1 ESS, it was necessary to do much of this work, particularly the debugging of X-ray programs, concurrently with the debugging of the hardware. On the average, one program problem was cleared for every 55 program words verified.

System testing using X-ray programs read from program stores began at Holmdel on April 5, 1963 (see Fig. 5) and at Succasunna on August 16, 1963 (see Fig. 6). This work was carried out on a two- and three-shift basis at both places. Fig. 10 gives the number of troubles cleared per shift-month at Holmdel and Succasunna and makes a comparison by referencing them to the month in which X-ray testing started at each location. As can be seen, troubles were found at a faster rate at Succasunna than at Holmdel. The major reason for this is that testing on the Succasunna system made use of X-ray programs that had already been debugged on the Holmdel system.

The size of a system hardware testing task is dependent not only on the number of system units involved but also on their complexity. Complex units have larger X-ray programs associated with them than less complex units. This suggests better ways of measuring the size of a system hardware testing task than merely totaling the number of units in-

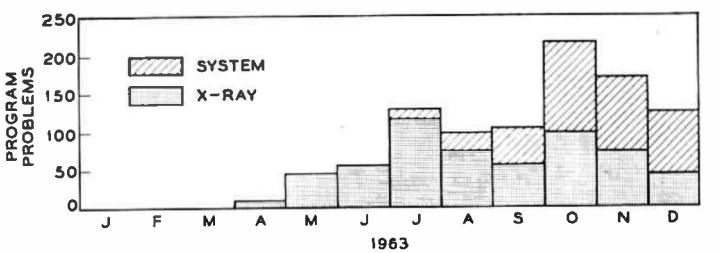


Fig. 9 — Program problems encountered per month on the Holmdel system.

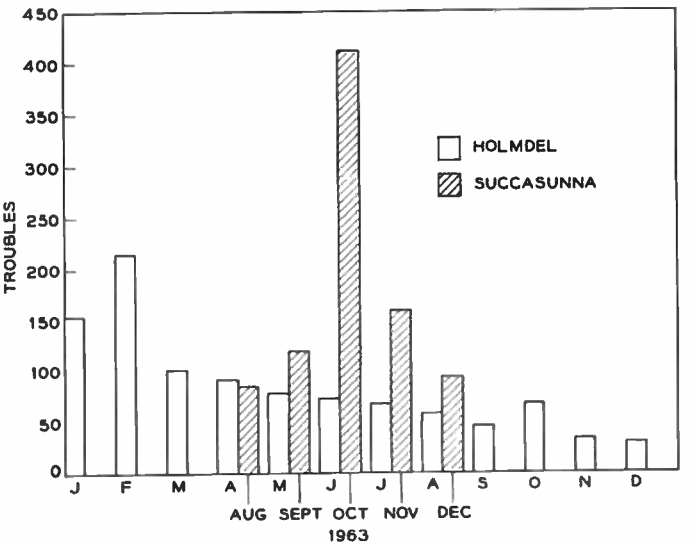


Fig. 10 — Troubles cleared per shift-month at Holmdel and Succasunna.

volved. A more meaningful and easy-to-use measurement is the sum of the products of the number of units and the number of associated X-ray program words, unit-words. This concept is illustrated for several No. 1 ESS units in Table II.

Using cumulative unit-words per shift, a comparison of the initial progress at Holmdel and Succasunna is given in Fig. 11. The solid-line curves show the progress by actual date. The dashed-line curve references the progress at Succasunna to the Holmdel X-ray start date. The more rapid progress at Succasunna was made possible by the use of debugged X-ray programs and by the absence of troubles common to both systems that had already been cleared at Holmdel.

Fig. 12 is a plot of the progress made in debugging X-ray programs concurrent with their use in debugging the Holmdel system hardware. As can be seen, the verification of these programs was nearing completion

TABLE II — UNIT-WORDS FOR REPRESENTATIVE UNITS AT SUCCASUNNA

Unit	Number	Words/Unit	Unit-Words
Central control	2	19,100	38,200
Central pulse distributor	2	840	1,680
Call store	4	3,750	15,000
Master scanner	2	1,120	2,240

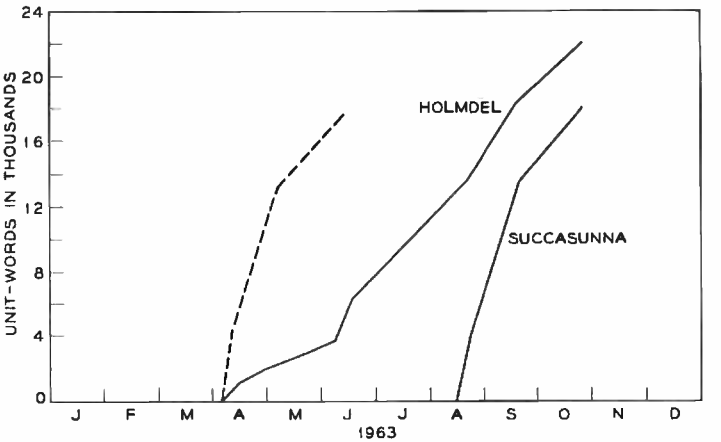


Fig. 11 — Cumulative unit-words per shift at Holmdel and Succasunna.

at the end of 1963. Also shown is the early experience in debugging system and utility (see Section III) programs. By using the data given in this figure and the dates on which system program debugging started as a one-shift and later two-shift operation (see Fig. 5), it can be calculated that an average rate of 2000 words per shift-month was realized.

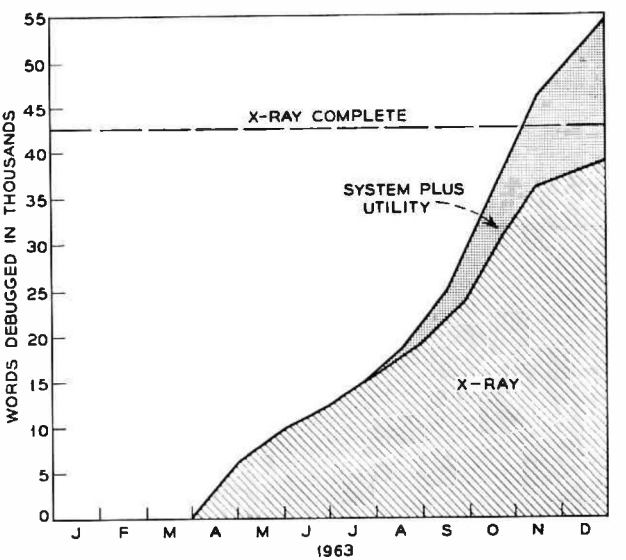


Fig. 12 — Program debugging progress on the Holmdel system.

VI. ACKNOWLEDGMENTS

The authors wish to thank their numerous colleagues who participated in test planning, the development of test facilities and programs, and in carrying out the testing reported on in this paper. The contributions and assistance of T. T. Butler, R. S. Cooper, T. S. Greenwood, B. G. Hemmendinger, F. R. Huff, J. L. Kodner, H. D. Straley and D. R. Teeters deserve particular mention. The advice and encouragement of H. E. Vaughan has been of great value.

REFERENCES

1. Keister, W., Ketchledge, R. W., and Vaughan, H. E., No. 1 ESS System Organization and Objectives, B.S.T.J., this issue, p. 1831.
2. Downing, R. W., Nowak, J. S., and Tuomenoksa, L. S., No. 1 ESS Maintenance Plan, B.S.T.J., this issue, p. 1961.
3. Connell, J. B., Hussey, L. W., and Ketchledge, R. W., No. 1 ESS Bus System, B.S.T.J., this issue, p. 2021.
4. Harr, J. A., Taylor, F. F., and Ulrich, W., Organization of No. 1 ESS Central Processor, B.S.T.J., this issue, p. 1845.
5. Cagle, W. B., Menne, R. S., Skinner, R. S., Staehler, R. E., and Underwood, M. D., No. 1 ESS Logic Circuits and Their Application to the Central Control, B.S.T.J., this issue, p. 2055.
6. Ulrich, W., and Vellenzer, Mrs. H. M., Translations in the No. 1 Electronic Switching System, B.S.T.J., this issue, 2533.
7. Haugk, G., and Yokelson, B. J., Experience with the Morris Electronic Switching System, IEEE Trans., Part 1, Comm. and Elect., No. 64, Jan., 1963, pp. 605–610.
8. Tsiang, S. H., and Ulrich, W., Automatic Trouble Diagnosis of Complex Logic Circuits, B.S.T.J., **41**, July, 1962, pp. 1177–1200.

Contributors to This Issue

CYRUS F. AULT, B.E.E.E., 1950, University of Southern California; M.S., 1955, Stevens Institute of Technology; Bell Telephone Laboratories, 1955—. He first worked on the electronic switching of the Morris, Illinois, trial installation. He is now working on the program store (twistor) of No. 1 ESS. Member, IEEE.

RULON BIDDULPH, B.S., 1925, Brigham Young University; Bell Telephone Laboratories, 1929—. Mr. Biddulph has engaged successively in investigations of the physiological acoustics of the hearing process, optical recording and reproduction of stereophonic sound, magnetic recording, antisubmarine subsurface ordnance, acoustic torpedoes, the structure and specification of speech, and the automatic recognition of spoken digits. He is currently concerned with transmission aspects of No. 1 ESS. Member, Acoustical Society of America.

ALBERT H. BUDLONG, B.N.S., 1945, B.E.E., 1948, M.S. (Physics), 1950, faculty, 1950–1953, Marquette University; Bell Telephone Laboratories, 1953—. He was initially associated with a group engaged in switching training and later supervised a group conducting switching training courses. He currently supervises a group engaged in the design of trunk and service circuits for No. 1 ESS. Member, Sigma Pi Sigma and Pi Mu Epsilon. Visiting Lecturer in Electrical Engineering at Stevens Institute of Technology.

WILLIAM B. CAGLE, B.S.E.E., 1953, University of Oklahoma; Bell Telephone Laboratories, 1953—. Mr. Cagle was first engaged in the design of semiconductor logic circuits for the Morris, Illinois, experimental electronic switching system. He subsequently supervised the characterization of the semiconductor devices, design of the logic circuits, and logic design of the central control and signal processor for No. 1 ESS. At present he heads a department responsible for development of the PICTUREPHONE set, mobile telephones, and exploratory development of lineless telephones. Member, IEEE, Tau Beta Pi, Eta Kappa Nu and Sigma Tau.

DAVID H. CARBAUGH, B.S.E.E., 1959, University of Pittsburgh; M.E.E., 1961, New York University; Bell Telephone Laboratories, 1959—. At the Laboratories he has been chiefly concerned with the system program development of No. 1 ESS.

R. C. CASTERLINE, B.S.E.E., 1956, Newark College of Engineering; Western Electric Co., 1941–1951; Bell Telephone Laboratories, 1951—. With Western Electric, Mr. Casterline was engaged in the inspection and testing of radio and radar equipment. With Bell Laboratories he was until 1956 a member of the component development department, where he was engaged in testing and evaluating ferroelectric devices and analog computer networks. In 1956 he transferred to the electronic switching department and has been concerned with circuit design for the Morris, Illinois, field trial and with trunk and service circuit development for No. 1 ESS.

J. G. CHEVALIER, B.E.E., 1951, Ohio State University; Bell Telephone Laboratories, 1956—. He has been engaged in various phases of printed wiring processes and applications for military projects and No. 1 ESS. More recently he has worked on the design and packaging of electronic equipment and the study of contact finishes.

JOSEPH B. CONNELL, B.S., 1958, University of Rhode Island; M.E.E., 1960, New York University; Bell Telephone Laboratories, 1958—. Mr. Connell worked on the maintenance planning aspects of the No. 1 ESS system design and served as an instructor in the communications development training program. He presently supervises a group responsible for generating systems engineering requirements dealing with new applications and new services for No. 1 ESS. Member, IEEE, Tau Beta Pi and Phi Kappa Phi.

DANIEL DANIELSEN, Ing. (Electrical Engineering), 1949, Stockholm Technical Institute; Bell Telephone Laboratories, 1953—. He was engaged in the circuit design and systems development of several phases of electronic switching systems. He has worked on the recording of AMA data on magnetic tape, data transmission over voice-frequency channels, and switching network design for No. 1 ESS. At present, he is concerned with the problem of interchange of data between No. 1 ESS and remotely located data recording systems.

H. J. DOUGHERTY, B.S.E.E., 1955, M.S.E.E., 1956, University of Maine; New England Telephone and Telegraph Co., 1941–1956; Bell

Telephone Laboratories, 1956—. He was first engaged in the design and development of circuits for the experimental electronic switching system. More recently he has been concerned with formulating systems engineering requirements for electronic switching systems. He presently supervises a group responsible for ESS operating requirements and maintenance. Member, IEEE and Tau Beta Pi.

RANDALL W. DOWNING, B.S.E.E., 1959, University of Washington; M.E.E, 1961, New York University; Bell Telephone Laboratories, 1959—. He has been concerned primarily with the problems of providing automatic maintenance facilities for electronic switching systems. He presently supervises a group responsible for the maintenance planning and programming for the control units of No. 1 ESS. Member, ACM, IEEE, Phi Beta Kappa, Tau Beta Pi and Eta Kappa Nu. Associate Member, Sigma Xi.

GLEN G. DREW, B.S.E.E., 1936, Rutgers University; Bell Telephone Laboratories, 1946—. After assisting in the development of automatic message accounting and crossbar tandem circuits, he worked with an exploratory development group studying the introduction of electronic techniques to telephone switching. Since then he has been engaged in various aspects of electronic switching developments. He currently supervises a group responsible for the introduction of metropolitan office features in No. 1 ESS. Member, Tau Beta Pi, Phi Beta Kappa and IEEE.

KERMIT S. DUNLAP, B.S. (Eng. Physics), Lehigh University, 1937; Bell Telephone Laboratories, 1937—. He first joined an acoustics research group, where he was concerned with tone generators for customer key pulsing systems. During World War II, he worked on antiaircraft fire-control analog computers. In 1946, he joined the newly formed electronic switching research group, where he was engaged in the first electronic switching experimental projects. In 1954, he transferred to the No. 1 ESS development department, where he now supervises the group on switching network design. Member, IEEE and Phi Beta Kappa.

R. K. EISENHART, B.S.M.E., 1957, Pennsylvania State University; M.E.E., 1959, New York University; Bell Telephone Laboratories, 1957—. Mr. Eisenhart was first engaged in the development of the automatic photographic processor used in conjunction with the flying-spot store of the Morris electronic switching system. At present he supervises

a group responsible for the design of apparatus and equipment used in No. 1 ESS. Member, Pi Tau Sigma and Tau Beta Pi.

ALEXANDER FEINER, M.S. (Electrical Engineering), 1952, Columbia University; Bell Telephone Laboratories, 1953—. He has been engaged in the application of electronic techniques to switching. At present he heads a department responsible for the development of switching networks, trunks and scanners, and for transmission aspects of No. 1 ESS. Member, Sigma Xi.

JAMES G. FERGUSON, B.S.E.E., 1923, Queen's University (Canada); Northern Electric Co., 1923–1926; Bell Telephone Laboratories, 1926—. Prior to World War II he developed equipment for step-by-step central offices, community dial offices, private branch exchanges, and station systems. Through the war, he supervised the equipment development of a variety of radio and radar projects. From 1945 to 1958 he was responsible for the equipment development of the No. 5 crossbar, and since that time he has made equipment studies for No. 1 ESS. Life member, IEEE.

LAIMONS FREIMANIS, Technische Hochschule, Munich; B.S.E.E., 1951, M.S.E.E., 1952, Michigan State University; Bell Telephone Laboratories, 1952—. He first worked on development of signaling systems for Civil Aeronautics Administration networks. Since completing the communications development training program he has been engaged in exploratory and development work on electronic switching systems. Member, Tau Beta Pi, Eta Kappa Nu and Sigma Pi Sigma.

DOLIVE L. FUNK, B.E.E., 1956 and M.S.E. (E.E.), 1957, University of Florida; Bell Telephone Laboratories, 1957—. He prepared the switching network controller automatic maintenance for the ESS trials at Morris, Ill. On No. 1 ESS he worked on the duplication, switching, and maintenance of the central pulse distributor and was also responsible for designing the automatic maintenance of trunk and service circuits. He is currently planning the operational testing of No. 1 ESS. Member, IEEE.

LEE E. GALLAHER, B.S.E.E., 1951, M.S.E.E., 1956, Case Institute of Technology; Bell Telephone Laboratories, 1955—. He first worked on the design of the flying-spot store for the Morris experimental central

office, and later worked on the design of the program stores for No. 1 ESS. Member, Sigma Xi and Tau Beta Pi.

RICHARD M. GENKE, B.S.E.E., 1954, University of Wisconsin; M.S.E.E., 1961, Columbia University; Bell Telephone Laboratories, 1954—. Since 1954 Mr. Genke has been engaged in memory system development for electronic switching systems, including the barrier grid store for the Morris electronic central office and the No. 1 ESS call store. Member, Tau Beta Pi and Eta Kappa Nu.

H. GHIRON, B.S., 1952, M.S., 1954, Massachusetts Institute of Technology; Bell Telephone Laboratories, 1955—. He has been engaged in systems development work, first for the Morris ESS experiment and currently for the No. 1 ESS. Member, A.C.M., IEEE and Sigma Xi.

L. F. GOELLER, JR., B.E.E., 1953, M.E.E., 1954, University of Virginia; Bell Telephone Laboratories, 1954—. Since coming to Bell Laboratories he has worked on circuits for electronic switching systems. Member, IEEE.

T. SPENCER GREENWOOD, B.S.E.E, 1951, Northeastern University; M.S.E.E, 1953, Massachusetts Institute of Technology; Bell Telephone Laboratories, 1953—. Since starting at the Laboratories, Mr. Greenwood has been associated with the development of memory systems for electronic switching. These included the barrier grid store and flying-spot store used in the Morris experimental electronic switching system, and the permanent magnet twistor store for No. 1 ESS. Member, Eta Kappa Nu, Tau Beta Pi, Sigma Xi and IEEE.

W. E. GRUTZNER, B.S.E.E, 1935, Cooper Union; Bell Telephone Laboratories, 1925—. He first worked for the Laboratories as an office boy, draftsman, and instructor. Since 1935 he has worked in various equipment development organizations, including trial installations; panel, step-by-step, and toll analyzation; radar development; and No. 1 and No. 5 crossbar. He also worked in the systems engineering department, where he supervised a group responsible for the establishment of requirements for audio recording and announcement systems. At present, he is in charge of a group responsible for the equipment development of No. 1 ESS switching networks and associated peripheral equipment.

ANTHONY M. GUERCIO, B.S.E.E., 1955, Mississippi State University;

M.S.E.E., 1960, New York University; Bell Telephone Laboratories, 1955—. He was initially associated with a group developing the flying-spot store. After completing the communications development training program, he worked on scanners until recently. He is now a member of a common systems circuit group. Member, Tau Beta Pi and Phi Kappa Phi.

PHILIP A. HARDING, B.E.E., 1954, Cooper Union; M.S.E.E., 1960, Columbia University; Bell Telephone Laboratories, 1954—. He has been primarily concerned with the development of No. 1 ESS call stores. He presently supervises a group concerned with development of memories, design of logic circuits, and evaluation of semiconductor devices for application in No. 1 ESS and the DAC systems. Member, Tau Beta Pi.

J. A. HARR, B.S., 1940, Roanoke College; M.S. (Applied Science), 1951, Harvard University; Bell Telephone Laboratories, 1955—. Mr. Harr worked in the Harvard Computation Center from 1947 to 1955. His work involved both design and programming of computers developed by the staff of the computation center. Since 1955 he has been engaged in both the design of central control equipment and programming of the electronic switching system. He is now Head, Electronic Switching Programs Department. Member, Harvard Engineering Society.

G. HAUGK, B.S.E.E, 1952, Newark College of Engineering; New York University; Western Electric Company, 1947–1948; Bell Telephone Laboratories, 1952—. After completing the communications development training program course, Mr. Haugk worked on design of circuits for electronic switching systems for three years. Following this he supervised a group responsible for field trial and testing of electronic switching systems. He presently heads a department responsible for No. 1 ESS evaluation, personnel training, and information activities. Member, IEEE.

W. S. HAYWARD, JR., A.B., 1943, S.M., 1947, Harvard University; Bell Telephone Laboratories, 1947—. He was first engaged in traffic studies of telephone switching systems. Since 1958 he has been working on the systems engineering of the No. 1 ESS. Member, IEEE, Operations Research Society of America, Association for Computing Machinery and Society of Harvard Engineers and Scientists.

H. ROBERT HOFMANN, B.E.E., 1957, University of Florida; M.E.E.,

1962, New York University; Bell Telephone Laboratories, 1957—. Mr. Hofmann first worked on the experimental ESS remote line concentrator, followed by work on the ferreed switch. Later he was involved in the design of the ferrod used in the No. 1 ESS scanner. More recently he was concerned with circuits for the No. 1 ESS switching network control. At present, he is working on the addition of centrex to No. 1 ESS. Member, Sigma Tau.

ERNA S. HOOVER, B. A., 1948, Wellesley College; Ph.D., 1951, Yale University; Bell Telephone Laboratories, 1954—. Mrs. Hoover has been engaged in systems engineering studies, including the formulation and evaluation of the system plans for the No. 1 ESS data processor and its program. Member, Association for Computing Machinery, Association for Symbolic Logic, American Philosophical Association and Phi Beta Kappa.

LUTHER W. HUSSEY, A. B., 1923, Dartmouth College; A.M., 1924, Harvard University; B.S.E.E., 1930, Union College; Bell Telephone Laboratories, 1930—. Mr. Hussey's major activities have been in the development and application of semiconductor devices, starting with studies of copper-oxide and similar devices for modulators in carrier systems, and magnetic materials such as permalloy for amplifiers and pulse generators. He was active in the early development of transistors and in exploratory work on their applications. At present he is supervising a group working in the control area of No. 1 ESS. Senior member, IEEE.

WILLIAM KEISTER, B.S.E.E., 1930, Alabama Polytechnic Institute; Bell Telephone Laboratories, 1934—. His early work with Bell Laboratories was on switching and signaling systems. During World War II, he instructed Army and Navy personnel on the operation and maintenance of fire-control radar equipment. He organized and taught courses on switching circuit design to Laboratories personnel and is co-author with A. E. Ritchie and S. H. Washburn of the book *The Design of Switching Circuits*. In 1958 he was appointed Director, Electronic Switching Systems Engineering Center, with responsibility for the engineering planning of electronic telephone switching systems. Member, IEEE, Eta Kappa Nu, Tau Beta Pi and Phi Kappa Phi.

RAYMOND W. KETCHLEDGE, B.S. and M.S., 1942, M.I.T.; Bell Telephone Laboratories, 1942—. Until 1946 he was associated with military

developments in the fields of infrared detection and underwater sound. During the next six years, Mr. Ketchledge participated in the development of a submarine cable system and a broadband coaxial carrier system. In 1953 he became responsible for gas tube and storage tube development as an electron tube development engineer. The next year he was appointed a switching system development engineer, responsible for electronic memories and switching networks for electronic switching systems. Mr. Ketchledge was made Assistant Director of Switching Systems Development in 1956. He became Director, Electronic Switching Laboratory at Holmdel in 1959. He has had 51 U.S. patents issued and 5 are pending. Member, Sigma Xi; Senior Member, IEEE.

D. C. Koehler, B.S.E., 1941, University of Illinois; M.S.E.E., 1950, Stevens Institute of Technology; Bell Telephone Laboratories, 1941—. He first participated in the mechanical design of the M9 gun data computer, various bombsight and antiaircraft radar equipment, the M13 depth-charge mechanism, and switching relays. Since 1954 he has supervised the mechanical design of the memory systems for both the Morris trial and No. 1 ESS. These systems have included the barrier grid store, the flying-spot store, the ferrite sheet call store, the twistor program store, and related equipment for handling and writing the twistor memory cards. He is also responsible for the No. 1 ESS frameworks, cable racks and end guards. Member, Tau Beta Pi and Phi Kappa Phi.

N. A. Martellotto, B.E.E., B.S. (Applied Mathematics), 1957, Georgia Institute of Technology; M.S.E.E., New York University, 1959; Bell Telephone Laboratories, 1957—. Mr. Martellotto was initially associated with the Bell System data processing project, where he did some preliminary logic design work and also contributed to a machine aids to design program system. Using a general-purpose computer as a tool, he later did exploratory work on the analysis of logic circuits and also completed an interpretive simulation program for the No. 101 electronic switching system. He currently supervises the compiler program group in No. 1 ESS with the responsibility for developing several utility system programs. Member, IEEE, Tau Beta Pi and Eta Kappa Nu.

Harold F. May, B.S.E.E., Cooper Union, 1927; graduate studies at New York University and Polytechnic Institute of Brooklyn; P.E., New York State; New York Telephone Company, 1922–1929; Teleregister Corporation, 1929–1954; Bell Telephone Laboratories, 1954—. He has been associated with special customer service developments,

stock quotation services, air traffic control, military digital and analog computer simulators, magnetic drum applications to crossbar systems, and electronic switching systems. At present, he is supervising a group which is concerned with peripheral sensing and control functions and transmission in No. 1 ESS. Member, IEEE, Communications Switching Committee and Communications Division.

ROBERT S. MENNE, B.E.E., 1953, Clarkson College of Technology; M.S.E.E., 1959, Newark College of Engineering; Bell Telephone Laboratories, 1953—. After completing the communications development training program, he worked on transmission system terminal equipment. Since 1958 he has been engaged in the logic design of circuits for the No. 1 ESS central control. Member, Eta Kappa Nu.

J. S. NOWAK, Bell Telephone Laboratories, 1955—. Mr. Nowak has worked in the system planning area on the Morris experimental electronic telephone central office. He is currently concerned with the system planning and maintenance requirements for No. 1 ESS.

H. OEHRING, B.S. (Mathematics), Lafayette College, 1958; M.S. (Mathematics), Ohio University, 1960; Bell Telephone Laboratories, 1960—. He has assisted in the assembler and macro language design of PROCESS III, written utility routines for computers, and engaged in military systems probability studies. Currently, he is engaged in real-time programming and debugging of the No. 1 ESS stored program. Member, Sigma Pi Sigma.

MARVIN C. PAULL, B.E.E., 1952, Clarkson College of Technology; Bell Telephone Laboratories, 1953—. At Bell Laboratories his work has included studies in the use of magnetic cores in computer design, research and development in the logical design of computer circuitry, and computer programs. Member, IEEE, Eta Kappa Nu and Tau Beta Pi.

HELMO RAAG, B.S.E.E., 1953, Oklahoma State University; M.S.E.E., 1959, New York University; Bell Telephone Laboratories, 1953—. He completed the communications development training program in 1956. He was first engaged in the development of the flying-spot store and permanent magnet twistor memory systems for electronic switching systems, and later supervised a group concerned with the development of the ferreed crosspoint. Since 1962 he has been supervisor of the group developing the master control center for the No. 1 ESS project. Member, Eta Kappa Nu, Phi Kappa Phi and Pi Mu Epsilon.

PHILIP G. RIDINGER, B.S.E.E., 1950, Lehigh University; M.S.E.E., 1960, Stevens Institute of Technology; Bell Telephone Laboratories, 1950—. He has been engaged in the exploratory development of remote line concentrators, ferreed crosspoints, and space division switching networks. More recently he worked on the design of the master control center for No. 1 ESS, and is presently in charge of a group responsible for the design of common systems circuits and the remote switching unit. Member, Eta Kappa Nu and Tau Beta Pi.

ROBERT S. SKINNER, B.S.E.E., 1939, University of Kansas; Southwestern Bell Telephone Company, 1939–1942; Bell Telephone Laboratories, 1942—. Mr. Skinner has been engaged in the development of military radar systems, the Nike-Ajax system, and the civil emergency reporting system, as well as a variety of telephone switching systems. He presently supervises a group responsible for control equipment design for electronic switching systems. Senior member, IEEE.

MICHAEL T. SKUBIAK, B.S.M.E., 1959, University of Akron; M.S.E.M., 1963, Ohio State University; Bell Telephone Laboratories, 1959—. He has been engaged in the development of the printed wire board connector, four-wire ferreed switch, and the distributing frames for No. 1 ESS. He currently is a supervisor of an equipment design group for UNICOM. Member, Sigma Tau, Pi Mu Epsilon and Sigma Xi.

ROBERT B. SMITH, A.B., 1950, M.A., 1951, Ph.D., 1957, Yale University; Bell Telephone Laboratories, 1956—. Mr. Smith has been engaged in programming development for electronic switching systems. He is at present in charge of a group preparing call processing programs for No. 1 ESS. Member, Association for Computing Machinery, American Mathematical Society and Phi Beta Kappa.

ROBERT E. STAEHLER, B.S.E.E., 1947, The College of the City of New York; M.S.E.E., 1948; further graduate studies, Polytechnic Institute of Brooklyn; Bell Telephone Laboratories, 1948—. His early work was on local signaling systems, voice-frequency toll signaling systems, and early Nike missile trainer studies. Since 1953 he has been engaged in development work on electronic switching systems. He headed the department responsible for the development of the memory systems for the experimental field trial of the electronic central office in Morris, Illinois, and the central control and temporary memory circuits for No. 1 ESS. Currently, as Director, Electronic Switching Projects Labora-

tory, he has responsibility for the development of electronic voice, message, and data switching projects. Member, IEEE, Eta Kappa Nu, Tau Beta Pi and Sigma Xi.

A. A. STOCKERT, B.S.E.E., 1957, University of Dayton; M.S.E.E., 1959, New York University; Bell Laboratories, 1957—. He was first engaged in the design of aircraft control data links. He is currently engaged in maintenance program planning for No. 1 ESS.

FRANK F. TAYLOR, B.S.E.E., 1955, University of Kentucky; Bell Telephone Laboratories, 1955—. His work at Bell Laboratories has included circuit and logic design in the development of electronic telephone switching systems. He has also taught courses in electronic switching as part of the Laboratories sponsored operating engineers training program. Member, Eta Kappa Nu and Tau Beta Pi.

S. H. TSIANG, B.S., 1947, University of Nanking; M.S., 1949, Carnegie Institute of Technology; Union Switch and Signal, 1949–1956; Bell Telephone Laboratories, 1956—. Mr. Tsiang has worked on maintenance and administration circuits and requirements for an experimental electronic central office. Currently he is engaged in test planning, programming and field testing for the No. 1 electronic switching system.

LEE S. TUOMENOKSA, B.S., 1952, Worcester Polytechnic Institute; S.M., 1954, Massachusetts Institute of Technology; Bell Telephone Laboratories, 1954—. Mr. Tuomenoksa first worked on operational program design for the Morris experimental switching system. Since then he has supervised a group responsible for No. 1 ESS maintenance planning and programming. Member, IEEE, ACM, Tau Beta Pi, Sigma Xi and Eta Kappa Nu.

W. ULRICH, B.S., 1952, M.S., 1953, Dr. Eng. Science, 1957, Columbia University School of Engineering; Bell Telephone Laboratories, 1953—. Mr. Ulrich was first engaged in the design of automatic maintenance circuits and programs for the experimental electronic central office installed in Morris, Illinois. In 1959, he started working on the No. 1 ESS; on this project, he worked on over-all systems coordination, central control organization, control of the network, translations, and the equipment numbering and addressing plan. Since February, 1964, he has headed a department developing programs for UNICOM, a military communications system.

M. Dean Underwood, B.S., 1952, Pennsylvania State University; Bell Telephone Laboratories, 1952—. Mr. Underwood's early work was concerned with improving the properties of point contact diodes. Since graduating from the communications development training program, he has been engaged in the design of semiconductor circuits for applications in electronic switching systems.

H. Earle Vaughan, B.S. (in C.E.), 1933, Cooper Union; Bell Telephone Laboratories, 1928—. Mr. Vaughan joined the transmission research department in 1928. He was first engaged in work on voice-operated devices and studies of the effects of speech and noise on voice frequency signaling systems. During World War II he worked on anti-aircraft computers, fire-control radar, and other military projects. He was later concerned with digital techniques and systems, data transmission, and electronic switching systems. He holds many patents and has written several papers in technical journals. He was appointed switching systems research engineer in 1955, and Director of Systems Research in 1958, in charge of research on television, mobile radio, switching and integrated communication systems. He was appointed Director, Electronic Switching System Center in 1962, with responsibility for system design and programming of a new electronic switching system. Fellow, IEEE.

Hildegard M. Vellenzer, M.S. (mathematics), 1939, Johann Wolfgang Goethe University, Frankfurt; Bell Telephone Laboratories, 1957—. Her entire career at Bell Laboratories has been in the area of electronic switching.

D. H. Wetherell, B.S.E., 1923, Lafayette College; Western Electric Company, 1923; Bell Telephone Laboratories, 1925—. Mr. Wetherell worked on development of equipment for all types of telephone switching systems until World War II, when he was supervisor of a group responsible for the design of airborne radar systems for the armed forces. After the war, he supervised a group working on the development of equipment for toll telephone switching systems and later headed a group responsible for the development of circuits and equipment for nationwide dialing. At present, he is in charge of a department engaged in the design and development of equipment for No. 1 ESS.

L. Zimmerman, B.E.E., 1952, Brooklyn Polytechnic Institute; Bell

Telephone Laboratories, 1952—. After completing the communications development training program course, Mr. Zimmerman worked in a telephone switching exploratory development group. Since 1956 he has been engaged in various aspects of testing electronic switching systems. Member, Eta Kappa Nu, Tau Beta Pi and Sigma Xi.