# THE BELL SYSTEM
# TECHNICAL JOURNAL

DEVOTED TO THE SCIENTIFIC AND ENGINEERING
ASPECTS OF ELECTRICAL COMMUNICATION

Volume 48 October 1969* Number 8





System Organization and Objectives
**A. E. Spencer and F. S. Vigilante** 2607

Control Unit System **T. E. Browne, T. M. Quinn, W. N. Toy and J. E. Yates** 2619

Peripheral System **J. Digrindakis, L. Freimanis, H. R. Hofmann and R. G. Taylor** 2669

Service Features and Call Processing Plan
**R. J. Andrews, J. J. Driscoll, J. A. Herndon, P. C. Richards and L. R. Roberts** 2713

Administration and Maintenance Plan
**H. J. Beuscher, G. E. Fessler, D. W. Huffman, P. J. Kennedy and E. Nussbaum** 2765

Apparatus and Equipment
**C. W. Lonnquist, J. C. Manganello, R. S. Skinner, M. T. Skubiak and D. J. Wadsworth** 2817

Service Programs **M. E. Barton, N. M. Haller and G. W. Ricker** 2865

Contributors to This Issue 2897

Index 2905


* *This issue of the Bell System Technical Journal was not published until after October 22, 1969.*

# System Organization and Objectives

By A. E. SPENCER and F. S. VIGILANTE

(Manuscript received February 18, 1969)

*This paper describes the system organization and objectives of the No. 2 Electronic Switching System. The place of No. 2 ESS in the Bell System and the design approach used are outlined and traffic capacity estimates are discussed. This paper also serves as an introduction to the detailed set of papers which follow.*

## I. INTRODUCTION

### 1.1. *General*

The No. 1 Electronic Switching System described by Keister, Ketchledge, and Vaughan in the September 1964 issue of the B.S.T.J. has been in commercial service since May of 1965, when the first installation was activated in Succasunna, New Jersey. Since that time, many additional installations have been made and electronic switching has been clearly established as a valuable switching medium for use in the Bell System.

The No. 1 ESS was designed for use in metropolitan areas where large numbers of lines with heavy traffic are served. Although a few installations of No. 1 ESS have been made in nonmetropolitan areas, the need within the Bell System for another system that would be economically attractive in the size range from 1000 lines to about 10,000 lines has been evident. This need will be met by the No. 2 ESS.

Broadly, then, the objectives established for the No. 2 ESS were to complement the No. 1 ESS by offering economical electronic switching service for nonmetropolitan offices in the medium size range. These broad objectives have more specific implications on customer services, maintenance and administrative features, traffic capacity, floor space, and, of course, cost.

## II. OBJECTIVES

### 2.1. *Customer Service Feature Objectives*

By and large, the service features offered on all local switching systems are and should be the same. However, there are some small differences that result from differences in the intended field of application. Just as offices for use in nonmetropolitan areas tend to be smaller than those used in metropolitan areas, the service features that are required tend to differ. For instance, four-party and eight-party lines are fast disappearing from large cities but are still required in some suburban and rural areas.

Since No. 2 ESS is intended for application in nonmetropolitan areas, the objective for the initial package of customer service features is to provide features appropriate for such areas. It is a further objective to provide the features in the form of a "basic group" that would be present in all offices plus additional groups of features that would be optional. Specific features to be provided in the initial installations of No. 2 ESS are covered in a companion paper.[1]

### 2.2 *Maintenance and Administrative Objectives*

A high degree of dependability, maintainability and operational efficiency are required in all local switching systems. Experience indicates that electronic switching systems can meet these requirements at annual costs considerably less than with conventional switching machines.

Because of its intended use in nonmetropolitan areas, the No. 2 ESS was designed to be largely unattended. That is, wherever economically practical, maintenance tests, translation changes, traffic and plant measurements, and so on, were designed to be controlled remotely. Of course, repair or replacement of circuit packs, connections to cables, and so on, all require work in the office.

The companion article by Beuscher and others covers both hardware and software aspects of the No. 2 ESS administration and maintenance plan.[2] Of particular interest in this article is the system trouble example which describes the sequence of actions from a particular trouble detection through system repair.

### 2.3 *Flexibility*

More and more it is becoming apparent that flexibility is an extremely important objective for any new switching system. Many services being provided today had not been conceived even 20 years

ago when the No. 5 crossbar system was designed, let alone at the time the step-by-step and panel switching systems were invented. Twenty to forty years from now, the electronic switching systems of today are likely to be called on to provide customer services and operating features not known today. For this reason, every attempt was made to make No. 2 ESS a flexible system.

The use of a stored program common control is, of course, a major step in providing the desired flexibility. In addition, hardware flexibility is provided by extensive use of modular equipment, connective frames and simplified peripheral communications.

### 2.4 *Traffic Capacity*

For an office of about 10,000 lines in a nonmetropolitan area, most Bell System requirements can be met with a minimum capacity of 16,000 busy hour calls and 40,000 ccs (hundred-call-seconds). However, it is clear that any capacity greater than these minima that can be achieved at reasonable cost would be valuable in the future for new services or increased calling rates. We discuss busy hour call capacity estimates for No. 2 ESS in Section V.

### 2.5 *Economic Objectives*

It is easy to say that the cost of a system to meet the preceding objectives should be as low as possible, and this, of course, is true. But what might be a realistic objective to shoot for? Since the No. 5 crossbar system is currently installed in the Bell System in this field of application, it was felt that a reasonable objective for No. 2 ESS, with its many attractive features, was to be competitive with standard No. 5 crossbar systems in terms of installed first cost. Thus the advantages of the No. 2 ESS in terms of simplified installation (relatively sophisticated factory test, shipment of several equipment frames as a factory wired unit and extensive use of plugs and jacks for interunit wiring) would be recognized in this economic comparison but other advantages, such as new service features, reduced maintenance, and administrative costs would not. This enables the using telephone company to benefit from its new equipment immediately upon installation and does not require it to anticipate future savings in order to prove in modern telephone equipment.

### 2.6 *Other Objectives*

Obviously many other objectives were also established for No. 2 ESS. For example, it should provide service as reliable as, or better

than, existing systems; it should occupy minimum floor space, and it should have as short an installation interval and as small an installation effort as possible.

## III. SYSTEM DESCRIPTION

### 3.1 *General Design Plan*

The design of No. 2 ESS was derived from experience with both the No. 1 ESS and No. 101 ESS. The system uses a ferreed network and other peripheral units similar to that of No. 1 ESS. The No. 2 ESS control is a new design based on the smaller type of control which is being used in the No. 101 Electronic Switching System.

The rationale behind these design decisions is clear. The No. 1 ESS serves large offices and creates a high production of network and peripheral apparatus. The designer of the small system gains the benefit of this large production by using the same apparatus. In the control area the small office designer has greater flexibility since he creates his own high production based on the large number of entities in the small office field. He, in addition, has the advantage of moderate traffic capacity requirements. It is, therefore, reasonable to have a specialized design for the small office control which attempts to minimize costs at some sacrifice of traffic capacity.

### 3.2 *Control Unit*

Figure 1 is a block diagram of the No. 2 Electronic Switching System. Although reminiscent of most electronic switching system block diagrams, there are several points which can be made in conjunction with this figure. The control consists of duplicated central controls, each working with a common maintenance center. Each central control has been designed as a single switchable entity. That is, the frames which contain the program and translation store, the call store, the program control and the input-output are treated as one entity rather than individually switchable units. This reduces both the quantity of equipment required and the quantity of program required for the administration of redundancy.

The processor frame used in No. 2 ESS contains the program control (instruction processing logic), a semiautonomous input-output section, capacity for 16,384 16-bit words of call store (temporary memory) and capacity for 512 bipolar central pulse distributor points. The call store may be equipped in steps of 4096 words while the cen-

tral pulse distributor may be equipped, as required, with packages of eight central pulse distributor points. High speed transistor resistor logic is used for all logic applications in this frame. This logic uses thermocompression bonded beam leaded silicon devices with thin film resistors on an alumina substrate. Growth frames are available which allow expansion of the call store to 32,768 words and the central pulse distributor to 16,384 points.

The basic program and translation store frame for No. 2 ESS contains four permanent magnet twistor modules with a total capacity of 65,536 22-bit words. In addition to the basic frame, growth frames are available which allow this store to be expanded in steps of 16,384 words to a maximum size of 262,144 words.

The semiautonomous input-output unit which is included in each processor frame shares the call store with the program control on what is generally a lower priority. The input-output unit works together with the program to: (*i*) scan for line originations, (*ii*) collect

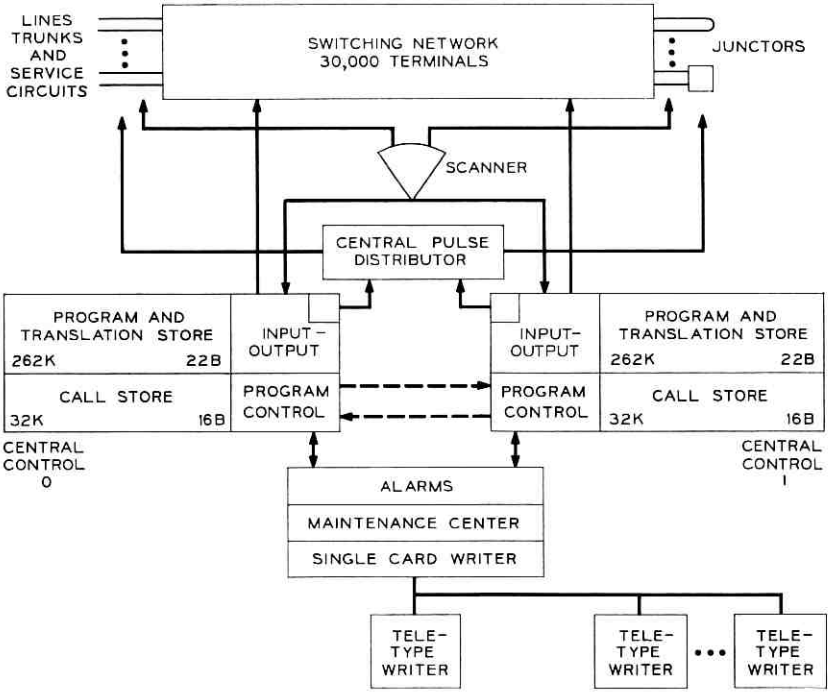



Fig. 1 — No. 2 ESS Block Diagram.

dialed digits and tone signals, (*iii*) outpulse, and (*iv*) transmit data. Use of the input-output unit for digit receiving and sending functions has allowed the normal program interrupt cycle to be set at 25 milliseconds rather than the usually required 5 to 10 milliseconds. In addition, the precise timing and short scan intervals available through the input-output unit has allowed simplification in circuits such as incoming trunks from step-by-step offices. The input-output unit is fully discussed in the Control Unit paper.[3]

### 3.3 *The Network*

While generally using the same apparatus as No. 1 ESS, the No. 2 ESS network has been designed to minimize "getting started" costs. This was easier to achieve in No. 2 ESS since the ultimate size of the system will be considerably smaller than the ultimate size of No. 1 ESS. The system uses a two-wire folded four-stage ferreed network with lines, trunks, and service circuits all appearing on the same side. It is therefore not necessary to have a separate line and trunk link network in small offices. Only two types of frames are used to make up the No. 2 ESS network:

(*i*) The network control junctor switch frame contains eight 64 × 64 octile grids which form the third and fourth stages of the network. Each grid is made up of sixteen 8 × 8 ferreed switches arranged for 64 inputs and 64 outputs. This frame contains a pair of controllers which operate the junctor network and the network and scanners of the line-trunk switch frames.

(*ii*) The line-trunk switch frame can accommodate 512 inputs which concentrate to 256 outputs. This frame is made up of sixteen 32 × 16 ferreed concentrators. These concentrators are made up of eight 4 × 4 switches in the first stage and four 8 × 4 switches in the second stage. This frame also contains a 512 point ferrod line scanner. One, two, three or four line-trunk switch frames can be connected through B-links to the 512 inputs of the network control junctor switch frame to form a complete line-trunk link network.

At 4:1 concentration ratio, each line-trunk link network can provide up to 2048 terminals for lines, trunks and service circuits. At 2:1 concentration ratio, 1024 terminals would be provided per line-trunk link network. Through use of circuit and wire junctors additional networks can be added with an ultimate capacity of fifteen networks which can carry over 100,000 ccs of traffic.

### 3.4 *Trunks, Junctors and Service Circuits*

The trunks, junctors, and service circuit designs of No. 2 ESS are generally the same basic types as those in No. 1 ESS. The primary difference has been in the use of a special integrated circuit package which has been designed specifically to control relays. This package, called the peripheral decoder, offers several advantages over previous trunk control schemes. Each peripheral decoder pack can operate and hold up to 12 relays and would typically be used to control four universal trunks or circuit junctors. Information is sent from a central pulse distributor point serially to the peripheral decoder where the information is stored on flip-flops and decoded to operate the desired relay. From the program point of view, the relays are thus controlled at electronic speeds. Use of the peripheral decoder also allows the trunk control to be equipped with a single pack at a time as trunks are added to the office. In addition, the peripheral decoder is a good application for integrated circuits since it will be a high production circuit package. The peripheral decoder is described in detail in the companion articles on the Peripheral System, and Apparatus and Equipment.[4,5]

### 3.5 *Installation and Growth*

A significant effort has been made in No. 2 ESS to simplify growth and change of equipment and to reduce installation intervals. Most of the connections between the central control frames and the connections to the maintenance center are through plug-in connectors. In addition, both the call store and program store growth units are equipped with plugs and jacks. It is planned that the entire control complex will be plugged together at the factory and tested as a system through X-ray programs. The same X-ray programs will be used when installing the control complex at the central office site.

Some of the maintenance center functions also include capabilities which aid in both factory test and installation of the machine. These functions provide (*i*) manual access to internal processor states, registers, and memory locations; (*ii*) control capability over execution of program segments, and (*iii*) emergency override features which put the machine under manual control.

To aid in peripheral system installation, bus communication between the control and the peripheral frames and all B-link connections within the network are on plug-in connectors. In addition, trunks and service circuits are equipped on easy-to-install mounting plates

while peripheral decoders, as already mentioned, are on plug-in packages.

## IV. PROGRAMMING

### 4.1 *Program Size*

Basic program and translation requirements for small (single line-trunk link network) No. 2 ESS offices are estimated at less than 65,000 22-bit words. Thus these offices may be served by a single four module program store per central control. As the offices grow in size and feature requirements, additional program modules may be added as already described.

The instruction format used in No. 2 ESS is the same as that used in No. 101 ESS and consists of two types. Type 1 uses a 22-bit word which contains a 5-bit operation code and a 16-bit address plus a check bit. Type 2 contains two complete 10-bit instructions, each of which contains a 5-bit address and a 5-bit operation code. The short 5-bit address is made effective by extensive use of relative and implicit addressing of both program and call store information. Experience on No. 101 ESS and No. 2 ESS has shown that this format is highly efficient in the use of program store space thereby aiding in meeting the low cost objectives for the system.

### 4.2 *Call Programs*

We leave the details of the systems programming to later papers.[1,2,6] However, with the emphasis on low system cost, a few points are worth reviewing. We have already mentioned the highly efficient micro-type orders used for program instructions. The major feature of these kinds of instructions is that very few bits are wasted in unused options and unneeded address bits, as is often the case with larger, fixed length program instruction words.

Also simplifying both program and administration is the use of general purpose registers within the temporary memory for all types of calls. These registers consist of a progress mark which defines the state of the call plus other information pertinent to the call. Call registers containing progress marks generally use the same format and the same register size, thereby allowing common programs to be used for many different call states.

The progress marks themselves define the first address of the

program required to process the particular call. The preponderance of call processing can then be described as a series of transfers to the individual progress mark programs. In addition to straightforward call processing techniques, great emphasis has been given to the use of common subroutines. Many progress mark programs consist entirely of calls to nested subroutines. In order to help in this effort some special orders and push-down list features have been included in the basic instruction repertoire to aid in the writing and to facilitate the use of subroutines. The Control Unit System article further describes the No. 2 ESS order repertoire.[3]

### 4.3 *Maintenance and Administrative Programs*

Maintenance and administrative programs constitute over half of the programming job, hence considerable effort has been extended to seek simplifications in this area. In the control system the maintenance structure is based on a single interrupt level, a single match circuit, a few internal check circuits and good program access between the on- and off-line units. These factors, in conjunction with low component counts and the single switchable entity concept, all tend to reduce and simplify the programs. In the periphery, the small number of frame types and options has reduced the adminstrative and growth program needs as well as creating some maintenance simplification.

## V. TRAFFIC CAPACITY

In designing a small office, many trades are made which favor low cost over real-time traffic capacity. Care must be taken, however, to leave sufficient capacity to allow a reasonable range for newly started offices and growth. Current estimates indicate that the No. 2 ESS may be engineered for up to 19,000 busy hour calls with a traffic mix of one-third intraoffice, one-third outgoing, and one-third incoming calls. This estimate has been derived from several sources. Among these are: (*i*) cycle counts in the programs, (*ii*) simulation of the programs, (*iii*) laboratory measurements as the programs are being debugged, and (*iv*) comparisons with existing systems.

The busy hour call calculation has been made by determining the number of calls that may be carried with the processor occupied at 95 percent (peak capacity) and reducing this result by an allowance for false starts (number of false starts assumed equal to 30 percent

of originating calls). The total is further reduced to allow a 30 percent peak with the final result being 19,000 busy hour calls for engineered load. In general, all calculations have been very conservative.

### 5.2 *Dual Office*

While it is generally expected that the capacity of 19,000 busy hour calls will be quite adequate for the No. 2 ESS field of application, techniques are being explored for expansion beyond this capacity. These techniques are aimed at the situation where very large unanticipated growth is necessary at a No. 2 ESS installation. Figure 2 is a block diagram for a dual No. 2 ESS office. This diagram shows a pair of No. 2 ESS offices with the networks joined by junctors. In addition, data links would be provided for communications between the controls of each suboffice. With this plan, all connections within a suboffice or between suboffices would use only eight stages of network. In addition to this, trunk groups would be distributed between the suboffices in the same way as they would be for a single office. In order to facilitate the use of common trunk groups and a common network, the data links would be used to exchange network status information, trunk status information, and supervision where necessary.

The dual office concept represents a form of multiprocessing highly suited to telephone applications. In general, each suboffice does business as usual except where a connection is to be completed between the suboffices. The number of inter suboffice connections can be minimized by not permitting service circuit connections between suboffices. For example, when a line originates, the dialing connection will always be made within the suboffice on which the line homes. When a line is called, the ringing connection will always be made within the called suboffice. By proceeding in this way the real time consumed in completing inter suboffice connections is limited primarily to the final talking path.

Except for network status information, the data communication between the offices is simple and does not add a major amount of new work to the normal single office program. Early traffic estimates on dual processing indicate that inter suboffice calls will incur an approximate 20 percent real time penalty while the penalty on intra suboffice calls appears negligible. The emphasis in the dual office is clearly to retain network and trunk efficiency since this is where, in a large telephone office, most of the expense occurs. On the control

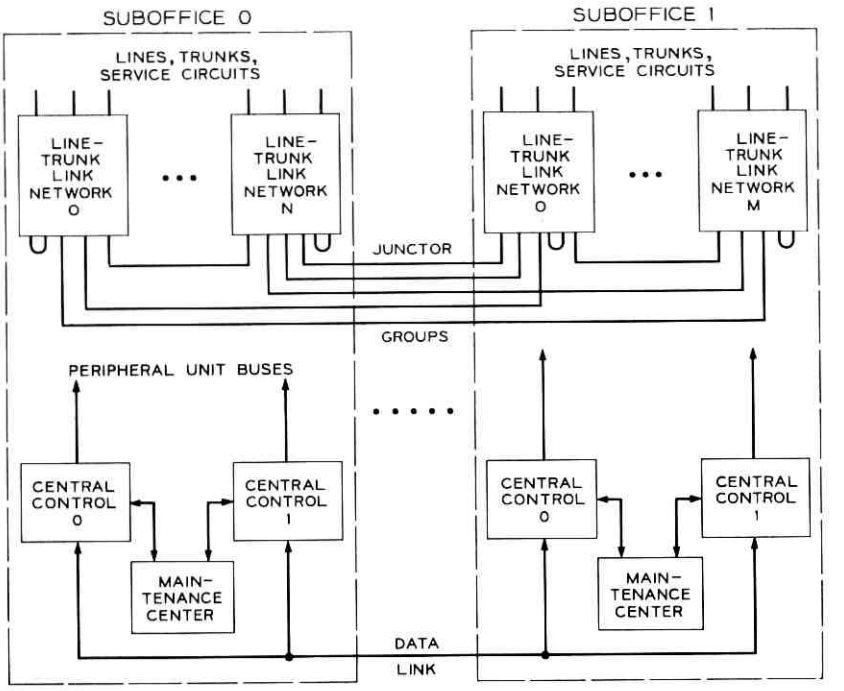



Fig. 2 — No. 2 ESS Dual Office.

side the emphasis is on relatively simple growth and retention of a large part of the single office programs. The dual office version of No. 2 ESS is not aimed at the new office field; the primary objective is to allow a greater growth range for the system while not penalizing the "get started cost" of a single office.

## VI. STATUS

The first No. 2 ESS office is scheduled to go into service during the last quarter of 1970 at Oswego, Illinois. Hardware and software are being debugged on two test models at Bell Telephone Laboratories, Indian Hill, Naperville, Illinois. The 2100-line Oswego office is expected to be followed in service by three other offices in 1971, ranging from approximately 1000 to 5000 lines, and thus representative of the typical No. 2 ESS new starts. Western Electric Company price estimates indicate that No. 2 ESS is meeting its economic objectives and wide penetration in the Bell System central office field is expected.

VII. ACKNOWLEDGMENTS

The design of No. 2 ESS contains the contributions of many people at the American Telephone and Telegraph Company, Western Electric Company, and Bell Telephone Laboratories. In addition, the experiences of the many Bell System operating companies using electronic switching systems has provided invaluable information for the design process.

REFERENCES

1. Andrews, R. J., Driscoll, J. J., Herndon, J. A., Richards, P. C., and Roberts, L. R., "Service Features and Call Processing Plan," B.S.T.J., this issue, pp. 2713–2764.
2. Beuscher, H. J., Fessler, G. E., Huffman, D. W., Kennedy, P. J., and Nussbaum, E., "Administration and Maintenance Plan," B.S.T.J., this issue, pp. 2765–2815.
3. Browne, T. E., Quinn, T. M., Toy, W. N., and Yates, J. E., "Control Unit System," B.S.T.J., this issue, pp. 2619–2668.
4. Digrindakis, J., Freimanis, L., Hofmann, H. R., and Taylor, R. G., "Peripheral System," B.S.T.J., this issue, pp. 2669–2712.
5. Lonnquist, C. W., Manganello, J. C., Skinner, R. S., Skubiak, M. T., and Wadsworth, D. J., "Apparatus and Equipment," B.S.T.J., this issue, pp. 2817–2863.
6. Barton, M. E., Haller, N. M., and Ricker, G. W., "Service Programs," B.S.T.J., this issue, pp. 2865–2894.

# Control Unit System

By THOMAS E. BROWNE, THOMAS M. QUINN, WING N. TOY and JOHN E. YATES

(Manuscript received March 7, 1969)

*The control unit system consists of the combination of elements which direct and control the operation and functions of the system. It includes the stores and processing units as well as the elements of the man-machine interface. This paper describes the organization of these units, the processor order structure, and some rationale as to why certain design choices were made.*

## I. INTRODUCTION

The goal of this paper is to provide a general view of the organization and structure of the No. 2 ESS control unit. An attempt is made to emphasize the background and philosophy behind some of the design choices and their relation to the broad requirements of an economical, flexible, and reliable stored program control system.

### 1.1 *Definition of Control Unit System*

The role of the control unit is to direct the processing of calls and maintenance in the system. It is comprised of the central processing unit, the stores, and the interface to the peripheral units. Two control units and the maintenance and administration center are referred to as the No. 2 ESS control complex. The various subunits which form a control unit include the program control, the input-output control, the call store, and the program store.

The program control is the central processing unit which executes the instructions read from the program store. The input-output control performs the function of interfacing with the peripheral units by providing circuits such as a central pulse distributor, address bus drivers, and scan answer bus receivers. In addition, the input-output control autonomously performs certain processing functions associated with high rate, time critical, peripheral communication. The call store is

the variable memory used to hold information of a transient nature and also that which must be easily modified. The program store is a read-only memory which contains the operational programs as well as parameter and translation data. Physically, the program control, input-output control, and up to 16,384 words of call store are contained in one double bay frame called the central processor frame. Up to four 16,384-word modules are contained in a program store frame (65,536 words), four of these frames may be addressed by the system. A companion article in this issue has photographs of these frames.[1]

### 1.2 *Design Objectives and Their Influence*

The control unit was designed to control a small, low cost, electronic central office. Much of the economy was realized by the program and call store word sizes which, in turn, determine the size of various registers and gating paths in the processor. In addition, the use of a "micro" type of instruction repertoire allows the multiple use of logic elements for many of the instructions and the efficient use of the program store.

The command structure is somewhat tailored to the progress mark approach of call processing with efficient means of handling information contained in call registers or records.[2] However, the majority of instructions are general purpose, simple, and compact. Hardware check circuits are judiciously used in areas that are difficult if not impossible to check by program and where immediate detection is important. A high degree of internal access is provided for fault detection and diagnostic purposes. Much of this access is common to call processing as well as maintenance.

Some of these economies and the rather serial manner in which processing is performed sacrifice real-time capacity. A fairly modest investment in input-output wired logic minimizes the sacrifice by relieving the program of performing some of the highly repetitive simple functions and quite significantly avoiding the burden of a more frequent input-output interrupt with its associated overhead. This design resulted in a small enough quantity of hardware that an entire control unit forms one switchable system block, thus avoiding the expense and complexity of providing communication paths and control for switching subunits within the control unit.

## II. GENERAL ORGANIZATION

Figure 1 shows the No. 2 ESS control complex which consists of two control units and the maintenance and administration center. The

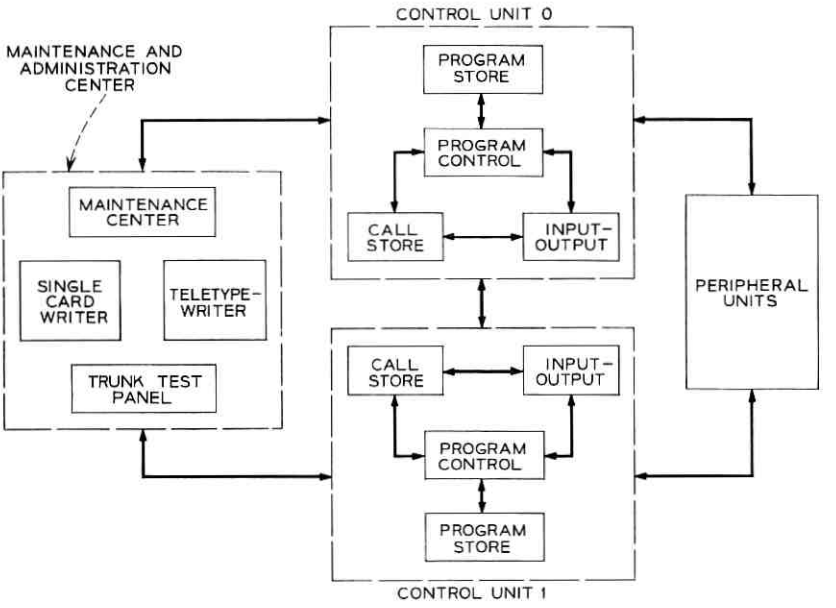



Fig. 1 — No. 2 ESS control complex.

control unit is duplicated and normally running in synchronism with its mate. One control unit activates the peripheral equipment and is called the ACTIVE or more descriptively, the "on-line" control unit. The control units are matched at the maintenance center. The call store is shared by both the program control and input-output control within a control unit. Each control unit has access to only one peripheral communication link but each link provides access to the duplicated controllers of the peripheral unit frames.[3]

The subunits are defined in Fig. 1 and are described in detail later in this article. The bus system is considered part of the control unit and a fault in the active bus will require switching of control units. Communication between control units is for maintenance purposes as in the communication between the control units and maintenance center.

The savings involved by avoiding the interswitching of subunits within the control unit has already been mentioned. Additional savings result from the compact nature of the design in that simple direct wire (dc) communication is possible among all units in the central processor frame, between the pair of control units, and between the control units and the maintenance center. The need for

transformer-coupled (ac) drivers, receivers, and receiving registers is avoided in these instances. However, the nature of the program store signals, the store's physical size, and the growth needs, dictate the use of transformer-coupled paths for the address and read out signals between the program control and program store. This is not a severe economic penalty since there are only about 20 pairs involved in each direction. For the most part, the connections between major units are handled via plug-in cables. This connectorization of blocks leads to advantages in testing, installation, and growth and is covered in detail in a separate article in this issue.[1]

## III. LOGIC CIRCUITS AND STORES

### 3.1 *Building Block Logic Circuits*

The logic circuits of the control unit are fabricated using the transistor-resistor NOR logic gate of Fig. 2 as a basic building block. Constructed with high-speed silicon transistors and tantalum-nitride thin film resistors as described in a companion article in this issue,[1] this gate, with a typical propagation delay of 35 nanoseconds, competes favorably with other available discrete component logic forms from an economic point of view. To provide building blocks with medium and high fanout capability, and a binary counter-shift register function, variations on the basic gate have been developed. These are shown in Fig. 3.

The fan-out of the basic gate is three and the fan-in is two. By connecting the collectors of a number of transistors to a common load resistor, a gate with greater fan-in (maximum of 40) can be formed, although not without suffering some loss of speed. The margin of this gate against false turn-on is determined primarily by the difference between the saturated collector-emitter voltage and the conduction threshold of the base-emitter junction of the transistor. In the worst case, this is only about 200 mV, and this imposes a constraint

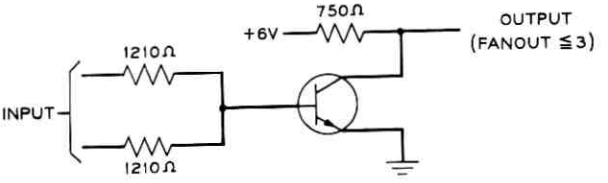



Fig. 2 — Transistor-resistor logic gate.

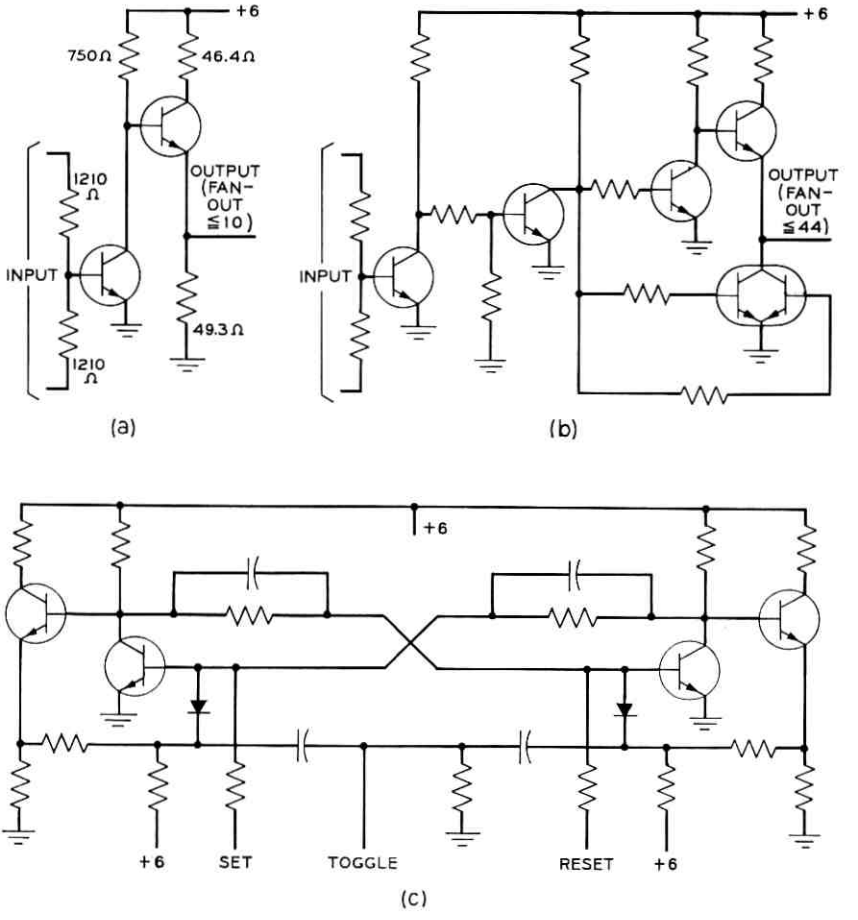



Fig. 3 — (a) Intermediate fan-out gate, (b) high fan-out gate, and (c) binary counter-shift register.

on the backplane wiring of the machine. The wiring is also constrained by the need to avoid excessive negative noise pulses and ringing on signal leads since these may cause "on" transistors to turn off falsely. In some cases, "catching diodes" and terminating resistors have been connected on the far end of long signal leads to prevent negative undershoots from turning off gates. In the development of the No. 2 ESS, these various constraints have been evaluated and wiring rules formulated to assure reliable performance of the circuits over the range of operating voltages and environmental conditions.

### 3.2 *Call Store Circuit*

The call store circuit provides the temporary information storage required in the control unit. It is a random-access memory of 8192 words of 16 bits each and operates with a destructive read-write cycle time of 6.0 microseconds.

#### 3.2.1 *Memory Device*

The basic memory device in the call store circuit is a 1-inch square ferrite sheet approximately 30 mils thick, having a 16-by-16 array of 25 mil diameter holes located on 50 mil centers. [4,5] This is the same device that is used in the No. 1 ESS call store circuit.[6] Two hundred fifty-six sheets are wired together to form a 4-wire coincident current memory module of 4096 sixteen-bit words. Two wires are used for access current ($X$ and $Y$), a third ($Z$) for inhibit current, and the fourth is used for sensing the readout signal.

The wiring pattern used in this construction is designed to maximize the readout signal-to-noise ratio by minimizing the coupling of access circuit transients into the readout circuit and by canceling the delta noise along the readout signal wire. Signal-to-noise ratio is further controlled by time-staggering the application of drive currents in reading the memory. The sheet material characteristics are such as to allow a 1.90 microsecond destructive read access time and total cycle time of 6.0 microseconds with coordinate access currents of 280 mA and 0.4 microsecond rise times.

#### 3.2.2 *Call Store Operation*

Figure 4 is a block diagram of the call store circuit. Address and input data signals are brought to the store by dc connections to the call store address and call store input registers in the program control circuit. Output data from the store is delivered by dc connections to either the general register in the program control circuit or the digit-data output register in the input-output circuit depending on which of two kinds of read command is being executed. Because of these dc connections, no registers are needed in the call store itself, and none are shown in Fig. 4.

The $X$ and $Y$ access windings are the crosspoints of bilateral voltage selection matrices in each of which one horizontal and one vertical coordinate is selected by the address information presented to the store at the time of a read or write command. This selection is made by the address decoders and bilateral switches under control of the

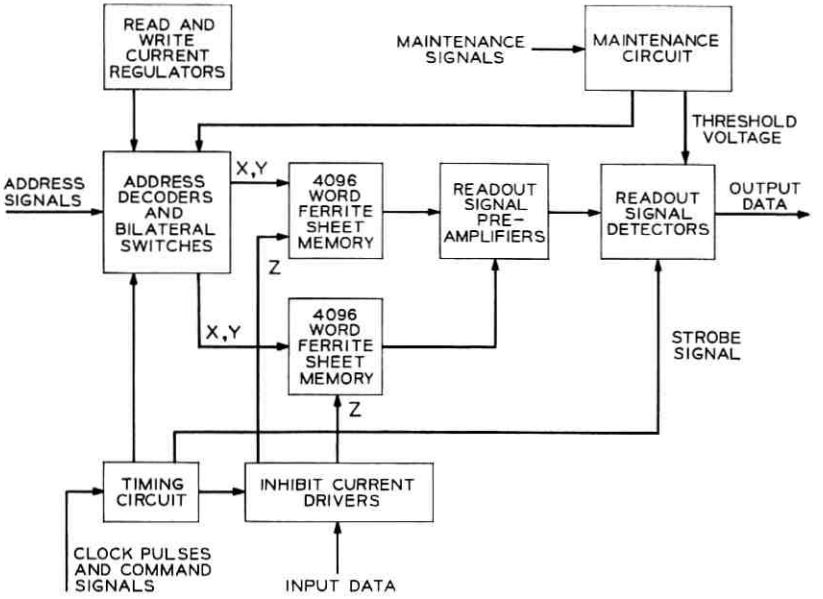



Fig. 4 — Call store block diagram.

timing circuit shown in Fig. 4. Access current flowing through the switches and selection matrix is controlled in both amplitude and rise and fall times by the read and write current regulators. Output signals from the ferrite sheet memories are amplified and discriminated during a specific time interval under control of a strobe signal generated by the timing circuit. This strobe signal is inhibited by the program control on "call store write" commands when the information being read is to be destroyed.

Following the read operation, the access currents reverse polarity to accomplish writing. This will switch the ferrite material at the selected holes to the "one" state except at those holes where inhibit current is applied. This is done on the inhibit windings of those bits which are to be zero, thereby cancelling the effect of half of the access current. The access currents persist long enough to switch all of the material surrounding the selected holes to the "one" state in the absence of inhibit current. Following the write operation, a post-write disturb current is applied in the read direction on the inhibit windings in order to adjust the magnetic state of the ferrite material to achieve minimum delta noise.

### 3.2.3 *Maintenance*

In Fig. 4, a "maintenance circuit" is shown. This circuit monitors the operation of the access circuitry to insure that on each store cycle the access currents flow in the correct direction at the correct time and that exactly two of the bilateral switches on both the $X$ and the $Y$ selection matrix operates. On any one cycle, if all of these conditions are not met, a "call store error" signal is sent to the program control and maintenance center circuits, indicating a fault in the access circuits of the store. In order to insure the integrity of these checking circuits, maintenance signals are brought into the store which logically simulate the access failures that should be detected by the check circuits.

Other maintenance signals which cause the store to write all 1's or all 0's at a particular address can be activated by the maintenance programs to facilitate diagnosis of faults in either the store or the circuits to which it connects. Finally, there is a third set of maintenance signals by which the threshold voltage of the sense amplifiers may be varied, either manually or by program, to evaluate the operating margins of the store.

### 3.2.4 *Growth*

In order to achieve minimum cost in the smaller office sizes, the call store has been designed to provide only 4096 words by eliminating one of the memory modules. Furthermore, the physical design of the call store and the program control has included connectorization to allow for growth of up to four 8192-word stores, any of which may be partially equipped so that the call store may be any size from 4096 to 32,768 words in 4096-word increments. The smallest office is expected to use 8192 words.

## 3.3 *Program Store*

The program store circuit provides memory for semipermanent information in the No. 2 ESS control unit. This information is the program and translation data used by the system. It is called semipermanent because it is fixed and independent of the calls being processed by the system at any given time. However, as the services provided by the system are changed, or as lines and trunks are changed, the program and translation data must change. The store is a random-access read-only memory of 65,536 words of 22 bits each and operates with a 2.5 microsecond access time and a 6.0 microsecond

cycle time. Each half of the control unit may be equipped with up to four stores (in the largest office), and each store may be partially equipped to provide as few as 16,384 words.

### 3.3.1 *Memory Device*

The program store uses the permanent magnet twistor memory for the basic storage device, in a fashion similar to the way it is used in the No. 1 ESS program store.[7] Information is stored in the state of small magnets affixed to aluminum cards, and the state of the magnets is sensed by the twistor wire. A detailed description of the actual readout process has been published in a number of earlier papers. The present store differs from No. 1 ESS in its organization, maintenance features, and capability for growth. These aspects of the store are described in the following paragraphs. As in the No. 1 ESS store, information is changed by withdrawing the magnet cards from the twistor module and changing the state of the magnets by means of a special writing mechanism. The operational aspects of this feature are described in some detail in Section VI.

### 3.3.2 *Program Store Operation*

Figure 5 is a block diagram of the store. Because of its physical size and the cable lengths that would be encountered when four stores are provided in an office, the store is connected to the rest of the control unit by way of an ac bus system. Thus, within the store, there is an 18-bit address register which is cleared and filled by the signals on the address bus. All stores receive the address information and a "read" command signal from the bus and the store identified by the two most significant bits of the address responds on the answer bus. A store contains up to four connectorized memory units, each of which contains a twistor memory module and its associated address decoding and high current access circuitry. Each unit stores 16,384 22-bit words in the form of 4096 88-bit words. Readout signals from corresponding bits of the four memory units are paralleled through a resistive impedance matching network and amplified in 88 readout amplifiers.

On the basis of two of the address bits, a one-out-of-four selection is made on the 88-bit word to produce a 22-bit word which is transmitted to the program control on the answer bus. Since, on the basis of the address, only one memory unit is active at any time, the access current regulation, timing control, readout circuits, and answer cable drivers are common to the four memory units.

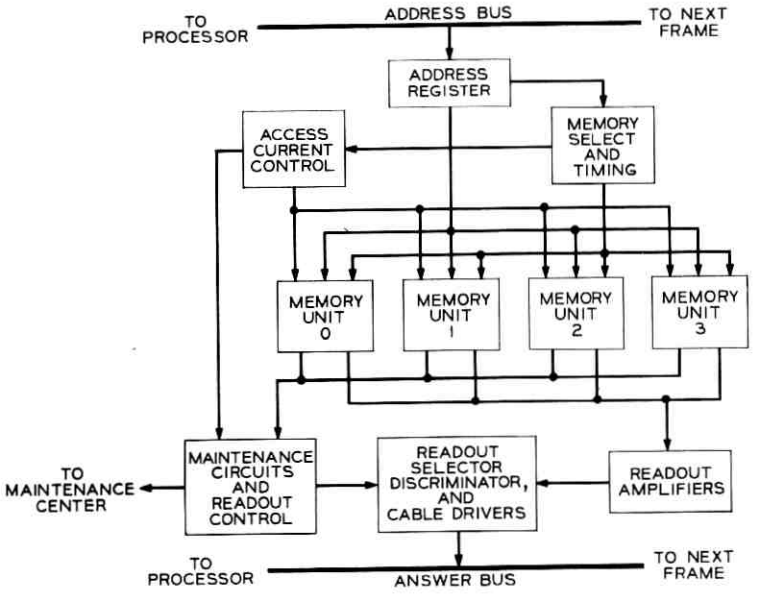



Fig. 5 — No. 2 ESS program store.

3.3.3 *Maintenance*

In order to provide rapid detection of access circuit failures, the program store includes checking circuits to verify the correct operation of the store on every read cycle. These circuits verify the correct flow of module access currents and the correct selection of a unique current path through the selected module. In the event that these conditions are not met in a given read cycle, a "program store error" signal is sent to the program control and maintenance center circuits and certain bits of the readout information sent on the answer bus are forced to be "ones," depending on which of the check circuits detected the failure conditon. This provides for fast resolution of a failure to specific circuit packages by the diagnostic program. All that is required is that the program read a word which contains all "zeros." As in the call store, there are dc control signals brought to the program store from the maintenance center to simulate the various kinds of failures that the check circuits are designed to detect. These signals are used to verify the integrity of the check circuits and may also be used in resolving certain kinds of failures.

There is also a set of maintenance signals which provide for varying

the sense amplifier threshold voltage either manually or by program to obtain a measure of the operating margin of the store. There are two discrete amounts by which the threshold can be varied by the program from its nominal position. These are called "inner margins" (closest to nominal) and "outer margins." A store which works properly with the threshold voltage at the "outer margins" is presumed to be in good health and to contain well written magnet cards.

A store which fails to work with outer margins applied but does work with the inner margins applied is presumed to contain either an incipient hardware fault or poorly written cards. If replacing the cards with a well written set does not enable the store to operate with outer margins applied, then the existence of an incipient failure may be concluded. If a store fails to work when "inner margins" are applied, then a hard fault is presumed to exist in the store. Thus, by virtue of the access checking circuits and the program controlled maintenance signals, hardware failures and magnet card writing errors can be quickly and accurately detected and diagnosed.

#### 3.3.4 *Growth*

To provide orderly simple growth procedures, as well as to minimize installation and repair time, the physical design of the program store uses connectors at both the frame and unit level. This means that a frame can be added to an office by simply bolting it to the floor, connecting power, and connecting it to the adjacent frame by plug-in connectors. Furthermore, a memory unit can be added to an existing frame (either as growth or replacement of a failed unit) by simply bolting it in place in the frame and plugging umbilical cables from the unit into matching connectors in the frame. As a result, a given office can have from 16,384 to 262,144 words of storage in 16,384-word increments. The estimated time to add an increment to an existing frame or to add a frame is less than one hour.

## IV. ORDER STRUCTURE

The No. 2 ESS order structure was designed with emphasis on data processing rather than arithmetic operations. The basic 22-bit instruction words, shown in Fig. 6, are divided into two types. The type 1 contains one instruction with a 5-bit operation code, 16-bit address, and a transfer check bit. This check bit is used for detecting illegal transfers and has proved to be quite useful in indicating hardware faults as well as program faults resulting from bugs or data mutilation. The instructions belonging to the type 1 class are rather few.

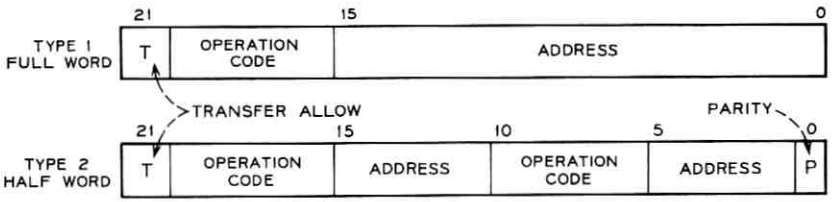



Fig. 6 — No. 2 ESS instruction formats.

They are used primarily for absolute program transfers and for supplying constants and masks for various functions.

The type 2 word consists of two 10-bit instructions, each with a 5-bit operation code and a 5-bit address. As for the remaining two bits, one is assigned as a transfer check bit and the other as a parity check bit. The 5-bit address is used to denote a value or a modifier. For example, a value associated with a rotate instruction specifies the amount of rotation. A modifier associated with the gating operation specifies the source and destination register combination. In many cases, the combination of the two 5-bit fields can be more appropriately considered as a 10-bit operation code rather than the division just outlined. In addition, the 10-bit operation can be assigned as a full-word instruction and the second 10-bit field is available for option bits, modifiers, addresses, or other pertinent data. The majority of instructions fall into the simple type 2 category and provide for efficient program storage use.

### 4.1 *Clock and Cycle Timing*

The operation of logic circuitry within the central processor is generally synchronous, whereby the clock circuit provides all the basic timing signals. These signals define a machine cycle and phases within that cycle. The duration of a machine cycle is primarily determined by the rate at which the instruction can be fetched from the program store. With the type 2 order format of two instructions per program word, a machine cycle is set equal to one-half of the program store access time, or 3 microseconds. The number of clock phases is chosen to permit many of the short word instructions to be implemented in one cycle, thereby making it possible to process two short word instructions while the next word is being fetched from the program store. This maximizes the average data processing rate of the system.

A machine cycle consists of eight 750-nanosecond clock phases with

each pulse overlapping the preceding and succeeding pulses by one-half their width (375 nanoseconds). This arrangement provides an adequate number of intervals to fully realize the data processing steps of the system in response to a diversity of instructions. At the same time, it is consistent with the speed of the transistor-resistor logic gate. The basic timing signal is provided by a 2-⅔ MHz crystal-controlled oscillator with the output from the on-line oscillator driving the clock in the off-line processor. In this way, synchronism between the processors is maintained.

The clock circuit provides timing signals throughout the entire system. These signals coordinate the various activities as dictated by the program sequence. Improper timing not only causes errors in the system, but causes them in such a manner that they are extremely difficult to diagnose. For this reason, error detection circuits have been incorporated into the clock. Furthermore, accessibility is provided to allow ease of program check of the clock circuitry for fault detection and diagnosis.

### 4.2 *Order Decoding*

The central processor decodes and executes only one instruction at any given time. Most of the orders are relatively simple, that is, setting and clearing various registers and bits, gating data from one register to another, or testing various registers and bits. The decoding of some simple instructions leads directly to the specified actions without any additional logic circuitry. In the more complex instructions, outputs from the order decoder are combined logically with the appropriate system conditions and clock signals, thus producing the necessary controls required for implementing a given order.

Each order, regardless of its format or complexity, has a 5-bit operation code and is decoded into one lead active out of 32. For most full-word and for some half-word orders, the address portion of the instructions (16-bit or 5-bit) is used directly as an address or a constant. These instructions are concerned mainly with providing direct addresses for program transfers and call store access, loading registers with constants, or providing data for certain logical operations. For the remaining half-word orders, the address and the operation code form a 10-bit binary combination that specifies a unique instruction. These instructions are decoded by two stages of one-out-of-32 decoders (Fig. 7). Eight outputs in the main decoder are dedicated to enabling one of eight auxiliary decoders. Each one decodes the 5-bit address to one lead active out-of-32 basis when it is

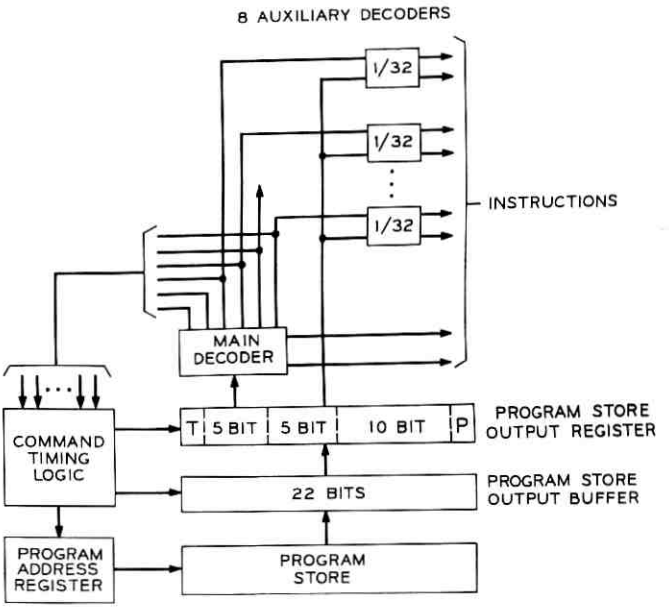



Fig. 7 — Instruction order decoding arrangement.

selected by the output from the main decoder. This arrangement accommodates numerous short and simple instructions.

The decoder outputs extend throughout the system to implement the prescribed actions for each instruction. Outputs for a subset of orders extend to the maintenance center and to the duplicated central processor to permit the on-line machine to communicate with the off-line processor for maintenance purposes. The main decoder outputs also combine logically in the command timing logic to control the flow of instructions from the program store to the decoders.

### 4.3 *General Description of Order Flow*

The flow of program store information is normally in sequential order with the program address incremented by one each time the store is read. At times, it is necessary to branch from the regular sequences and transfer to a new one. The address will then be changed completely corresponding to a starting point for a new sequence.

The 22-bit output from the program store is first temporarily stored in a flip-flop register. It acts as a buffer for storing the information until the preceding instruction has been fully processed. The

information is then gated to the program store output register for decoding and executing. Immediately thereafter, the program address counter is advanced by 1 and the program store is instructed to read the next sequential word.

The buffering action of the program store output buffer permits the next program word to be read while the instruction in the program store output register is being executed, thus reducing the overall sequencing time. If a transfer should be executed, the revised address, rather than the next sequential location, will be used in addressing the store. However, if the next location has been read, and a transfer should be executed, the contents of the program store output buffer will be discarded. The store is read at the new location.

In many cases, concurrent operation of executing the present instruction and reading the next program word requires no additional time because of transfers. If the word contains two short instructions, the decoder acts on the left one first. After the first instruction has been executed, the second instruction is then gated to the left half of the program store output register for decoding and execution. Upon completion, the new information (the next program word) in the program store output buffer is gated to the program store output register. The instructions are thus stepped along one at a time to the decoder circuit.

The program store can be read once every 6 microseconds. Consequently, most of the half-word instructions are implemented in 3 microseconds to allow the program store to operate at its maximum rate. The full-word instructions are allotted a minimum of 6 microseconds since the next word is not available for that length of time.

### 4.4 *Data Transfer and Register Functions*

The central processor, as shown in Fig. 8, contains a large number of flip-flop registers in addition to the data processing logic. In general, the content of any one register can be gated to any other register in the system. The transfer of information is done by means of the common bus designated as the program gating bus. Most of the registers are 16 bits wide, with two sets of gates (input and output) associated with each. The corresponding bits from the output gates are ORed to form the common bus outputs, fanning out to all parts of the system.

Normally, all the input and all the output gates are disabled by their respective control signals. In transferring information from one

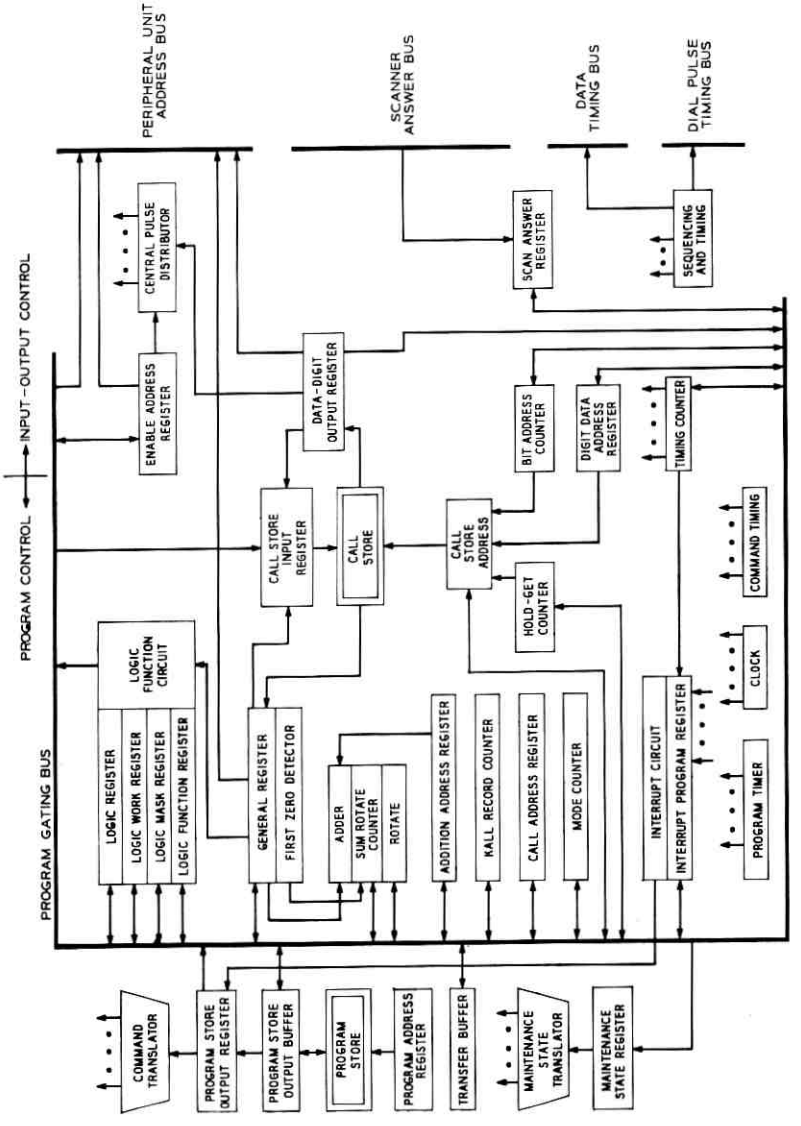



Fig. 8 — No. 2 ESS call processor block diagram.

register to another, the output gates from the first are enabled to the gating bus and the input gates to the second are enabled from the bus. The use of a common bus is an excellent method of transmitting data signals within the processor, keeping the number of gates and the number of long leads to a minimum. The bus outputs extend to the input-output circuit, the duplicated control unit, and the maintenance center to allow data transfers via the bus among the circuits under programmed control.

There are several registers associated directly with the operation of the call store. As Section V elaborates, this store is time-shared between the processor and the input-output circuits. The outputs from the store are directed either to the general register in the processor or to the digit-data output register in the input-output depending upon which circuit is accessing the store. In addressing the addition address register or the call address register may be gated to the call store address register for use as an address in reading or writing. The data in the call store input register, normally gated from the general register, is the store input.

The registers mentioned thus far are related mainly to the operation of the call store and the program store. Other functional flip-flop registers are used either with associated logic circuits for data manipulation by the program or are dedicated to various system functions. The data modifications involve such functions as addition, rotation, identification, and marking the least significant zero in a register, and various logic functions of two variables. Some of the specific system functions are concerned with interrupt facilities, subroutine facilities, program timing, and maintenance features.

Adding to the numerous flip-flop registers in the system, is the call store, which is segmented into many registers. Much of the call processing program is concerned with call store registers. A group of eight words is assigned as a register or record for each call in progress. Several instructions are provided for quick access to these records.

### 4.5 *Transfer and Decision-Making Orders*

Call processing programs tend to be highly decision-oriented. In order to facilitate decision making, a multiplicity of test instructions are provided to check various data words and individual bits of certain data words. Test instructions indicate their result by the state of the condition flip-flop which is later utilized by a conditional transfer instruction. If the specified test condition is satisfied, the condition flip-flop is set to the "1" state; otherwise, it is reset to the "0"

state. Many system functions also affect the state of the condition flip-flop. For example, in adding two binary numbers, the end carry is stored in this flip-flop. Most of the decision making is based the state of the condition flip-flop.

Many conditional transfers involve short jumps, hence coding economy can be realized by the use of a 5-bit address. When a transfer is made, the low order five bits of the current address are modified. This range, however, can be extended by preloading the transfer buffer with the next five higher order bits. In this case, the five bits from the instruction and the five bits from the transfer buffer form the low ten bits of the program store address when a transfer is made. In addition to short transfers, the entire range of the 18-bit program store address is accessible by the long, or full, transfer instruction. The low 16 bits are provided by the instruction while the remaining two bits come from the transfer buffer. Indirect transfers, that is, transfers to an address contained in a system register, are also provided.

### 4.6 *Data Manipulation Orders*

Data manipulation orders allow the operation on data, contained in system registers, in a particular manner to implement the call processing functions. Logical rather than arithmetic needs are paramount. However, a binary ADD order is included in the instruction repertoire. This allows other arithmetical operations to be implemented rather easily by a software combination of other logical operations.

A very flexible and useful set of orders for data manipulation is the Boolean function of two variables. The general register and the logic register are the two variables which combine logically to form the four minterms ($\bar{G}_n\bar{L}_n$ , $\bar{G}_nL_n$ , $G_n\bar{L}_n$ , $G_nL_n$) bit by bit. The subscript $n$ represents a unique bit of the general register ($G$), or logic register ($L$). The 4-bit logic function register ($a$, $b$, $c$, $d$) specifies the coefficients of the minterms to give the 16 possible combinations. The Boolean function of two variables represented by:

$$f(G_n , L_n) = a\bar{G}_n\bar{L}_n + b\bar{G}_nL_n + cG_n\bar{L}_n + dG_nL_n$$

generates many useful logical functions. For example, if $a = d = 1$ and $b = c = 0$, the function ($\bar{G}_n\bar{L}_n + G_nL_n$) is used to match or compare two binary numbers. The results are placed in the logic work register and, optionally, also placed in the general register. The logic mask register controls or selects the bits in which the logic function is to be performed.

Most of the data in the call store are continuously undergoing changes. In many instances, the data read out are logically operated on with the result written back into the store. This usually involves a logical operation between two memory cycles. Since the operation of read-change-write occurs rather frequently in the system program, a considerable savings in program words and real-time can be realized by combining all three steps. This is possible within the framework of one memory cycle with no additional timing penalty. More importantly, it eliminates the possibility of having an interrupt program modifying the same word in the midst of read-write-change operation. Therefore, the store interwrite problem is avoided. Figure 9 shows that the data return to the call store input for writing may come directly from the general register or from the logic output via the gating bus or a combination of the two. The combination is a form of insertion masking which permits the bits specified by the logic mask register to be returned unchanged directly from the general register to the call store input register. The other bits are modified according to the content of the logic register and the setting of the logic function register.

Other useful data manipulation functions include rotation of a register's contents and a particular function of detecting and identifying the rightmost "0" in a register. The latter function's implementation by several conventional instructions is rather awkward and time consuming. Therefore, a special order, find low zero test, is provided to facilitate operations such as the selection of idle trunks. The status of each trunk is indicated by a binary bit, a "0" corresponds to the idle condition and a "1" to the busy condition. A word of 16 bits represents a group of 16 trunks with the bit position denoting the trunk number. In the find low zero test instruction, the logic searches for the first zero in the general register, starting with the least significant bit, and successively checks through the higher order bits in sequence until a zero is found. When a zero is detected, the logic automatically ignores the remainder of the bits, sets the selected "0" to "1," records the bit position, and marks the condition flip-flop to "1" to indicate finding an idle trunk.

### 4.7 *Interrupt Facilities*

A multilevel interrupt mechanism is provided in the central processor to permit a signal to enter at any time and seize control of the system. The interrupt circuit responds to an interrupt request by injecting or replacing the next program word with a program in-

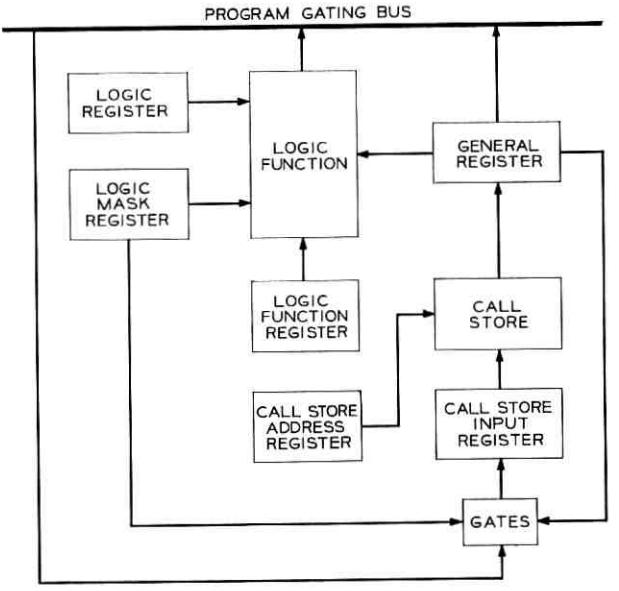



Fig. 9 — Call store operation.

terrupt begin order. The order stores the program address where the program was interrupted, and then the program transfers to the entry point of the appropriate interrupt program. The first chore in the execution of the interrupt program is to store the state of the central processor in memory. At the conclusion of the required work function, the central processor is restored to its initial state. The interrupt program then ends with a program interrupt end order which transfers back to the previous interrupted program.

Presently, interrupts are used for two functions. One is to handle input-output data processing, that is, digit receiving, digit sending, scanning, data sending, and peripheral communication functions. These tasks are scheduled regularly in real time at 25 (or a multiple of 25) millisecond intervals.

The other function is to gain immediate access to one of several maintenance programs. The interrupt program register and logic are arranged in a hierarchy of eight levels to allow the running of the program with preference to the most urgent or highest level present at any time. Only three of the eight interrupt levels have been assigned; two for maintenance and one for input-output. The remaining five are for future growth and other applications of the No. 2 ESS control complex.

### 4.8 *Subroutine Facilities*

Many of the No. 2 ESS programs consist largely of a series of subroutine calls. This not only has the advantage of saving program store words, but also makes debugging simpler since a subroutine need only be debugged once. A special push-down address arrangement has been implemented and storage allocated in the call store to facilitate entry, nesting, return, and releasing of major system registers for use by subroutines. This arrangement provides 32 groups of eight words within the call store. Each time a new subroutine level is entered by means of a transfer and save address order, the push-down level or the hold-get counter is decremented by one. In execution of the transfer and save address order, the return address is stored within the eight-word block. The remaining words are available for the major system registers. Special HOLD and GET instructions are provided to facilitate storage and retrieval of data between a selected register and the call store. When the subroutine is returned via the transfer to saved address order, the push-down level (hold-get counter) is incremented by one.

### 4.9 *Orders for Repetitive Functions*

Certain tasks are very repetitive and can be accomplished by a sequence of operations which is repeated a predetermined number of times forming a loop. Each time the program is repeated, different data are encountered. The transfer on index instruction in the No. 2 ESS is used primarily for program loops. An 8-bit transfer on index counter (kall record counter) is used to keep track of the number of loops. In executing the transfer on index instruction, when the kall record counter is not equal to 1, the kall record counter is decremented by 1 and a transfer is made to the beginning of the loop to repeat the given set of operations.

Many of the repetitive functions can be implemented rather easily with program loops using the transfer on index instruction. However, there are some repetitive functions that must be performed very frequently by the system. Although the program to be repeated usually contains only a few instructions (including transfer on index), the program's high usage consumes a considerable amount of system time. A real-time saving can be realized by combining these instructions into a single highly efficient macroinstruction. One order thus performs the actions of several and is repeated as many times as is specified by the contents of the kall record counter unless the data are

being tested by the program and satisfies the exit conditions prior to that. The real-time saving is obtained chiefly by: (*i*) not having to fetch a series of instructions from the program store, (*ii*) not having to perform a transfer at each loop back, and (*iii*) concurrent operation of a set of instructions.

There are several repetitive or macroinstructions used in the No. 2 ESS. Among these are: (*i*) an instruction for performing logic on the contents of a number of words contained in call store, (*ii*) an instruction that scans ferrods and compares the results with call store status words, (*iii*) an instruction for sending serial data to peripheral units, and (*iv*) an instruction that scans call store records and performs indirect transfers to programs that process the records. The last instruction is the ADV (advance) command and is described in the call processing article in this issue.[2] The ferrod scanning instruction as well as the other peripheral unit orders are described in Section 5.4.

### 4.10 *Maintenance Facilities*

In the duplicated system under normal conditions both the on-line and off-line processors are run in synchronism while handling the input-output data and solving their problems independently. Only the outputs from the on-line processor are permitted to activate the external equipment, such as setting up network connections, outpulsing, and data sending. In this synchronous mode, both call stores contain identical information and both processors are in step, executing the same sequence of instructions. As a detection means, the information being written into the call stores is matched by circuitry in the maintenance center. In addition, a number of error checking circuits are strategically located throughout the system for the detection and diagnosis of faults. They provide:

(*i*) Parity checks of words read from the program store (odd parity for the program words, and even parity for the data words),

(*ii*) Transfer checks to ensure a legal program entry,

(*iii*) Double output checks for more than one output active from the instruction decoders,

(*iv*) Sequence checks of the instruction timing logic,

(*v*) Subroutine level checks for exceeding the number that can be nested,

(*vi*) System clock checks for the proper sequence and rate of the clock output signals,

(*vii*) Simultaneous access checks of the call store by the processor and the input-output control,

(*viii*) Multiple access checks in the call store and the program store,

(*ix*) Timing checks in the input-output control for checking dial pulse and data timing buses, and

(*x*) Input-output error checks involving proper access to and response from peripheral units.

Check circuits are generally provided to cope with faults which are too difficult or too time consuming to detect strictly by program. An example is the double output detector which checks for more than one output active from the instruction decoders. Within the central processor one and only one instruction is decoded and executed at any given time. Faults causing no output at all are relatively simple to detect.

If, however, the instruction decoder produces two simultaneous outputs, an additional extraneous instruction will be performed. This can seriously affect the overall operation of the system and produce obscure errors. Therefore, the double output check circuit is incorporated as part of the processor for quick detection and diagnosis. Except for input-output errors, when a check circuit in the on-line system indicates an error condition exists, the faulty processor is immediately taken off line. Input-output error signals are handled by special "working mode" programs which determine the proper actions to be taken.

As previously indicated, a method of trouble detection available when the control units are running in synchronism is the matching of call store information by a match circuit located in the maintenance center. A mismatch causes a maintenance interrupt program to run detection tests on the on-line control unit. Upon failure, the control unit is immediately switched off-line and stopped while the other control unit is given an initialized start. In some cases, the faulty processor may be incapable of making any decision. Under this condition, the hardware emergency timer will time out and initiate the switching and initiatization functions.

In order to analyze the defective circuit, the on-line processor has direct access to the off-line machine by means of a set of external orders. Simplicity of implementation rather than coding efficiency has been the goal in the design of these orders since the total number of external orders used in the system program is relatively small.

Consequently, each external order is a full word and takes two cycles to execute. The order format consists of a 5-bit operational field and two 5-bit addresses designated as the infield and the outfield. The operation code and the infield form a 10-bit binary combination which specifies the unique operation within the processor. Similarly, the operation code and the outfield specify the unique operation for the other processor. By a proper combination of the functions being performed in each of the two processors, many different operations are possible for communication and information transfer between the two processors and for direct control of the off-line processor. Some of the special orders are exceptionally useful in implementing the maintenance routines of the more difficult logic circuits, such as orders that permit the on-line machine to perform a step-by-step check of the off-line machine, or orders that start and stop the off-line clock to check timing leads.

After the fault has been diagnosed and repaired, the off-line processor must be restored to its normal working mode. In order to do this, the call store must first be updated to contain the same data as the on-line call store. If the complete transfer of the entire store is done by one continuous program, excessive time will be consumed, thereby causing errors in call processing. As a result, the data transfer from the on-line to off-line call store is interleaved with the normal program, moving a small block of words at a time until the entire store is updated. However, the data in the on-line call store is transient and the transferred portion of the store may undergo a continuous change. To ensure that the updated portion of the off-line store remains current, its address and store input data are "slaved" by hardware to the on-line store. When any read or write operation is performed, the same address and input data are used to alter the data in the off-line store. The updating of the call store is thus accomplished by the combination of software and hardware.

### 4.11 *Means of Checking Error Detection Circuits*

Error detection circuits are checked periodically to ensure their usefulness. When they fail in the error mode, the failures are self-detecting and are recognized immediately. On the other hand, if the failures prevent the circuitry from giving any error indication, faults normally detected will be ignored. Consequently, a periodic program check of error detection circuitry is essential to guarantee the reliable operation of these circuits. As an aid for testing the check circuits and for testing maintenance equipment that is not easily tested by

normal program functions, a maintenance state register and an associated decoder are provided for simulating faults and test conditions. The outputs from the maintenance state decoder are used, for example, to check the instruction double output detector or to enable a programmed margin check of the call store.

### 4.12 *Emergency Switching of Control Unit*

In any duplicated system, the mechanism for switching controls must be highly reliable. The emergency timer and associated circuitry provide the very vital function of switching the bad control unit off-line and the good control unit on-line. During the process, proper steps are taken to ensure a smooth transition in the transfer of controls.

The timer provides a further protection against faults or errors that cause programs to loop indefinitely or to bypass certain major program functions. The main call processing program is cyclic and returns to the starting point upon completion of all the tasks required by the call processing. The time required to complete one scan varies from scan to scan depending upon the amount of work required of the program. The basic timeout interval is 320 milliseconds. If the program takes longer or does not reset the timer, an equipment or program error has occurred. The timer and its associated logic then produce an output signal to switch controls to the other processor.

There are two timers, one in each processor. The off-line timer is used as "backup" and will take over the task of switching if the on-line timer does not work. Under most error conditions, immediate switching is performed to minimize any detrimental effect on call processing.

## V. INPUT-OUTPUT FACILITIES

In the No. 2 ESS, all inputs and outputs between the peripheral units and the central processing unit (program control) are handled by the input-output control. Figure 1 shows the control complex block diagram. The input-output control performs the interface function both by acting as an extension of the program control in providing and controlling the special purpose circuitry required to communicate with the peripheral units under control of program instructions and also providing wired logic for autonomously performing a portion of the digit receiving, digit sending, line scanning, and data sending tasks. By assigning the repetitive high rate functions which must be performed

on a tight schedule to the wired logic, program real-time is conserved by not having to perform these functions as well as not bearing the burden of an input-output interrupt and its associated overhead more frequently than every 25 milliseconds. The division of work between the input-output control and the input-output interrupt program was based on the criterion that only the simple, highly repetitive functions should be performed autonomously. To retain their flexibility, functions are directed by information placed by program in the common call store. In addition, economic justification requires the delegation to wired logic only those functions which return a substantial real-time saving for logic cost since the wired logic not only adds to the cost of the hardware for that function but also adds to the maintenance costs. This cost in borne in maintenance access as well as detection and diagnostic programs. The maintenance costs are often high because of the special purpose nature of the functions that must be tested and the difficulty in controlling autonomous circuitry.

### 5.1 *The Peripheral Interface*

A characteristic which frequently determines the general usefulness of a control system is its interface with the outside world (referred to here as the periphery). No. 1 ESS has developed an inexpensive yet attractive form of communication utilizing ac bus circuits over simple twisted pairs.[8] The bus frequency response, transmission characteristics, and noise immunity, satisfactorily meet the requirements and goals of the No. 2 ESS. Consequently, the natural conclusion was to design an "ESS 1-compatible" peripheral interface.

For economy and simplicity, the No. 2 ESS peripheral interface provides a limited bus structure with fanout in only two directions and access from only a single control unit. In addition to the buses which are used for parallel data transmission, a pulse distributor allows for private pair communication with particular receivers. Again, simplifications in the pulse distributor organization has resulted in further system economy. Figure 10 shows the peripheral connections.

#### 5.1.1 *Peripheral Buses*

The peripheral interface consists of the peripheral unit address bus, the scan answer bus, and the central pulse distributor. Both of the buses are groups of balanced twisted pairs which are transformer-coupled to and from the communicating circuits. The peripheral unit address bus is the output bus (from the input-output control) while the scan answer bus is the input bus. The bus character-

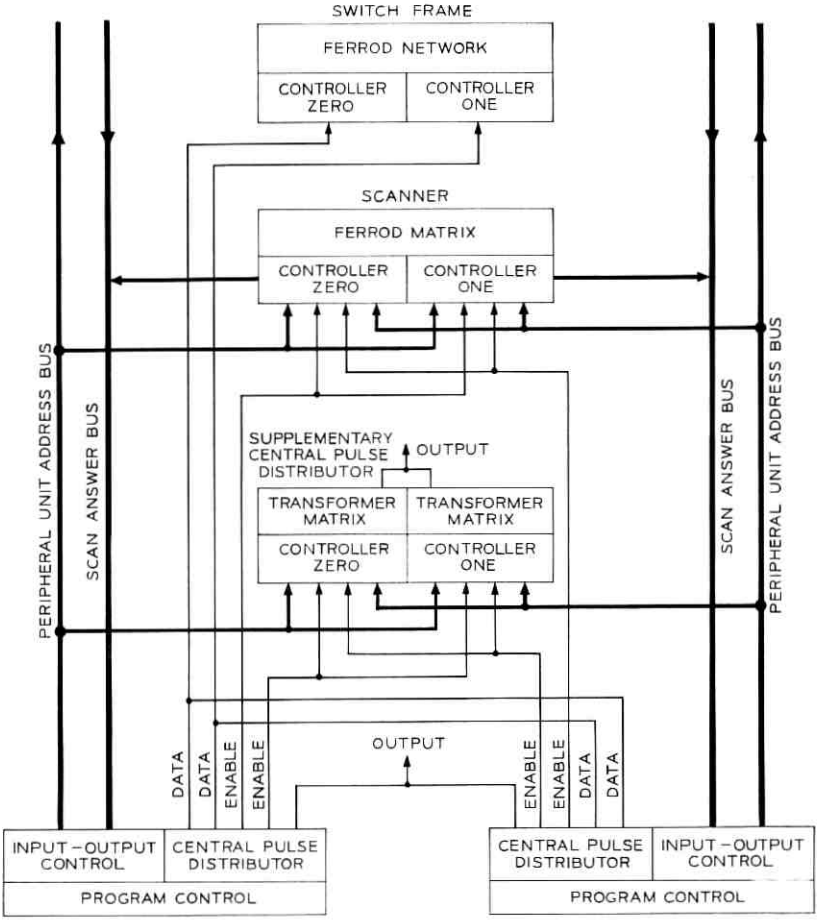



Fig. 10 — Peripheral connections.

istics are identical to those of No. 1 ESS; however, the cable drivers and receivers interface with transistor-resistor logic. Special input-output orders produce particular data formats on the peripheral unit address bus as well as general purpose binary information depending on the specific use.

### 5.1.2 *Central Pulse Distributor*

The local central pulse distributor is an integral part of the central processor frame and is directly controlled by input-output registers and control signals.

A signal from the central pulse distributor can be used for:

(*i*) Selecting and enabling a peripheral unit to receive information from the output bus (peripheral unit address bus),
(*ii*) Communicating with a shift register device via a stream of positive and negative pulses, and
(*iii*) Setting or resetting a flip-flop in the periphery.

There are a number of special purpose instructions which activate the peripheral interface in various modes which will be described in the next few pages.

The central pulse distributor is a three-stage transformer matrix through which a shaped pulse is steered to one-out-of-512 output pairs. The output is transformer-coupled and may be of either polarity depending on the selection address. The electrical characteristics are similar to those of No. 1 ESS,[9] while the organization and control is specific to No. 2 ESS. Selection of the output is determined by the enable address register for program orders and autonomous line scanning while the digit-data output register performs selection for autonomous digit scanning. Remotely located supplementary central pulse distributors allow growth beyond 512 outputs for high runner items such as trunk circuits.

The general use of the enable address register allows the high 10 bits to select the one-out-of-512 outputs and the polarity. The particular order which is used to simply transmit pulses through the central pulse distributor, as in uses (*ii*) and (*iii*) of Section 5.1, uses the low six bits of the enable address register to determine whether the local or a supplementary central pulse distributor is to produce the output pulse. A supplementary distributor contains a fully duplicated set of control logic and output transformers which provide multiple access to a common output pair, as shown in Fig. 10. Selection of a particular supplementary central pulse distributor and its control logic is determined by the low six bits of the enable address register. If these bits are all zero, the local central pulse distributor is selected. If not, five of the bits pick one of the supplementary distributors and the remaining bit selects the control logic. The system is thus capable of selecting up to 31 supplementary central pulse distributors.

An output pulse from the local central pulse distributor enables the selected supplementary distributor to receive the desired output code from the peripheral unit address bus which contains the upper ten bits of the enable address encoded into $\frac{1}{8}$, $\frac{1}{8}$, $\frac{1}{16}$. This coding allows

direct control of the supplementary distributor matrix as well as error checking to be performed at the supplementary distributor. Since the supplementary central pulse distributor requires use of the peripheral unit address bus to be enabled, it cannot in turn be used for enabling other circuits, but is used to provide independent signaling.

### 5.2 *Autonomous Work Cycle*

The input-output control performs certain tasks associated with digit receiving and sending, data sending, and line scanning. Since the input-output control requires use of the common call store and peripheral access, it interleaves its work with the program use of this equipment in order to avoid interfering with program execution. Figure 11 contrasts the call store interaction mode with the program interruption mode. Input-output work performed by "cycle stealing" not only saves the program from doing that work but also avoids the substantial overhead required to save and restore the state of the machine at the time of the interrupt. The call store usage priority structure

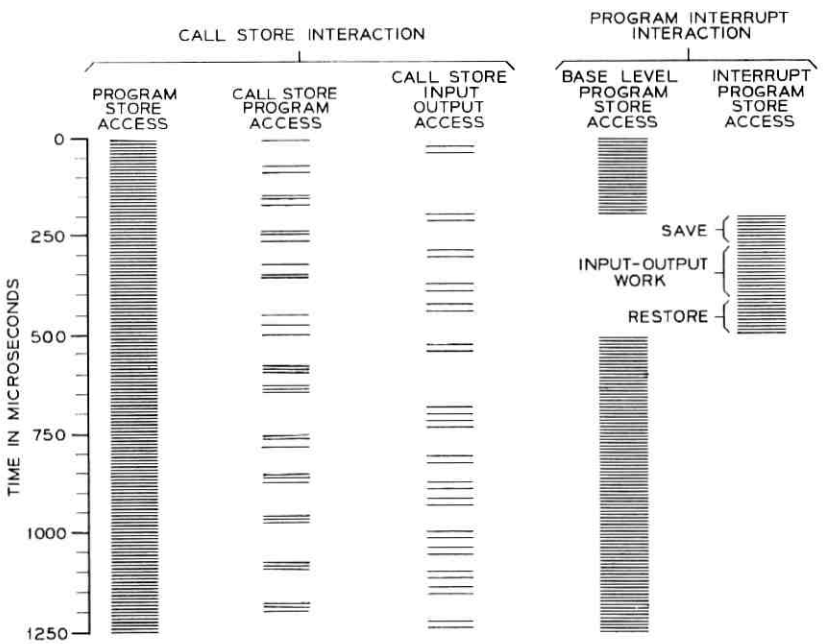



Fig. 11 — No. 2 ESS shared storage.

allows the program first choice, the digit scan portion of the input-output control second choice, and the line scan portion of the input-output control third choice. The input-output control has a mechanism for upping its priority in situations when the program usage has denied the input-output control sufficient access to keep up with its work schedule.

The input-output control is driven by a wired logic schedule which causes the performance of the various input-output tasks at the proper repetition rates. The lowest rate function determines the major repetition rate which, in this case, is dial pulse sending (10 pulses per second), requiring a 100 millisecond rate. A 10 millisecond rate is provided for scanning digit receivers. Division by eight yields the 1.25 millisecond interval which is the basic input-output work cycle used to control work slippage as described in the following paragraphs. The source of timing is the timing counter which is incremented by the system clock. The timing counter also generates the 25 millisecond input-output signal to interrupt the program so that the cooperative functions performed by the input-output interrupt program will be in step with the wired logic functions. The autonomous input-output control requires access to the peripheral unit address bus and central pulse distributor in order to interrogate scanners, and it requires access to the call store for control information, to record inputs, and to get outputs.

The primary function of the autonomous input-output control is digit receiving. Digits are stored in an eight word area of the call store (16 bits per word) called an originating register as shown in Fig. 12. There are a maximum of 128 originating registers, but the program can adjust the number to be used. The input-output control requires 12 microseconds to process an originating register, accessing two call store words and a ferrod scanner during that interval. The originating registers are sequentially scanned every 10 milliseconds with the input-output and program controls competing for use of the call store, peripheral unit buses, and central pulse distributor. The program control normally has priority in the use of this equipment in order to maintain efficient use of program time.

The 10 millisecond scan interval is divided into eight intervals of 1.25 milliseconds, each of which should handle 16 originating registers. This distributes the work load and minimizes the variations in scan rate. The work assigned to each interval is performed as soon as possible in the interval but may be delayed varying amounts because the program control is using the call store or peripheral unit access.

If the input-output control cannot complete the work assigned to a given 1.25 millisecond interval, it will raise the priority of its request for call store and peripheral unit access so that it can become active at the conclusion of the program control command that is denying the input-output control access. The input-output control remains in this force mode until work for that interval has been completed. In this way, the scan interval for originating registers is maintained at 10 milliseconds plus or minus 1.25 milliseconds.

Except when in the forcing mode as just described, once the input-output control determines that the program control instruction is noninterfering, it seizes the store and periphery and has control for 12 microseconds. The program control must now wait until the end of the 12 microseconds if it requires access to the call store or peripheral units. Since instructions may be multiples of 3 microseconds, the program control may have to wait 3, 6, or 9 microseconds. The input-output usage within a 1.25 millisecond interval is only 16 percent of the time. Computer simulations have shown that for a random program control usage of the call store or peripheral units at one-third of the time, the real-time loss resulting from input-output blockage is approximately 2 percent.

The 1.25 millisecond interval provides a convenient data bit rate (800 bits per second) for the data sending function. How data sending is performed and how it fits into the work cycle will become evident as more detail is presented.

## 5.3 *Input-Output Autonomous Functions and Their Relation to Programs*

### 5.3.1 *Digit Receiving and Sending*

As mentioned previously, digit receiving is the primary autonomous function of the input-output control but only those functions which require attention every 10 millisecond are performed by this logic. The remaining functions are performed by the input-output 25 millisecond interrupt program and the base level call processing programs. An originating register contains storage for up to 16 digits. Any originating register may be used in combination with a receiver or sender of many different types. Both receiving and sending of digits may be performed at the same time with the same originating register for nearly all combinations of receiver and sender types.

The input-output control reads the first word of the originating register to obtain the scanner address of the receiver ferrods and sends an order to interrogate the scanner. It then reads the second

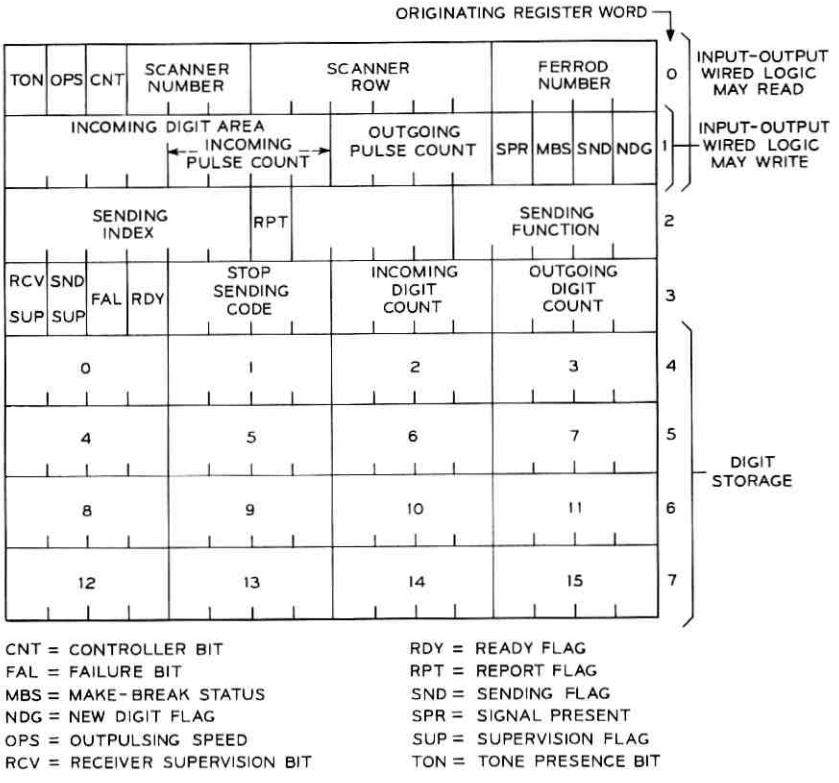



Fig. 12 — Originating register layout.

word of the register and combines this with the scanner reply and some information saved from the first word to determine the modifications to be made to the second word. The modified second word is then written back into the originating register and the input-output control goes on to the next originating register. These actions require two read-write cycles in the call store and a scanner interrogation, all performed in a 12 microsecond input-output cycle. The same cycle time is required for data sending since two 16-bit words must be read from the call store in order to get one bit for each of 32 possible data channels. The data cycle and 16 originating register cycles are allocated to a 1.25 millisecond interval. The combinational logic that modifies the second originating register word and the input-output program interaction combine to perform the various types of digit receiving and sending.

The input-output control is designed mainly to handle dial pulse, multifrequency, or *Touch-Tone*® telephone receivers. However, provision has been made for revertive and panel call indicator as well. In each of these cases, the cooperative action of both wired logic and program make up the whole job of receiving and sending digits. Rather than attempt to cover the details of each type of signaling, the following descriptions of dial pulse receiving and sending illustrate the type of interaction.

5.3.1.1 *Dial Pulse Receiving.* When an originating register is selected by call processing programs for dial pulse receiving, the first word is loaded with a number indicating the scanner, scanner word, and ferrod number in that word where the status of the pulsing relay can be observed. The remainder of the originating register is all zero. As the wired logic scans the originating register and digit receiver, every 10 milliseconds the state of the relay is compared with its pervious state stored in the make-break status bit (see Fig. 12). If there is a change from break to make, the wired logic increments the incoming pulse count. If there is *any* change in the state of the relay, the make-break status is updated and the new digit flag bit is *reset*. Every 125 milliseconds, the input-output 25 interrupt program checks the new digit bit and then sets it to one. If it does not get reset in a wired logic scan for a make-break status change over the next 125 milliseconds, the input-output 25 program will see the flag and assume pulsing is complete for the digit. The program checks that the party is still off-hook and inserts the digit in the proper position in the originating register, incrementing the count of digits received. Call processing programs check the originating register periodically to act upon digits as they are accumulated and to time between digits to insure that the interdigital timeout interval is not exceeded. Translation for office code, area code, and routing are performed by call processing programs.

Dial pulse receiving requires that the input-output 25 program check for a new digit flag at 125 millisecond intervals. Other types of receiving or sending require more frequent attention, so the input-output program looks for new digit flags at 50 millisecond intervals. This means that if a new digit flag is found in an originating register that is receiving dial pulses and the interval is not yet 125 milliseconds, the flag must be ignored. Dial pulses for the digit may still be arriving even though no change has occurred for 50 milliseconds.

The dial tone must be turned off by the program when the first dial pulse or tone is received after the receiver is connected. Since the input-output program is normally not flagged until a new digit is complete, some dif-

ferent means is required for the logic to detect when to flag the program to turn off the dial tone. Since digit sending cannot be in progress at this time, one of the values (zero) of the outpulse count is used to indicate dial tone is still on. The input-output control recognizes a nonzero incoming digit area, and sets a flag (sending flag bit SND) for the input-output program only if the outpulse count is zero. When the input-output program finds this flag on the next 50 millisecond pass, it will turn off the dial tone and set the outpulse count to its maximum value (decimal 15) to prevent the flag from being set again for this purpose. Both the outpulse count and the sending flag bit are normally used for sending functions as described in a later section. The outpulse count of 15 remains until sending is required, or the originating register is cleared, or until a second dial tone must be given.

5.3.1.2 *Dial Pulse Sending.* A dial pulse sender is enabled and disabled for pulsing by a private twisted pair from the central pulse distributor. Make-break timing for the dial pulses are supplied over common twisted pairs (buses) from the input-output control. Timing is provided for both 10 and 20 pulse-per-second sending.

When an originating register is to be used for dial pulse sending, call processing programs will place the sender circuit identity in the third originating register word, and indicate the type of signaling (dial pulse, tone, and so on) in coded form. It is assumed that the digits to be outpulsed were received in the same originating register or have been placed there by call processing programs. The digit number where sending is to start (outgoing digit count) and stop (stop sending code) are also entered in the originating register and the outpulsing speed bit is marked to indicate outpulse speed prior to sending. The last initializing action of call processing programs is to set the outpulse count to a value from 1 to 14, depending on the delay desired before sending starts. Normally, the input-output control will decrement the outpulse count at the outpulsing rate indicated (outpulsing speed) unless the outpulse count is 0, 1, or 15.

The outpulse count normally contains the count 15 when not in use for sending and the count 0 is used to indicate that dial tone has not yet been turned off for receiving. The count of one indicates that the number placed in the outpulse count by the program was decremented the proper number of times and that the input-output program should be flagged. The input-output control sets the sending flag bit as a flag to the program anytime the count of one is detected in

the outpulse count. The input-output program will see the sending flag bit on a 50 millisecond pass and set a constant in the outpulse count for interdigital timing, or obtain the next digit to be sent from the originating register and place the digit, plus one, in the outpulse count. This program also enables or disables pulsing for the dial pulse sender as required. The outgoing digit count is incremented by program for each digit. Sending proceeds in this way until the outgoing digit count is equal to either the stop sending code or the incoming digit count. This latter condition prevents sending from overtaking receiving in the case of overlapped operation.

5.3.2 *Data Sending*

Provision has been made for up to thirty-two 800 bit per second serial data channels, each of which could be associated with the control of a remote switching unit or other general data receiving facility. Data sending requires a message bit from the call store for each channel every 1.25 milliseconds. Corresponding message bits for 16 channels are read from the same word in call store by the input-output control each data bit interval. Messages of up to 64 bits are assembled vertically in the call store by the program with one bit in each of 64 words in preparation for sending. Data senders, located in separate frames from the control unit, receive message bits over private transformer-coupled twisted pair leads. Data receivers would also be located in separate frames from the control unit. Information received is stored in shift registers which are interrogated by a program using a ferrod scanner.

This method avoids the need for buffering outgoing data in shift registers. Incoming data are buffered in shift registers because they cannot be sychronized easily with the timing in the input-output control. In general, the quantity of incoming data is small so that the register size and program interrogation rate are also small.

The outgoing data bits are buffered at the transmitter and the 1.25 millisecond data timing signal generated in the input-output control is used to gate from the buffer to the transmitter itself. When the input-output control has sent the last bit of each message, it will set a bit in the input-output control register that stops data sending. New messages can then be stored in this area by the program. In order to start sending, the program sets the bit address counter in the input-output control for the address of the first message bits and resets the inhibit in the control register.

### 5.3.3 *Line Origination Scanning*

The autonomous input-output control performs the detection of line service requests or originations by sequencing through the ferrod scan points associated with customer lines and searching for an off-hook state. Once an off-hook state is encountered, the wired logic will stop sequencing and set a flag for the program. The identity of the scanner row can be determined by the contents of the enable address register. A program which is executed in the input-output 25 interrupt will perform a directed scan of the row, 50 milliseconds after storing the row identity in the call store and restarting the autonomous line scan function. Thus, some hit protection is provided.

The directed scan will result in the scan point number of the off-hook line being passed onto the main program for processing which includes the disconnection of the line ferrod. Thus, only unserved off-hooks are seen by the autonomous wired logic. Unless stopped by an off-hook signal, the wired logic continues to scan until the end of a scan block is reached at which time the program must provide a new starting address. The scan block can consist of a maximum of 512 consecutive addresses or 8192 lines. There can be a number of blocks and the blocks need not be maximum as long as they are continuous to the end of block boundary.

The line scanning function uses the central pulse distributor and peripheral unit buses in order to interrogate the ferrod scanners. The call store is not used since last look is not required to detect the simple off-hook state. The autonomous digit scanning of originating registers competes with line scanning for peripheral unit access but each has its own register access to the central pulse distributor and peripheral unit address bus. However, the program instructions use the same access register (enable address) as used by line scanning and, therefore, will interrupt the line scanning function (saving and restoring its state) when program access to the peripheral units is required. This access is mainly in the input-output 25 interrupt program so that the line scanner need only be stopped at the beginning of the input-output 25 program and restored at the end. The common and exclusive access provides simplicity and economy as opposed to a cycle-stealing mode which would provide less interference to the line scanning function. The time remaining outside of the input-output 25 interrupt is more than sufficient to allow the wired logic to scan the maximum number of lines at an adequate rate.

5.4 *Program Controlled Input-Output Orders*

Program access to the central pulse distributor, peripheral unit address, and scan answer buses is required for control of, and communication with, the peripheral units. General purpose instructions as well as special purpose orders, that are designed to conserve real time when repetitive access is required, are provided. The peripheral unit order structure is compatible with No. 1 ESS frames and additional orders are included to provide faster and more efficient communication with serial receivers of central pulse distributor signals such as the No. 2 ESS network frames and the peripheral decoder.

The No. 1 ESS frames require information on the peripheral unit address bus in one-out-of-*N* form to avoid the replication of translators in the peripheral frames and to check for errors in the received information. In No. 2 ESS, wired translators for access to the peripheral unit address bus are generally not provided except where real-time considerations are paramount, as for the ferrod scanner. The scanners, for example, require two groups on one-out-of-eight to select one of 64 scanner rows.

A scan order is provided which activates the central pulse distributor based on the high ten bits of the enable address register, translates the low six bits into two one-out-of-eight groups pulsing them onto the peripheral unit address bus, receives a scanner reply from the scan answer bus, and places the result in the logic register. For general use, an order is provided which allows the contents of the logic register, general register, and the low six bits of the enable address to be pulsed directly onto the peripheral unit address bus. In this way, the format of the information being sent is controlled by the program and translated using the general purpose program instructions. Economy and flexibility is thereby gained at the expense of real time. In addition, as mentioned in the central pulse distributor description (Section 5.1.2), an order is provided for pulsing central pulse distributor points, either those contained in the local distributor or those in a supplementary distributor.

The scan order just described is convenient for handling a single row of ferrods. Another typical call processing function would be to sequence through a number of scanner rows, comparing them with last-look bits in call store and indicating when a change is detected. In addition, it is desirable to sometimes mask out the status of certain ferrods for maintenance reasons or, for example, when the ferrod

status definition changes during the course of a call, as for the by-link trunk. The by-link trunk ferrod provides supervisory status as well as dial pulse information; therefore, the supervisory scan program must ignore the status of the ferrod when dialing is in progress (the ferrod is then being scanned by an originating register). To accommodate this need, a macroinstruction is provided which uses the enable address register to address the scanner, a call store address register (CA) to direct the reading of status bits, and optionally a second call store address register (AA) to direct the reading of a mask. These are combined in a logic function which causes the instruction to terminate if the condition is met or else repeats after incrementing the enable, call, and addition address registers until the cycles as specified by the kall record counter are completed.

Additional flexibility is provided by options in the instruction, which determines the increment for the enable address and call store registers.

Two other macroinstructions are provided, both of which are designed for serial communication over central pulse distributor pairs. The bit pattern in the logic register is shifted with each instruction loop, the low bit controlling the polarity of the central pulse distributor pulse over the pair defined by the enable address register. Typical central pulse distributor action requires four machine cycles in order to include the various checks that are performed during, and as a result of, a central pulse distributor order. However, after sending an order with each polarity which includes all central pulse distributor checks, the remaining bits can be sent at a rate of one cycle per bit by excluding further checks and reducing the pulse width to improve transformer recovery. Thus, one macroinstruction transmits a bit per four cycles with central pulse distributor checks and the other instruction transmits a bit per cycle with no distributor checks. The faster order is restricted to the local central pulse distributor and is used for sending network orders and orders to peripheral decoders controlling high usage service circuits such as multifrequency transmitters.

### 5.5 *Input-Output Error*

In sending an order to the periphery, the input-output control performs error checks that are designed to detect faults in itself, in the buses, and in the peripheral units. The reaction to an input-output error is not like other processor error conditions, which simply switch processors, since the location of the fault is not known. Here a special

program must be summoned to isolate the trouble and establish a working mode.

5.5.1 *Check Circuits*

There are three input-output errors that can result from a peripheral action: (*i*) central pulse distributor error, (*ii*) scanner ALL-SEEMS-WELL error, and (*iii*) enable verify error.

The central pulse distributor contains check circuits which monitor the enable access circuitry to see that one and only one access switch in each of the three groups is enabled. The magnitude of the current pulse into the matrix is monitored to determine that it is not excessive and the pulse in the primary leg of the final transformer stage is monitored to determine that its current magnitude is sufficient. These checks encompass many faults such as pulser failures, open transformers, shorted diodes in the matrix, shorted output pairs, translator failures, and so on. The supplementary central pulse distributor performs similar checks and replies to the input-output control with a central pulse distributor ALL SEEMS WELL signal when all checks pass. Absence of the ALL SEEMS WELL will cause the input-output control to indicate a central pulse distributor error, just as in the case of the local central pulse distributor.

The presence of a scanner ALL SEEMS WELL is expected with all scan orders. The scanner checks the validity of the information it receives, monitors its own operation, and replies with an ALL SEEMS WELL signal when the operation appears to be correct. The input-output control will indicate an ALL SEEMS WELL error when the reply is missing.

The enable verify check is intended to determine whether or not the frame received an enable signal. The input-output control checks for the enable verify reply on a common bus similar to the ALL SEEMS WELL bus, both of which are considered members of the scan answer bus.

5.5.2 *Reaction to Errors*

With the occurrence of any of these errors, the input-output control reacts differently depending upon whether the error was produced by a program order, an autonomous line scan order, or an autonomous digit scan order. Program orders encountering input-output errors will set the condition flip-flop and terminate the order if it is a macrotype. The condition flip-flop is investigated after all input-output program orders to determine whether to go to an appropriate working mode program.

An autonomous line scan order encountering an error will result in stopping the line scanner as though an origination was detected. Subsequent rescan by the program will result in the error being treated as a program scan order. The autonomous digit scan error results in more immediate action since digits can be lost or the call store mutilated. The error in this case causes the input-output control to freeze its timing and generate a high priority maintenance interrupt. This also results in a working mode program but, in this case, the identity of the trouble producing order is contained in the input-output digit logic. In all cases, the working mode program seeks to isolate the fault and might result in a reconfiguration of peripheral controllers or central processors.

## VI. MAINTENANCE AND ADMINISTRATION CENTER

### 6.1 *Introduction*

In order to maintain and administer the system, a means must be provided to allow the maintenance man to communicate with the system. The maintenance and administration center provides that means. Through it, the man may request such functions as system status printouts, diagnostics, verification of translation data, or updating of magnet cards. The goal in designing the center was to build a fairly compact, highly intergrated center which is closely tied into the central processor, rather than build several pieces of test equipment to do specific jobs. The center is a useful tool in factory testing, installation, system maintenance, and laboratory debugging.

The center includes the maintenance center, teletypewriter control circuits, maintenance teletypewriter, single card writer, and trunk test panel.

The first four units occupy one double bay frame, and the trunk test panel occupies a single bay frame. Reference 1 has photographs of these frames. The maintenance teletypewriter is in the center bay directly below the maintenance center panel. The bay to the left contains the single card writer and maintenance center logic. The trunk test panel is in the bay to the right. Such proximity requires little movement of the maintenance man to perform all the functions.

### 6.2 *Maintenance Center*

#### 6.2.1 *Introduction*

The maintenance center consists of several almost independent functional blocks which can be interconnected in different configura-

tions to accomplish the desired functions. These blocks are: a display buffer, internal registers for status display and control, a comparator, a central processor stop mechanism, a central processor interrupt mechanism, the control unit control circuit, a margin control circuit, and the emergency action panel.

The display buffer and three other internal maintenance center registers are connected to a maintenance center gating bus using the same concept as the program gating bus in the control unit. Also, coupled to the maintenance center gating bus by dc communications are the program gating bus, program address register, call store input register, and call store address register of each control unit. The display buffer can be used to automatically display the contents of the program address register, call store address register, program store, or call store, or it can be gated to by program. The display buffer can be "frozen" by hardware or program to permit an address to be trapped, or data to be displayed without being overwritten. The display buffer is shown on the maintenance center panel, Fig. 13, as a 22-bit register, marked in various ways to facilitate reading the information it contains. The lighted pushbuttons, which can be used to set the individual bits, are colored in groups of three for easy reading as octal numbers. Omitting bits 21 and 0, which indicate the transfer allow check bit and parity bit, the remaining 20 bits are indicated in groups of five by a white background mask for easy reading as one 5-bit or two 10-bit operation codes. The 5-bit groups are further divided to indicate the octal breakdown of the operation code.

The comparator will match two inputs of up to 22 bits each, with both a match and mismatch output. Inputs are from the maintenance center gating bus, the comparator input switches, and call store input register of each control unit. The comparator output can be used to trigger one of several functions, allowing the craftsman to capture events or conditions such as a register equal to a given value, or a logic gate becoming active. The two most common uses are to trigger the stop or interrupt mechanisms.

The main application for which the maintenance center was designed is as a system maintenance tool in an operating office. In this capacity, the maintenance center serves in four roles:

(*i*) An in-service monitor of the status of the system,
(*ii*) The test and control center for routine functions,
(*iii*) A backup means when planned diagnostic procedures fail, and
(*iv*) An emergency control in extreme situations.

As an installation tool, the maintenance center takes the place of

external test equipment in previous systems. With the maintenance center, it is possible to manually control, access, and monitor the control unit thus insuring the system's proper manufacture and installation. Its use as an installation tool (as well as at the factory) is in close harmony with the test specifications and initial test (X-ray) programs.

Also it is an aid in program development in the system laboratory. In this capacity, an individual programmer is permitted manual access to most control unit registers as well to the control logic.

The usefulness of the maintenance center in each of these applications can best be explained by describing its separate features.

### 6.2.2 *Operational Features*

The features of the maintenance center fall into three categories: (*i*) control until matching, (*ii*) routine operation, and (*iii*) special purpose operation.

Under certain conditions (explained later), the maintenance center

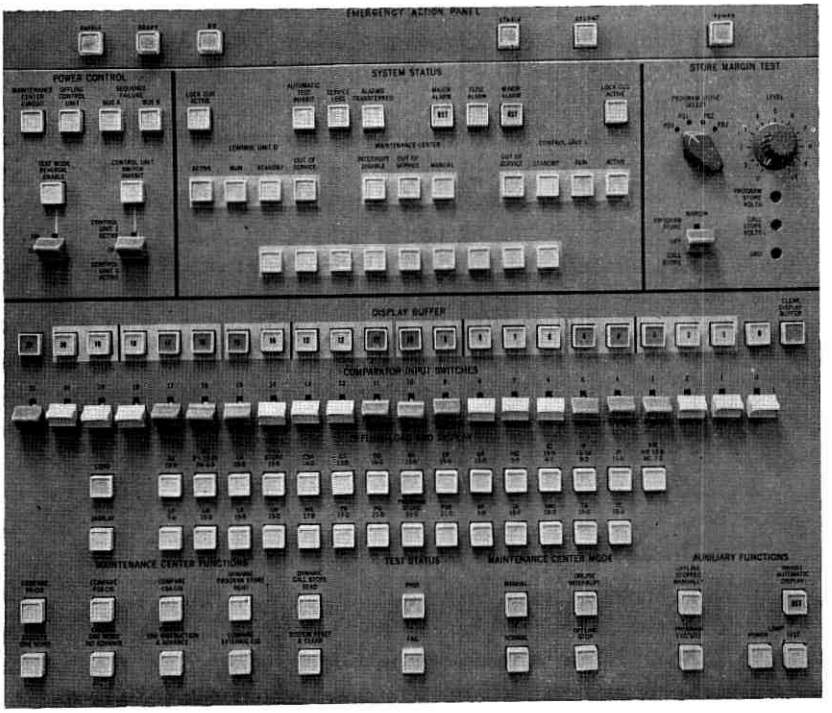



Fig. 13 — Maintenance center panel.

acts as a master check circuit, matching the results of the duplicated control units. Routine operations include those automatic or manual features which are used or observed routinely by the office personnel, or operated automatically by the system. The third category includes those features which are used when manual intervention is required by the office personnel, usually in a nonroutine capacity.

The maintenance center has two operational modes, NORMAL and MANUAL. The MANUAL mode is required in most cases of human intervention; however, this disables certain automatic features which are performed when the center is in the NORMAL mode. Use of the MANUAL mode does not necessarily imply an abnormal or emergency situation. Several day-to-day operations require the MANUAL mode because they use circuitry used for other purposes in the NORMAL mode.

### 6.2.3 *Control Unit Matching*

Without the intervention of a maintenance man or some detected trouble, both control units are usually running simultaneously, executing the same instructions in synchronism, with only the outputs of the off-line control unit to the periphery and to the maintenance center being inhibited. If the maintenance center is in the NORMAL mode, this state is said to be the NORMAL-COMPARE state, which enables the processor match circuit. In this mode, the contents of the call store input register in each control unit are being matched in the maintenance center. Since the call store is used quite frequently to store temporary results by the program (the call store input is used on both read and write), any trouble that is not detected by check circuits should cause the call store inputs to differ within a short time. This mismatch causes an interrupt signal to be sent to both control units, activating a mismatch detection monitor program in an attempt to locate the trouble. When the mismatch occurs, the on-line call store address register is gated to the display buffer, then "frozen." This aids the maintenance programs in the detection and recovery process.

Whenever the maintenance center is in the NORMAL-COMPARE mode and no mismatches have occurred, the current program address is shown in the display buffer. In the event of an automatic switch of the control units, the program address location at the time of the switch is trapped in the display buffer for use by the maintenance programs.

### 6.2.4 *Routine Operations*

Features that are part of routine operation are the system status displays, trouble recording, and control of the on-line—off-line status

of the control units. These features are active regardless of the MANUAL or NORMAL mode of the maintenance center. A fourth use of the maintenance center in routine operation is in conjunction with the magnet card updating and writing process using the single card writer. This use is explained later.

As can be seen from the photograph of the maintenance center panel (Fig. 13), the system status displays indicate the current alarm, control unit, and maintenance center status, as well as the status of the maintenance and call processing program capabilities. A line of lights at the bottom of the system status display area is permanently assigned by the particular application of the No. 2 ESS. These indicate status of unique programs or devices (for example, certain peripheral units).

In addition to these status indicators, the maintenance center contains a register which monitors the control unit check circuits. This register is called the error register.

The on-line—off-line status of the control units is normally determined by a flip-flop in the maintenance center, as controlled by the program timer or the maintenance programs. In emergency situations it can be controlled by the craftsman.

#### 6.2.5 *Special Purpose Operation*

The special purpose operations include most of the manually operated functions, except for card writing, and require the maintenance center to be in the MANUAL mode. They are used mainly for installation, program debugging, and diagnostic work. Probably the most useful of these features are the "compare" and "load and display" features. The compare feature uses the same comparator as the processor matching operation already described. Inputs to this comparator allow matching the following items:

(*i*) program address versus a set of comparator input switches on the panel,

(*ii*) call store address versus the comparator input switches,

(*iii*) program address versus the display buffer (can only be set up up by program),

(*iv*) program store output buffer versus the comparator input switches,

(*v*) any external signal (up to 22 bits) versus the comparator input switches.

When a match occurs, the match signal may be used for the following, depending upon the function and upon the control unit selected (on-line or off-line). It may cause:

(*i*) an interrupt in the on-line control unit whenever any of the above matches occur,

(*ii*) the off-line control unit to be stopped whenever any of the above matches occur,

(*iii*) the on-line or off-line program store contents to be displayed in the display buffer whenever the program address matches the comparator input switches (dynamic program store display), and

(*iv*) the on-line or off-line call store contents to be displayed in the display buffer whenever the call store is accessed and the call store address matches the comparator input switches (dynamic call store display.)

If the off-line control unit has been stopped by a match, the display buffer, which has been displaying the program address, is "frozen" and contains the last program address, thus effecting a "from address trap." This is most useful in tracing the progress through a program. After the desired operations have been performed, the control unit can be started again and allowed to run in any of the following modes: (*i*) continuously (remove the match function), (*ii*) up to another match, which could have been changed while the control unit was stopped, or (*iii*) for only one instruction.

The incrementing of the program address register can also be inhibited, allowing repetition of the same word, such as when it is necessary to look at timing signals with an oscilloscope. Two modes are available:

(*i*) execute one word (one or two instructions) continuously, and
(*ii*) execute one word at a time.

When the off-line control unit is stopped (on a match as above, or by depressing a pushbutton on the panel), several avenues of investigation are open to the maintenance man. The first is the ability to load or display any register in the stopped off-line control unit. This is accomplished by first selecting a LOAD or DISPLAY function, then selecting the appropriate register from the set of pushbuttons (see Fig. 13). Registers are displayed in the display buffer, and are loaded from the display buffer, which itself can be preset from the maintenance center panel, if desired. Call store can be read or written and program store can be read using the same mechanism, if the appropriate address register is loaded first.

Frequently it is desirable to be able to look at or trap conditions in the on-line processor without disturbing its call processing operation. For this purpose, the on-line processor may be interrupted (whereas

the off-line processor was stopped). The interrupt is used to trigger one of a set of utility programs which have been previously requested via the maintenance teletypewriter. Among the utility routines are the ability to do the following:

(*i*) read or write any on-line or off-line call store word,
(*ii*) read any on-line or off-line program store word, or
(*iii*) read or load most on-line or off-line control unit registers.

Contents of the program store, call store, or registers can be typed out on the teletypewriter either on a one-shot basis or whenever the data changes, or can be displayed dynamically in the display buffer. The program can turn off the interrupt, as in the case of a request for a single printout, or when the teletypewriter printout buffer area in the call store becomes full. Only one interrupt request can be active in the system at a time.

The above utility requests may also be base level requests. A base level request asks that the function be performed immediately upon receipt of the input message, or once per base level scan if a repetitive function is desired. These base level requests are recognized regardless of the mode of the maintenance center, and can be acted upon simultaneously with an interrupt request.

The remaining special purpose features are the store margin control, power control, and emergency action control. The store margin control allows margins to be checked manually on any one of the four program stores or on the call store, with the threshold voltage of the sense amplifier available at test points on the maintenance center panel. (This is independent of the preset automatic margin tests controllable by program.) The power control provides power on-off functions and a system override control which allows the flip-flop which controls the active status of the control units (program control flip-flop) to be bypassed. The override is to be used by the maintenance man only in emergencies, such as repairing the program control flip-flop. Normally, a control unit can be locked active by depressing the LOCK CU—ACTIVE pushbutton, which does not bypass the flip-flop but still holds that control unit active. The emergency action control allows the maintenance man to call in an initialization program manually. This program will initialize the control unit and, depending upon the status of two associated pushbuttons (STABLE and RECENT), zero the nonstable areas of call store, and neither, either, or both of two areas of call store: (*i*) recent change buffer area and (*ii*) stable call store (all except recent change).

### 6.3 *Teletypewriter Facilities*

The teletypewriter facilities contained in the maintenance and administration center include the teletypewriter control circuits, as well as the maintenance teletypewriter. Up to eight teletypewriter channels can be accommodated by the No. 2 ESS, with a minimum of two channels required. Each channel is controlled by one control circuit. Any standard teletype device can be attached to a channel, either singly or in combination, such as a send-receive teletypewriter, receive only teletypewriter, paper tape punch, paper tape reader, or automatic send-receive teletypewriter. Long distance channels can be handled with a 108/820 data set combination over a private line.

Each channel control circuit is an independent parallel-to-serial converter, accepting the 7-bit teletypewriter code from the program, and pulsing out the eleven 9.09 millisecond pulses necessary to control the teletype device. Data communication to the circuit is over a dc connection from the program gating bus, with the individual circuit selected by one of eight central pulse distributor points. When receiving information from the channel, the circuit acts as a serial-to-parallel converter with communication from the circuit to the program over a set of ferrods. There are seven data ferrods, one signal present ferrod, and several status ferrods—low paper, teletypewriter disconnect, data set disconnect, and so on—depending on the particular channel.

The layout of the control circuit consists of one common tray of logic providing timing selection (the on-line clock is used), power-off relays and data selection. The individual circuits grow in modules of two circuits per logic tray, but only one set of cards needs to be supplied at a time. The circuit trays can be added "in service" as needed.

The teletypewriter program routes messages to, and accepts messages from, the correct channel depending upon the content of the message. If that channel is out of service or does not exist, the message is routed to or accepted from its backup channel designated by the operating company. No backup can be designated if so desired.

A maintenance detection check is made on each output message to guarantee the communication loop. This check consists of sending a "who are you" code to the teletypewriter which responds with a fixed answer. If the answer is not received, the message is routed to the backup, and diagnostics are run on the original channel. Diagnostics and restoration can also be requested manually.

### 6.4 *Single Card Writer*

#### 6.4.1 *Application*

The single card writer is the device by which permanent magnet twistor cards are magnetized in the smaller No. 2 ESS offices. As the name implies, it magnetizes one card at a time, rather than an entire module as does the No. 1 ESS card writer. The single card writer has been designed as simply as possible to meet moderate change activity requirements and still keep the cost low so as to be attractive in the smallest No. 2 ESS offices.

#### 6.4.2 *Procedure*

When a card is to be written, a message is typed on the teletypewriter indicating the card to be magnetized, or asking for the next card to be magnetized. Upon the acknowledgement, the card is placed in the single card writer and the WRITE button depressed. This starts the magnet head moving and saturates a ferrod causing the program to gate 22 bits of information to the display buffer. The program notifies the single card writer via a central pulse distributor point that the display buffer is loaded. The polarity of the pulse is checked against the card type (A or B) which is in the writer, and a WRONG light indicates an incorrect card type.

When the single card writer head senses the first initializing magnet, slightly before the first row of magnets, it takes the low 11 bits from the display buffer. When the second initializing magnet is sensed, the high 11 bits from the DB are magnetized into the low 11 bits of the second row of magnets and the single card writer asks for more information. This sequence is repeated 32 times in all, with 33 milliseconds between each word request, then an end-of-row ferrod is saturated. The program checks that the proper timing exists between the 32nd word request and the end-of-row ferrod, and that 32 requests have been received. The process is then repeated for the next three passes. If the entire sequence is completed satisfactorily, an END light is operated.

The procedure for magnetizing a set of cards involves pulling the desired cards from a spare set, magnetizing them, placing them in the off-line store, and verifying the magnetization. This is then repeated with the on-line control unit switched off-line. The time required to magnetize 37 cards on both sides has been demonstrated to be one hour.

### 6.5 *Trunk Test Panel*

Manual testing of lines and trunks are an important part of every office. In the No. 2 ESS, most line testing will be accomplished from a standard No. 3 local test cabinet. This provides for continuity, leakage, foreign potential, and subscriber line checks. All manual trunk testing, as well as transmission measurements on lines, will be conducted from the trunk test panel.

The trunk test panel will occupy a bay adjacent to the maintenance center frame. This facility provides for switched-up access to any trunk, service circuit, or line terminated in the No. 2 office. It enables the following checks to be performed:

(*i*) operational and transmission tests on trunks and service circuits,
(*ii*) leakage and continuity checks on both lines and trunks,
(*iii*) transmission checks on subscriber lines and PBX-type trunks,
(*iv*) removal of trunk from service and restoral to service under key control, and
(*v*) voice communication via private or regular telephone channels.

Test connections are made via one of three access trunks established under key control and with the aid of a panel mounted *Touch-Tone*® telephone. The access trunks allow for:

(*i*) originating and terminating service,
(*ii*) unrestricted test access to all circuits, even under line load control and busy conditions,
(*iii*) connection, under key control, to a voltmeter test circuit, transmission and signaling test circuitry, and make-busy arrangements,
(*iv*) jack access to allow any portable instrument connection,
(*v*) holding a circuit while testing another, and
(*vi*) the ability to place a trunk in any of its operational states.

## VII. SUMMARY

This article has described the structure and relationship of the units which unite to form the major system control element, the No. 2 ESS control unit. The details of the implementation of the logic and circuit designs have been avoided but, instead, the general structure and features of the system have been stressed. The control unit design is a product of the needs for an economic and efficient stored program control system for a particular class of offices. The influences of both the call processing design and the maintenance plan are evident when

viewing the details of the system. Repetition and overlap between companion articles in this issue has been avoided whenever possible. As a result, further details and philosophy concerning the control unit design are in the associated articles, especially those concerned with call processing and maintenance.

REFERENCES

1. Lonnquist, C. W., Manganello, J. C., Skinner, R. S., Skubiak, M. T., and Wadsworth, D. J., "Apparatus and Equipment," B.S.T.J., this issue, pp. 2817–2863.
2. Andrews, R. J., Driscoll, J. J., Herndon, J. A., Richards, P. C., and Roberts, L. R., "Service Features and Call Processing Plan," B.S.T.J., this issue, pp. 2713–2764
3. Digrindakis, Jr., Freimanis, L., Hofmann, H. R., and Taylor, R. G., "Peripheral System," B.S.T.J., this issue, pp. 2669–2712.
4. Meinken, R. H., "Ferrite Sheet Memory," Proc. 1960 Elec. Components Conf., Washington, D. C., May 10–12, 1960.
5. Meinken, R. H., "A Memory Array in a Sheet of Ferrite," Proc. Magnetism and Magnetic Materials Conf., October 16–18, 1956, p. 674.
6. 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., *43*, No. 5, Part 1 (September 1964), pp. 2147–2191.
7. Ault, C. F., Gallaher, L. E., Greenwood, T. S., and Koehler, D. C., "No. 1 ESS Program Store," B.S.T.J., *43*, No. 5, Part 1 (September 1964), pp. 2097–2146.
8. Connell, J. B., Hussey, L. W., and Ketchledge, R. W., "No. 1 ESS Bus System," B.S.T.J., *43*, No. 5, Part 1 (September 1964), pp. 2043–2045.
9. Freimanis, L., Guercio, A. M., and May, H. F., "No. 1 ESS Scanner, Signal Distributor, and Central Pulse Distributor," B.S.T.J., *43*, No. 5, Part 2 (September 1964), pp. 2270–2277.

# Peripheral System

By JOHN DIGRINDAKIS, LAIMONS FREIMANIS, H. ROBERT HOFMANN and ROBERT G. TAYLOR

(Manuscript received March 6, 1969)

*The peripheral units of No. 2 ESS, described in this article, include the scanners, the network, the peripheral decoders and the trunks and service circuits. The scanners serve to collect information for the control complex. The network is a folded, space division network using ferreed switches. The peripheral decoder uses integrated circuits to control relays in user circuits as directed by the control complex. The program controlled trunks and service circuits provide an interface between the outside world and the remainder of the No. 2 ESS.*

## I. INTRODUCTION

Other articles in this issue describe the overall system organization and the processor complex with its busing and pulse distributing facilities.[1, 2] This article describes the major functional peripheral blocks of the system: the scanners, the network, the peripheral decoders and the trunks and service circuits.

## II. SCANNERS

Electronic scanners may be broadly defined as circuits which sense or detect the absence or presence of voltage or current. In No. 2 ESS, scanners are used to detect the on-hook, off-hook status of a customer's line, to check the status of talking paths for flash and disconnect, to monitor certain test points in various frames and to scan other miscellaneous points about which information is desired. In a certain sense, the scanners may be thought of as the primary source of information to the control complex regarding the actual physical state of the customers and circuits associated with the outside world.

### 2.1 *Ferrods*

The ferrod is the sensing element used in No. 2 ESS.[3] A single ferrod consists of a ferrite rod or stick, approximately the size of a large paper match, around which is wound a pair of solenoidal control coils. Threaded through two holes in the center of the ferrite stick are hair-pin single-turn interrogate and single-turn readout windings. Figure 1 is a diagram of a ferrod.

To determine the state of the ferrod, a 0.5 amp bipolar pulse is applied to the interrogate winding. The positive half cycle, approximately 3 microseconds long, switches the ferrite around the holes, as

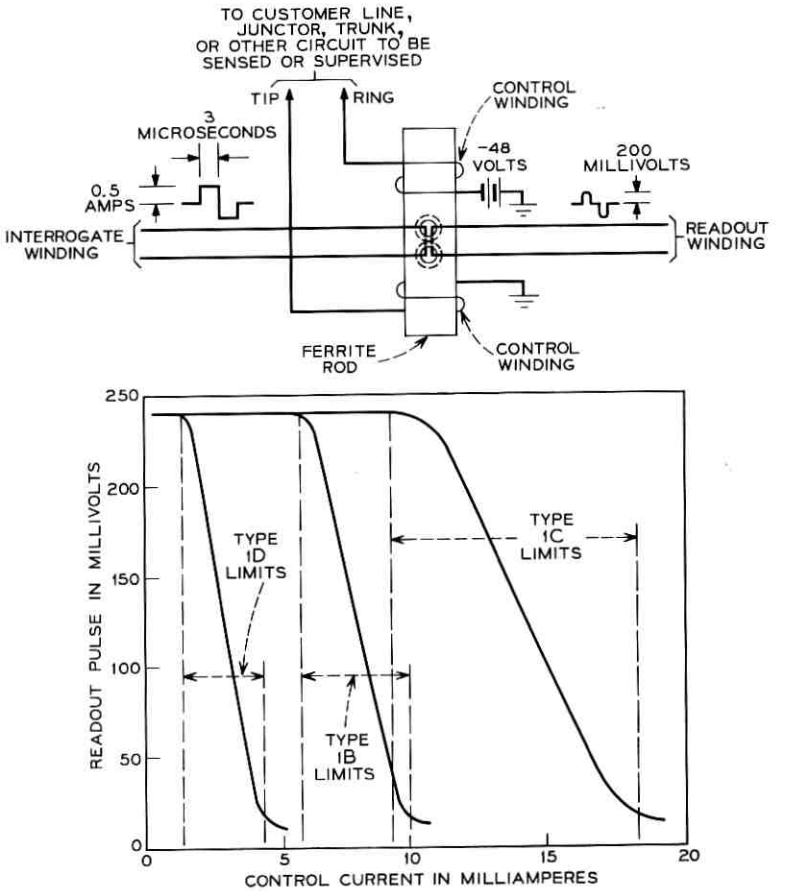



Fig. 1 — Ferrod schematic diagram and typical characteristics.

two small cores might be switched, and causes a voltage to be induced in the readout winding. This voltage is nominally 200 millivolts as read using an integrating circuit with a 1 microsecond time constant. The negative (second) half of the current pulse is used to reset the ferrite in the switched area. As current through the control windings is increased from zero, the ferrite stick becomes saturated, the switchable area of the "cores" decreases, and the output decreases. When the current level in the control windings completely saturates the ferrite stick, there is no ferrite switched and the output is less than 20 millivolts.

### 2.2 *Ferrod Types*

There are three types of ferrods used in No. 2 ESS. All operate as described above, and differ only in their sensitivity and winding resistance which are listed in Table I.

1B ferrods are used in the line scanner where they apply battery and ground to loop start lines. For ground start applications, such as coin or PBX lines, the two control coils are placed in series and supply battery voltage to the line. The 1B ferrod is wound with resistance wire which limits current flow in the presence of an accidental ground. The 1B ferrods are connected to the subscriber lines through contacts in the switching network so that they may be removed from the line during talking, because they would otherwise shunt the line and impair transmission.

The 1C and 1D ferrod assemblies, each consisting of two ferrods, are used for monitoring trunk and junctor circuits. The 1C ferrod is placed in series with the battery and ground leads to the junctor and to

TABLE I—ELECTRICAL SPECIFICATIONS OF FOUR FERROD SENSOR CODES

| Ferrod Sensor | 1B | 1C | 1D<br>1E* |
|---|---|---|---|
| Number of windings per ferrod | 2 | 2 | 2 |
| Resistance per winding (±10%) | 660 | 19 | 35 |
| Turns per winding | 1600 | 930 | 1300 |
| Approximate inductance (mH, both windings) | 220 | 70 | 500 |
| Maximum current (mA) | 100 | 100 | 100 |
| Maximum unbalance between windings (ohms) | — | 1.0 | 1.0 |
| Nonoperate current (mA) | 5.5 | 9 | 1.8 |
| Operate current (mA) | 10 | 18 | 3.9 |

* The 1E ferrod is electrically identical to the 1D.

the customer side of trunk circuits. The 1D ferrod is used mainly on the distant office side of trunk circuits and in service circuits.

### 2.3 *Scanner Organization*

Groups of ferrods, controlled by a duplicated control circuit, make up a scanner. Each of the control circuits under the direction of the central processor can interrogate groups of 16 ferrods at a time, up to a maximum of 1,024 ferrods, at a rate of 16 ferrods every 12 microseconds.

The functional arrangement of a scanner is shown in Fig. 2. The 1,024 ferrods to be interrogated are arranged in a matrix consisting of 64 rows of 16 ferrods. To address a scanner, an enable pulse (0.5 microsecond) is sent to the desired controller from a central pulse distributor point. This pulse is stretched (2.5 microseconds) and causes the controller to read in parallel a 17-bit word (0.5 microsecond long) on the address bus from the active control unit into pulse-stretching address register circuits (3.8 microseconds). The 17-bit word contains two 1-out-of-8 selections that are used to pulse an $8 \times 8$ biased core matrix. The output from each core of the matrix is connected to interrogate one of the 64 rows of ferrods. The output of the ferrods are multipled along the columns and are connected to 16 duplicated readout amplifiers which in turn send back the status of the 16 ferrods to the central processor over the scanner answer bus (0.5 microsecond pulses). A 17th bus bit is used for maintenance purposes.

The leads going to the ferrod matrix from the core matrix of one controller are connected in series with the leads of the core matrix of the duplicate controller. In this manner, either controller may interrogate any of the 64 rows of ferrods. The 16 duplicated readout amplifiers return the ferrod status to the central processors over two identical buses. The duplication of the control and readout circuitry provides the two central processors with the means of scanning the ferrod matrix in the event that one of the processors or one of the scanner controllers is out of order.

### 2.4 *Types of Scanners*

Line scanners are used to detect customer line originations. In the network each control frame contains two complete 1,024 point scanners, except for the ferrods. The ferrods are provided as needed, in growth steps of 512, in the line-trunk switching frames.[4]

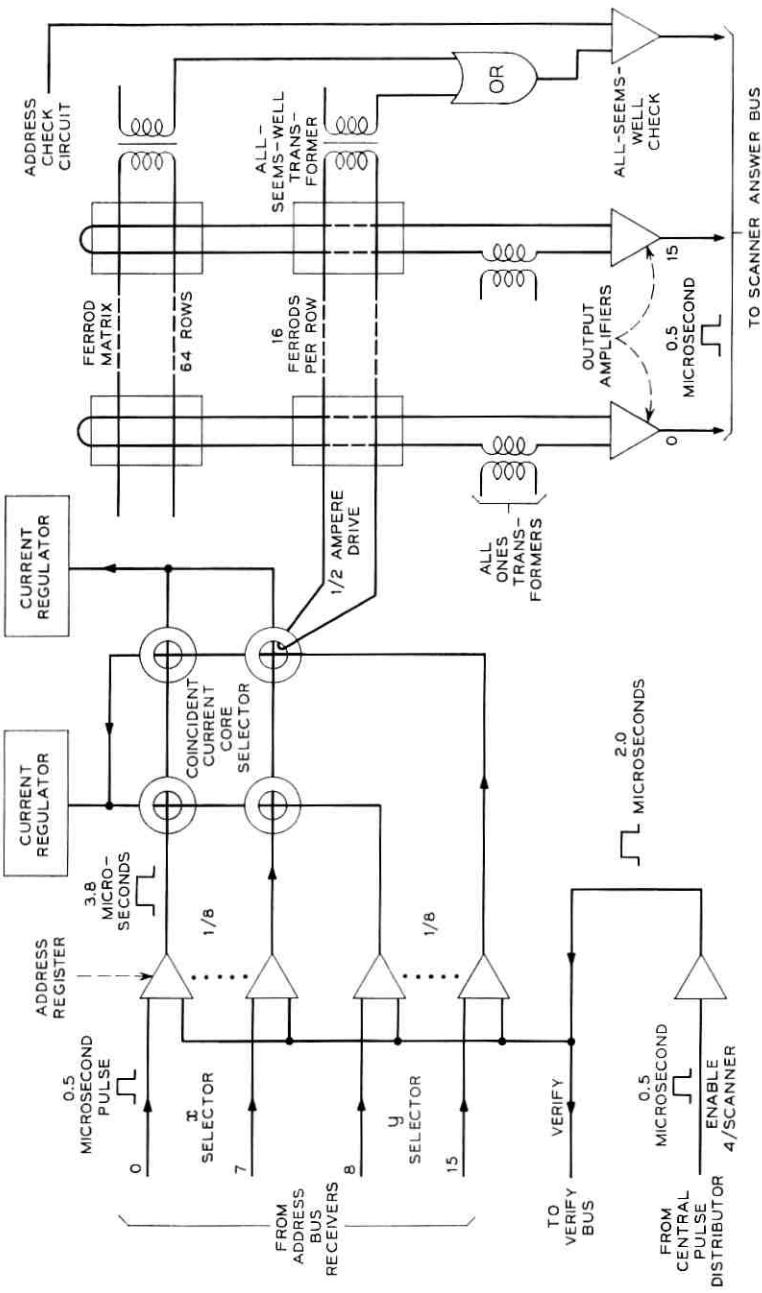



Fig. 2 — Functional diagram of a 1024-point scanner.

The master scanner is used for scanning miscellaneous trunks and service circuits and special scan points in various pieces of equipment around the office. The rate at which a row of ferrods is scanned in the master scanner is a function of the type of scan points contained in that row.

The universal trunk scanner is used for scanning trunk and junctor circuits mounted on the universal trunk and junctor frame. In most offices it also serves as a partial master scanner.

### 2.5 *Maintenance Features*

Each scanner controller contains several maintenance features designed to detect malfunctioning circuitry and inform the central processor when the information on the scanner answer bus is questionable. The enable pulse, which causes a controller to read the address information is also used to send an enable verify pulse back on the enable verify bus. Detectors are built into the core driver circuit in each controller to indicate that only one core drive horizontal, and one core drive vertical are pulsed. This will insure that only one row of ferrods is pulsed. In series with each ferrod row is a transformer, used to detect when that row of ferrods is pulsed. The outputs of all the transformers are ORed. A pulse on the output lead of the OR gate, along with the pulses indicating that only one horizontal and one vertical of the core matrix are being pulsed, will cause an ALL SEEMS WELL—SCANNER pulse to be transmitted on the seventeenth lead of the scanner answer bus. The absence of either of the three input signals will cause an ALL SEEMS WELL—SCANNER failure.

To determine that the output amplifiers are working correctly, provision is made to pulse all 16 outputs using a maintenance order. A seventeenth bit on the address bus is used, along with any address, to pulse 16 "all ones" transformers which are in series with each of the 16 ferrod readout loops. This simulates all ones to the readout amplifiers. The central processor expects to receive all ones and an ALL SEEMS WELL—SCANNER failure on this order.

## III. SWITCHING NETWORK

The No. 2 ESS network is a space division, two-wire network in which metallic connections are established through ferreed switches.* The combined line-trunk switching network has lines and trunks assigned to terminals at one side of the four-stage array with junctors

* The operation of ferreed switches is described in Refs. 5 and 6.

interconnecting the switches on the opposite side to form the folded eight-stage network (Fig. 3). By not having a separate line network and trunk network, "getting started" costs have been minimized. Up to fifteen networks can be interconnected to provide an ultimate traffic capacity of over 100,000 hundred-call-seconds.

To establish connections, the processor selects the desired network paths and sends peripheral orders to the network control circuits. These circuits in turn execute these orders to establish the specified paths operating the four stages on a cycle of 20 milliseconds between the receipt of an order and the end of the operation for simple one-part orders. Two-part four-stage orders, such as checking for foreign potentials and then setting up a connection, can be executed at a maximum rate of one every 40 milliseconds.

### 3.1 *Network Topology*

A line-trunk switching network is composed of from one to four line-trunk switching frames and a network control junctor switching

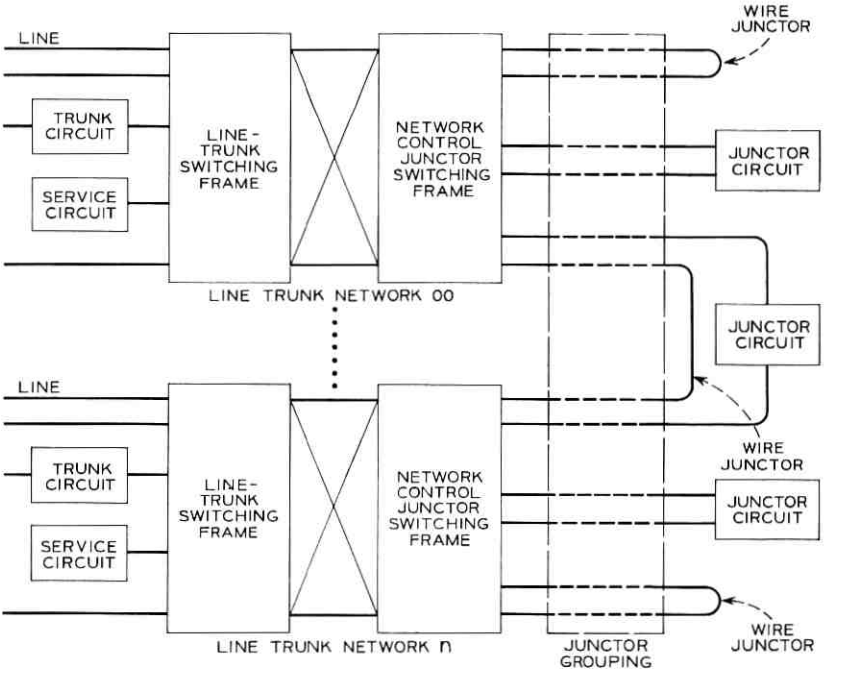



Fig. 3 — No. 2 ESS network diagram.

frame. The first two stages of the four stages of switching are located in the line-trunk switching frames. The third and fourth stages of switching are in the network control junctor switching frame. Concentration ratios of 2:1 and 4:1 are available. The concentration ratio is changed by the use of connectorized B-link multiples and equipping of line-trunk switching frames.[4]

Each line-trunk switching frame contains two concentrator groups, and each concentrator group contains eight line concentrators. A concentrator (Fig. 4) interconnects 32 lines and 16 B-links through its two stages. Thus, each concentrator group terminates 256 terminals on the input to the first stage switches and 128 B-links on the outputs

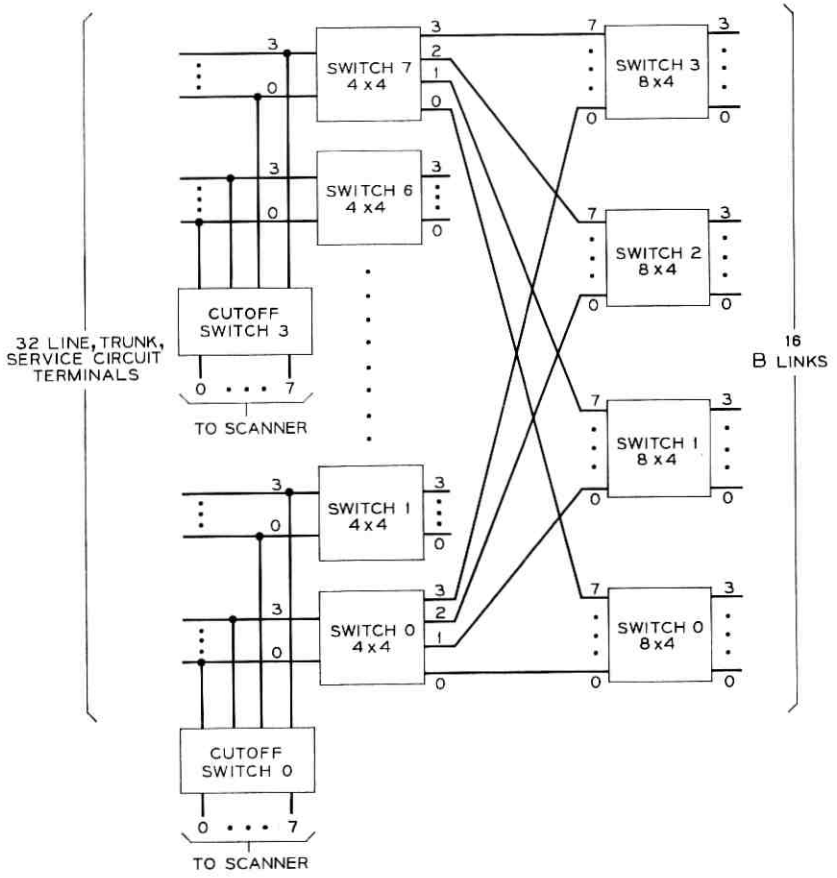



Fig. 4 — Tip-ring block diagram for 2:1 concentrator.

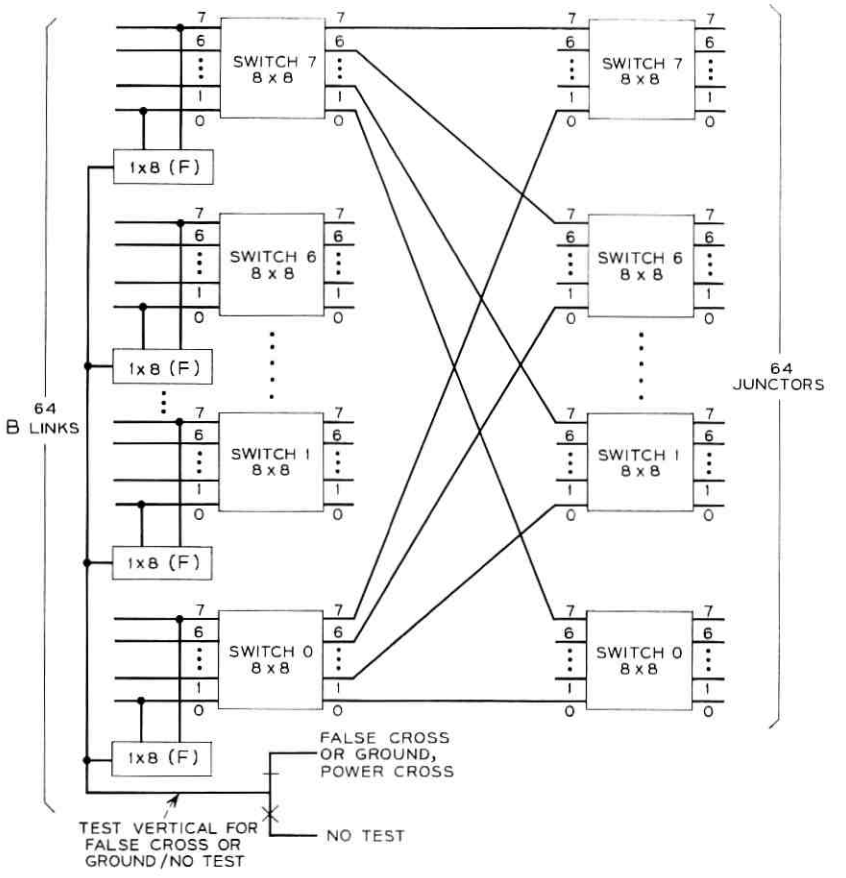



Fig. 5 — Tip-ring block diagram for junctor switches.

of the second stage switches. In addition to the crosspoint switches, each line-trunk switching frame contains one polar cutoff ferreed per input terminal to provide for scanner access.

The network control junctor switching frame contains eight grids, each of which is used to interconnect 64 B-links and 64 junctors through two switching stages (Fig. 5). The eight grids terminate 512 B-links on the input switches and 512 junctors on the output switches. There is also one polar ferreed per B-link, the F contact, for "test vertical" access.

The "test vertical" may be connected to either the no-test or to the

false cross or ground power cross circuits. The steering of the test vertical to either circuit is accomplished using two polar ferreeds connected in series with the coils in opposition. This permits opening the no-test contacts and closing the false cross or ground contacts with a pulse of one polarity, or the opposite with a pulse of reversed polarity. These two ferreeds are for each grid. The false cross or ground check is performed to check for internal network false crosses or grounds upon the establishing of new network paths. The power cross circuit is used to check for foreign potentials on customer lines. A bridged test connection may be made to any network path via the no-test contacts.

Eight grids are interconnected with four concentrator groups to form the fully equipped 2:1 network (Fig. 6). The addition of four more concentrator groups provides a 4:1 concentration ratio network (Fig. 7). The wiring pattern between concentrator groups and grids is such that two of the 16 outputs of a concentrator are wired to each of the eight grids. With this wiring pattern, each of the 1,024 terminals for the 2:1 concentration ratio (or each of the 2,048 terminals for the 4:1 concentration ratio) has access to each of the 512 junctors over either of two paths.

### 3.2 *Network Control Circuits*

The switching network is controlled by a pair of controllers, either of which can access the entire network (Fig. 8). Contacts on wire-spring relays are inserted in the control winding paths of every input, output, and intermediate link on both the concentrator groups and grids (Figs. 9 and 10). The interconnection of control windings between switches is parallel with the tip-ring interconnection pattern. The selection of a set of relays defines a unique control winding path.

A separate set of path selection relays is provided for each concentrator group and for each grid. The relays are double-wound so that they may be accessed by either controller. Under normal conditions, with both controllers in service, the controllers may operate simultaneously, provided that the two paths being connected are in different concentrator groups and in different grids.

#### 3.2.1 *Network Input Information to Input Register*

The input information received by the network controller from the central processor consists of path data and order data sent in the form of a bipolar pulse stream from the central pulse distributor. Figure 11 depicts the information sent to the controllers. The path data is re-

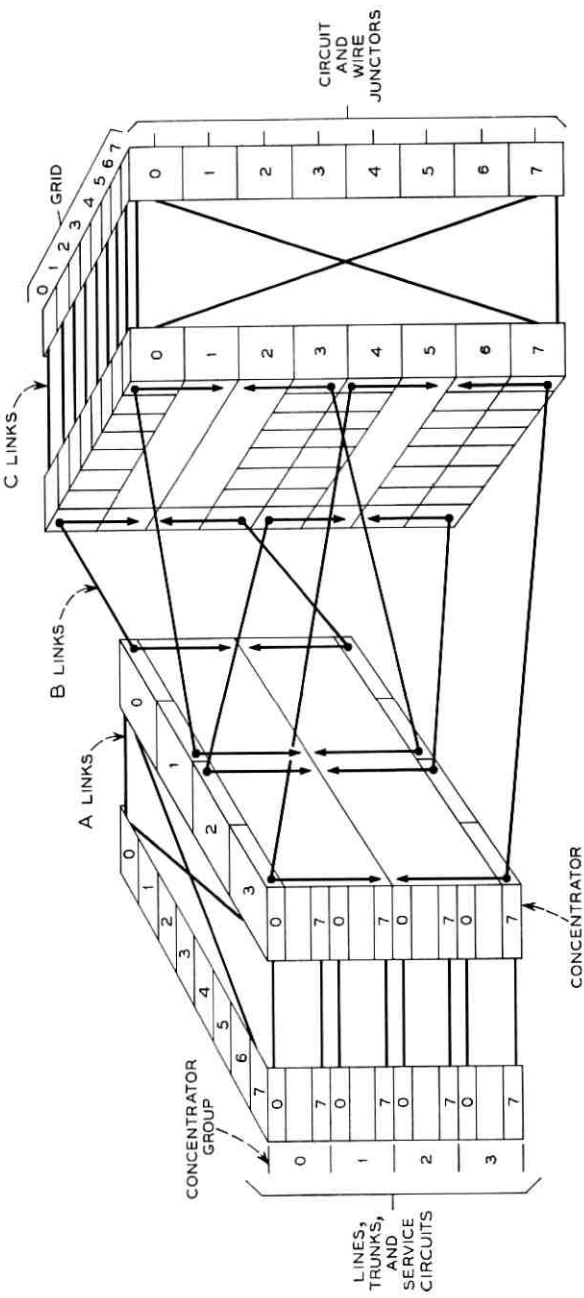



Fig. 6—2:1 concentration ratio line-trunk network fabric.

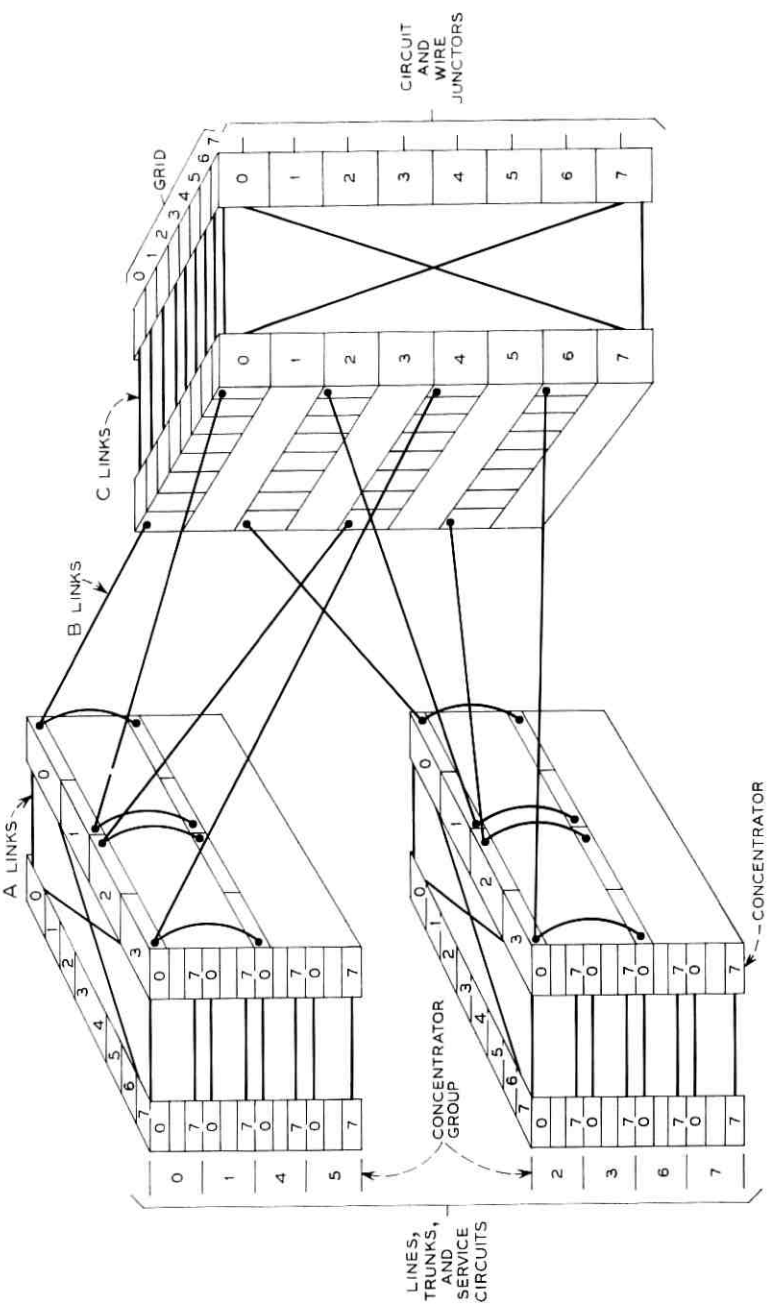



Fig. 7 — 4:1 concentration ratio line-trunk network fabric.

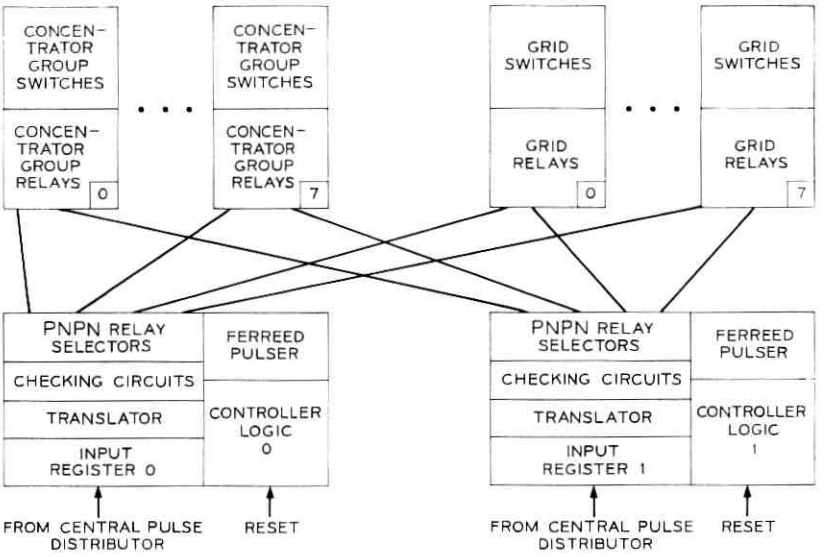



Fig. 8 — Network block diagram.

ceived in several subgroups. These groups contain information selecting the concentrator, the input switch and input level within the concentrator, the grid, and the output level and output switch within the grid. In addition, information is given as to which of the two B-links that interconnect these input and output points is the desired B-link. The information is provided in binary form and a parity check is made on the overall received information including the path, the order group data, and the start code. Parity will be odd, that is, an odd number of "1s" will be transmitted. By directly sending the input terminal identity and the junctor terminal identity in binary form, no translation of these quantities is required in the processor prior to sending a network order. Sixteen orders can be specified by the four order bits in the message. Thirteen are presently used. See Table II.

The bipolar pulses are received in a 28-bit integrated circuit shift register. The network responds with an "enable verify" pulse at the time that it receives the "first address bit" if that bit is a "one" and if no bits have been received since the controller finished its previous cycle or since the controller was reset using the external reset. The frame also responds with an "enable verify" to the central processor at the time that it has received all 28 bits and has checked that the parity and the start code of the information it has received is valid.

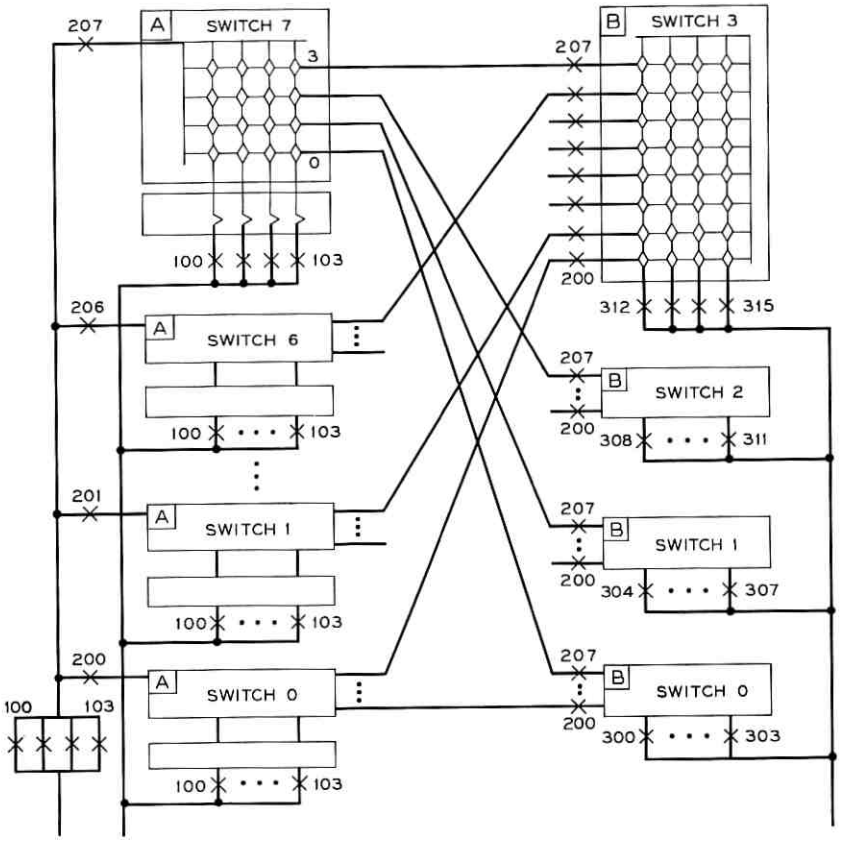



Fig. 9 — Ferreed control winding paths for concentrator 0. 100 series relays determine input level, 200 input switch, and 300 output level.

### 3.2.2 *Translators, Relays, and Control Logic*

The output of the 28-bit input register goes to two places in the network controller. The logical output that is used to generate the second "enable verify" is also used to send a start signal to the control logic portion of the network controller. The information concerning the input and the output points of the path goes to translators. The outputs of the translators are connected to PNPNS (silicon control rectifiers) which in turn operate wire-spring relays. The control logic portion of the controller is used to gate the output of the PNPNS to the wire-spring relays at the appropriate times and to cause the correct sequencing of the various internal circuits.

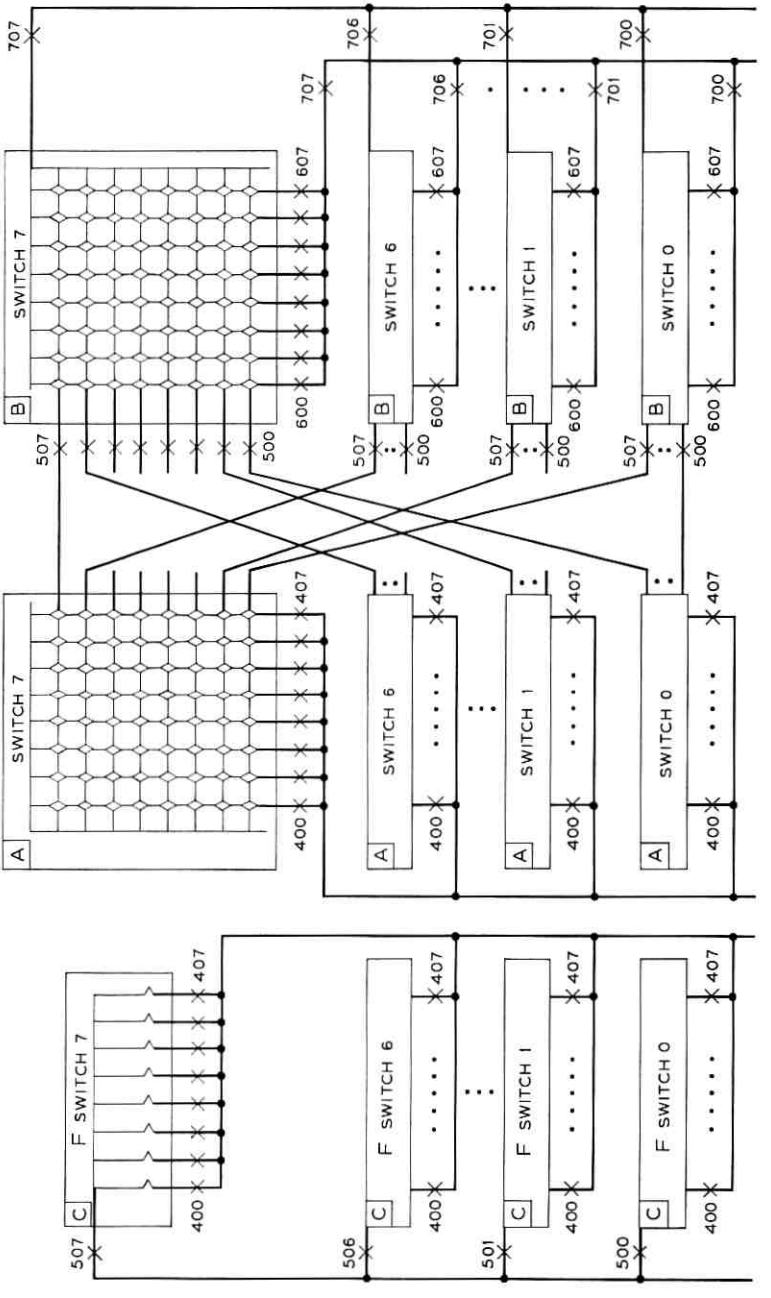



Fig. 10 — Ferreed control winding paths for grid 0. 400 series relays determine input level, 500 input switch, 600 output level, and 700 output switch.

#### 3.2.3 *Ferreed Pulser*

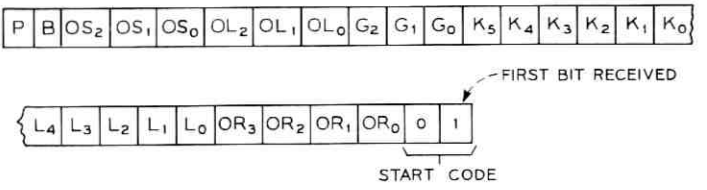
After the wire-spring relays have been operated and a pulsing path exists, the ferreed pulser fires. When the amplitude of the pulse flowing through the ferreed switches is above a certain minimum value, the ferreed pulser generates a pulser o.k. signal. If the order that is to be carried out requires only one pulsing of the ferreed network, the pulser o.k. signal causes the control logic circuit to reset itself. Certain orders to the network, however, require that the ferreed pulser fire into the network twice. For these orders, after the first ferreed pulser o.k. signal is received by the control circuit, a path is set up for the second half of the network operation. The ferreed pulser then fires a second time, and upon receipt of a second ferreed pulser o.k. signal, the control logic circuitry will be reset. Low and medium outputs are provided on the pulser and are used when fewer than the maximum number of ferreed coils are being pulsed.

### 3.3 *Maintenance and Check Features*

#### 3.3.1 *General*

The internal checking circuits monitor the operation of the network controllers. These circuits will prevent the controller from processing an order if an invalid address or other malfunction is detected. The controller remains in the state that it was in when the malfunction was detected until an external reset signal is sent to that controller. The system can monitor the network operation via scan points. Three scan points, S, F, and T, are associated with each controller. These scan points are terminated at a master scanner indicating the state of the controller at any time. Table III shows the state information coded on these three ferrods. Prior to sending out network orders, the program checks the S, F, and T points to see if each controller has successfully cycled the previous order sent it and has returned to the "idle" state.

Should a controller have "locked up," diagnostic routines are run to localize the trouble. By sending special orders to the duplicate controller, the faulty controller can be switched into one of several diagnostic modes. Additional internal points are then connected to a common set of ferrods via a "diagnostic bus." After diagnosing the trouble, the faulty controller may be left in the quarantine mode. It is then prevented from operating any path select relays.




| | | NO. OF BITS |
|---|---|---|
| $OR_3\ OR_2\ OR_1\ OR_0$ | ---1/16 ORDER SELECTION | 4 |
| $L_4\ L_3\ L_2\ L_1\ L_0$ | ---1/32 INPUTS | 5 |
| $K_2\ K_1\ K_0$ | ---1/8 CONCENTRATORS | 3 |
| $K_5\ K_4\ K_3$ | ---1/8 CONCENTRATOR GROUPS | 3 |
| $G_2\ G_1\ G_0$ | ---1/8 GRIDS | 3 |
| $OL_2\ OL_1\ OL_0$ | ---1/8 OUTPUT LEVEL | 3 |
| $OS_2\ OS_1\ OS_0$ | ---1/8 OUTPUT SWITCH | 3 |
| B | ---1/2 B LINK SELECTION | 1 |
| | | 25 |
| | START CODE AND PARITY | 3 |
| | TOTAL BITS | 28 |

OVERALL PARITY ON THE 28 BITS IS "ODD"

Fig. 11 — Network input message format.

### 3.3.2 *Group Check Circuit*

The group check circuit monitors to assure that one and only one relay is operated in each of the groups of wire-spring relays. The output of the group check circuit is fed to a separate flip-flop for each numbered group of relays. The output of each group check flip-flop can be connected to the diagnostic bus. This permits immediate identification of the numbered group that has caused a group check failure. In addition, to provide greater fault resolution, one contact on each wire-spring relay in each numbered group is connected to the diagnostic bus at the same point that the group check circuit for that numbered group is connected to the diagnostic bus. An additional flip-flop is connected to one bit of the diagnostic bus to indicate whether the information that is being presented on the diagnostic bus is the group check flip-flop failure indication or the wire-spring relay contact information. Multiplexing of the two functions, using one additional bus bit, saves seven diagnostic bus points.

TABLE II—NETWORK ORDERS

| | Network Order | Crosspoint Operation | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | CO | First | Second | FCG/NT | F | Third | Fourth |
| OR1 | Connect with cutoff and F open | O | C | C | NC | O | C | C |
| OR2 | Connect with FCG (two-part order) | NC<br>O | O<br>C | C<br>C | FCG<br>NC | C<br>O | C<br>C | O<br>C |
| OR3 | Open cutoff and first stage | O | O | NC | NC | NC | NC | NC |
| OR4 | Close cutoff, open first stage | C | O | NC | NC | NC | NC | NC |
| OR5 | Connect with cutoff closed and F open | C | C | C | NC | O | C | C |
| OR6 | Connect no-test | NC | NC | NC | NT | C | NC | NC |
| OR7 | Open F contact | NC | NC | NC | FCG | O | NC | NC |
| OR8 | Power cross (two-part order) | O<br>NC | C<br>NC | C<br>NC | FCG<br>FCG | C<br>O | C<br>NC | C<br>NC |
| OR9 | FCG | NC | NC | NC | FCG | C | C | C |
| OR10 | Test order | NC | NC | NC | NC | NC | NC | NC |
| OR11 | FCG with first stage and cutoff closed | C | C | C | FCG | C | NC | NC |
| OR12 | FCG with first stage closed, cutoff open | O | C | C | FCG | C | NC | NC |
| OR13 | Special diagnostic order | NC | NC | NC | NC | NC | NC | NC |

Orders 0, 14 and 15 are not assigned.
O = Open.
C = Closed.
NC = No change from previous state.
FCG = False cross or ground check.
F = Test vertical access fereed contact.

### 3.3.3 *Pulser Check Circuits*

The short-to-ground circuit in the ferreed pulser determines if there is a short to ground somewhere in the pulse path portion of the ferreed network. This circuit does not inhibit operation of the ferreed pulser but is available for connection to the diagnostic bus.

A check is made on the ferreed pulse path continuity before the ferreeds are pulsed. This check detects if a path exists.

A pulse verification check is the last check to be performed during a normal operating cycle. This circuit is operational every time the ferreed pulser is supposed to be firing. As described earlier the generation of pulser o.k. signals reset the controller to the idle state successfully completing the network cycles.

## IV. PERIPHERAL DECODER

Every electronic switching system must give its control complex access to many points in the peripheral area that require action signals. Many such signals involve controlling relays in junctor, trunk, and service circuits. The No. 1 ESS uses a signal distributor containing a relay contact translation tree to drive magnetically latching relays in the user circuits.[3] It has been difficult in the past to compete economically with contacts on relays as decoding elements. The rapid progress of monolithic integrated circuits in recent years, however, has made them quite competitive as translators and memory elements. An all electronic action signal distribution scheme used in No. 2 ESS meets the system objectives of low cost, high reliability, and—especially important in a small size system—capability of graceful small modular growth.

### 4.1 *General Operation of Peripheral Decoder*

As described for the network controller in Section 3.2.1, signals are sent to the peripheral decoder in the form of a serial bipolar pulse stream; in this case, seven bits long. Figure 12 is a block diagram of a peripheral decoder. In its most numerous application the decoder will be connected to four trunk or junctor circuits, each having three relays. Each relay is connected to and operated by a flip-flop. Twelve flip-flops are grouped into four buffers each having three flip-flops. The first bit sent to a decoder from the central pulse distributor is a logical "1" start bit. Then come two bits (B0, B1) specifying which of the four buffers the message is destined for. Then the three information bits (I0, I1, I2) followed by a stop bit "1" are shifted in. Upon

TABLE III—CODING OF S, F, AND T FERROD STATES

| Scan Points | | | State of Network Controllers |
|---|---|---|---|
| S | F | T | |
| 0 | 0 | 0 | Success state (idle) |
| 0 | 1 | 0 | Enabled state |
| 1 | 0 | 0 | Not assigned |
| 1 | 1 | 0 | FCG or power cross failure |
| 1 | 1 | 1 | Power removed mode |
| 0 | 1 | 1 | Diagnostic mode 1, 2, or 3 |
| 1 | 0 | 1 | Not assigned |
| 0 | 0 | 1 | Quarantined mode |

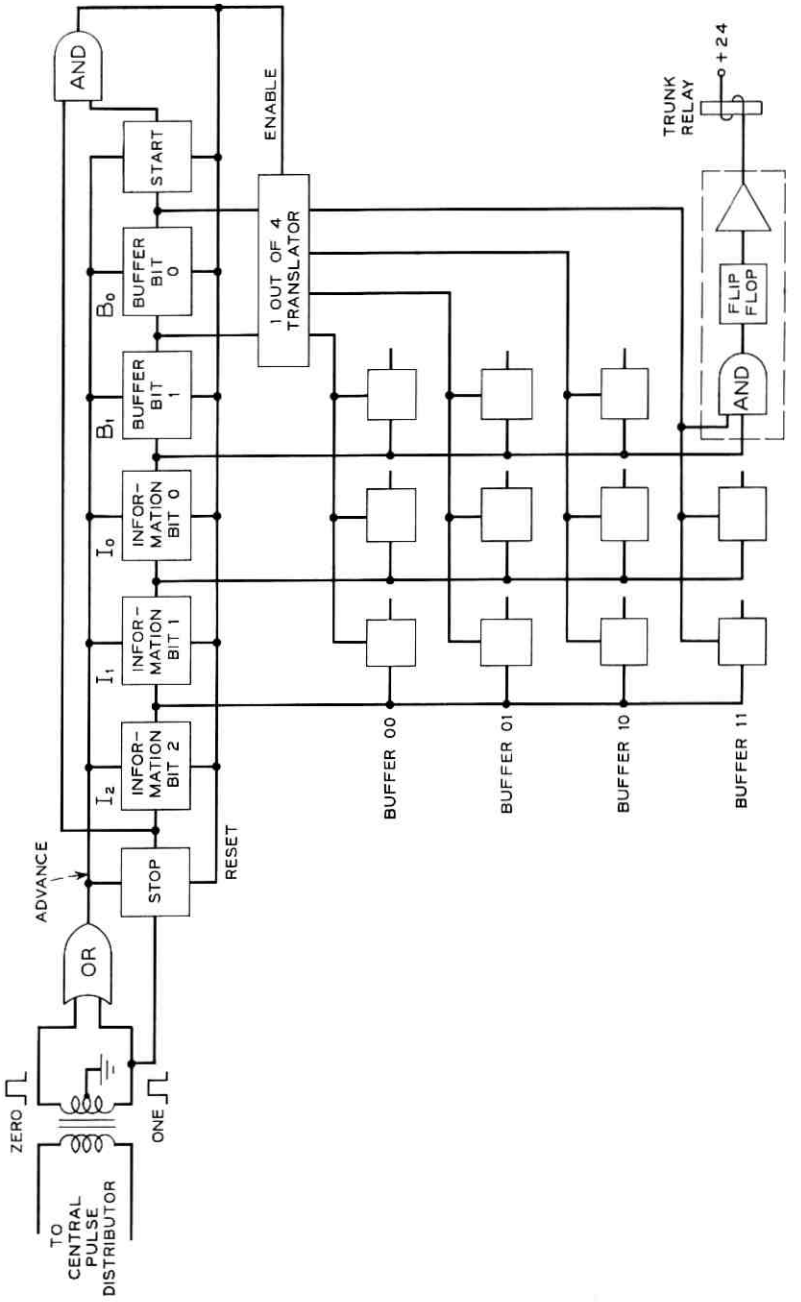



Fig. 12 — Peripheral decoder block diagram.

receipt of the stop bit, the three information bits are gated to the three flip-flops in the buffer specified by the two buffer bits. The shift register is then cleared. The three trunk circuit relays connected to the three flip-flops in the selected buffer will subsequently assume the same operated (released) states as the 1 (0) states of the buffer bits.

The signaling bits play an important role in indicating to the peripheral decoder that a message is complete and that it can now act upon the received information. If for some reason the processor cannot complete a message, the peripheral decoder can be cleared, without resulting in any relay action, by transmitting several logical zeros.

### 4.2 *Circuit Details of Peripheral Decoder*

#### 4.2.1 *Nonracing, Static Shift Register*

A static shift register cell basically consists of a cross connected flip-flop to which nonracing features are added. The nonracing features permit orderly shifting of information without information loss or skipping one or several cells. Discrete component shift registers usually contain reactive components, such as capacitors. On monolithic integrated circuits, capacitors require unreasonably large silicon areas and for this reason other solutions have been developed. The peripheral decoder uses a D-type flip-flop as a shift register cell (Fig. 13). The feature that makes a D-type flip-flop nonracing is that the advance pulse, while sensing and forwarding the information on the input terminal at the same time locks up the input via the two feedback connections. The switching times of the gates that are used in the shift register can be assumed to be reasonably uniform.

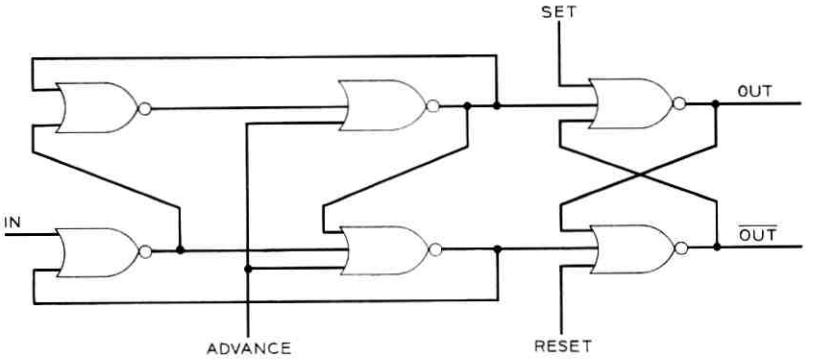



Fig. 13 — D-type flip-flop.

While the use of a D-type flip-flop eliminates races within the shift register itself, there still remain possible race conditions during the transfer and reset operations. Logically the transfer operation should be performed when seven serial bits that have arrived include the two signaling bits. If the shift register is so arranged that the arrival of the last bit, which is the second signaling bit, performs shifting of the bits already received, a race condition may result. The transfer may be attempted too soon and an ambiguous situation will result. This may be overcome by circuitry that behaves like a delay line. The signal generated by the presence of both signaling bits would be delayed long enough to assure that all the information bits have assumed their final position.

Delay circuits, however, usually involve external reactive elements. For this reason, a different approach is used in the peripheral decoder. The shift register is rearranged in such a way that after the sixth bit is received the shifting of the first six cells is inhibited (Fig. 14). The first signaling bit and all the information bits have assumed their final positions well before the expected arrival of the second signaling bit. When the last bit finally arrives it generates a pulse on the ONE side of the input circuit. This pulse is logically combined with the states of the sixth and seventh cells of the shift register. If both states are also ONE, a transfer signal is generated. The shifting between cells six and seven is never inhibited; therefore, a ONE is shifted from cell six into cell seven by the second signaling bit. This produces a properly delayed reset signal after the transfer is completed.

A reset signal for the first six cells should not be generated until the transfer signal disappears. This, again, could be accomplished by a delay circuit. Instead, certain properties of the basic shift register cell are utilized to perform a safe reset without using delay circuits. The advance or shift of information into a cell takes place when and only when the toggle input goes from a positive potential to ground. No shifting takes place on the opposite transition. The toggle input of the eighth cell is inverted so that information is shifted into it when the advance pulse disappears. Thus, the disappearance of the last signaling bit, which produced the transfer signal, shifts a ONE into the eighth cell. The cell in turn applies a reset signal on the first six cells and restores them to the ZERO state. The shift register is now ready to receive the next bit stream. The first bit in the next stream will restore cells seven and eight to ZERO and remove the reset signal.

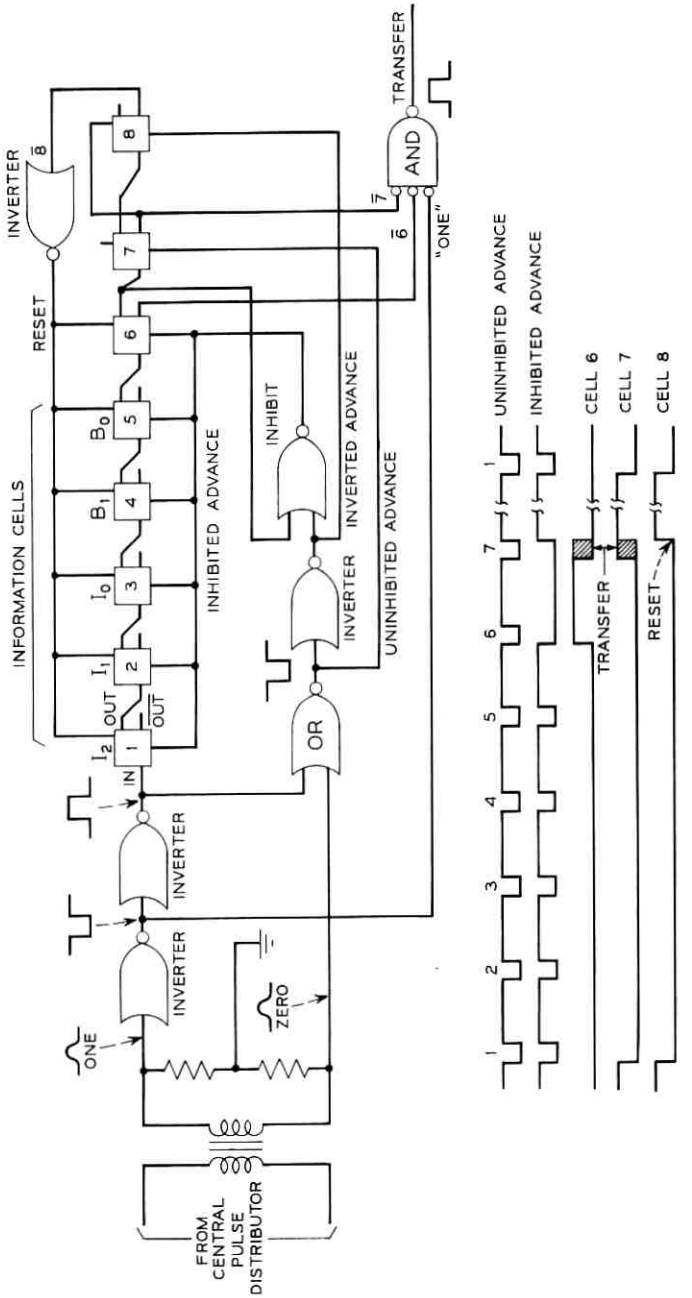



Fig. 14 — Peripheral decoder shift register showing generation of transfer and reset.

#### 4.2.2 *Gated Flip-Flop*

Basically any type of gated flip-flop can be used in the memory matrix for the four three-bit buffers. The gating requirements are such that a flip-flop should be affected only when the proper binary combination exists in the B0 and B1 cells of the shift register and the transfer pulse is present. Whenever the above conditions are satisfied, the flip-flop must assume a state indicated by one of the three information cells.

A decoupling buffer is included to drive a relay driver transistor directly. The two stage buffering that results also helps to attenuate the electrical noise that is coupled into the peripheral decoder from the trunk and junctor circuits.

#### 4.2.3 *Protection Against Noisy Environment*

Peripheral decoders operating relays in trunk and junctor circuits are exposed to severe electrical noise. The noise is generated whenever talking current that is fed to the subscriber through an inductor, is interrupted at or near the trunk or junctor circuit. This noise enters the peripheral decoder via the output leads which by necessity are near the tip and ring. If the noise is not blocked, it may change the state of either the buffer flip-flops or the shift register cells. This, of course, would result in erroneous operations. The problem is complicated by the fact that the noise levels generated on the tip and ring and the noise margins of the integrated circuits are several orders of magnitude apart.

The noise observable on the output leads may be several hundred volts in amplitude and of either polarity. Positive going noise is clamped by the relay coil protection diode, whereas negative going noise will tend to reverse the collector junctions of the buffers and enter the buffer flip-flop. A series diode in the output lead blocks negative going noise except for that part which, because of an extremely sharp wavefront, enters through the diode capacitance. A shunt capacitor of a value sufficiently higher than that of the diode provides the required attenuation.

#### 4.2.4 *Low Voltage Supply*

Most integrated circuits today are designed for low voltage operation. Low supply voltage reduces power dissipation on the chip and permits faster switching times. In a telephone office the lowest voltage provided by the battery plant is +24 volts. The simplest way to ob-

tain low voltage from 24 volts is to use a dropping resistor and a regulating diode.

Unfortunately, the current required by the integrated circuits on a peripheral decoder is such that 9 watts may be dissipated on each circuit unit if a dropping resistor is used. The heat generated by this power far exceeds the allowable level. For this reason a dc to dc converter is provided on each peripheral decoder. The converter reduces the total power dissipated on the printed wiring board to less than 2 watts. It also keeps the integrated circuit supply voltage within the specified limits when the battery voltage drops because of commercial power failure. The integrated circuitry of the peripheral decoder together with the discrete component dc converter and noise filters, is mounted on a standard ESS printed wiring board.*

## V. TRUNK AND SERVICE CIRCUITS

### 5.1 *Introduction*

The advent of stored program control in telephone switching systems has presented unique opportunities for simplifying and reducing size and cost of trunk and service circuits. Functions such as memory, timing, and logic previously performed by electromechanical relays have been taken over by software, with trunk circuits retaining only the most essential transmission related components. The concurrent development of fast switching networks has permitted the rapid connection of service circuits to customer lines and trunks for applying various tones and receiving or transmitting signals that were previously handled either autonomously by trunks or by common control circuits.

As a result of these developments, we have witnessed the fragmentation of large common control circuits of older switching systems into smaller, more specialized service circuits which are used more efficiently by the system. The trunk and service circuits of the No 2 ESS have, in general, followed this basic philosophy. In some cases, minimal hardware tradeoffs were made to avoid complex programs for time consuming system operations.

#### 5.1.1 *General*

The relationship of junctors, junctor circuits, trunk circuits, and service circuits to the switching network and to the scanners and peripheral decoders is shown in Fig. 15.

* Additional equipment information on the peripheral decoder is contained in this issue in Ref. 4.

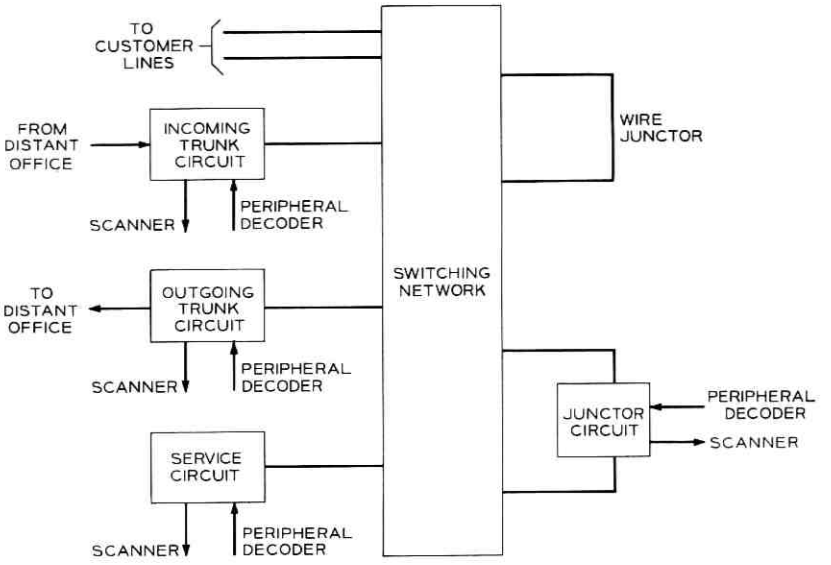



Fig. 15 -- Relationship of trunks, junctors and service circuits to switching network.

Customer lines, trunk circuits, and service circuits appear on one side of the network and wire and circuit junctors appear on the other. In general, circuit junctors are used in talking connections between customer lines of the same office, and in all other connections wire junctors are used. Trunk circuits terminate or originate transmission facilities from or to other offices. These circuits can be connected to each other or to customer lines through the network and wire junctors to establish telephone connections. Junctor and trunk circuits provide through transmission, talking battery voltage to customers, and means for receiving or transmitting call status signals (supervision).

To establish telephone connections a considerable amount of data is received from or transmitted to humans or machines. Service circuits are associated with customer lines and trunks for these functions.

Junctor, trunk, and service circuits receive control information from central pulse distributors via peripheral decoders and transmit call status information to the system processor via the various scanners. These circuits are switched in and out of a call under the control of the stored program as required by the progress of the particular type of call being handled. The points of supervision of a call reside in these circuits, passing from one to another as the call progresses. As examples, supervision is transferred from the calling line's line

ferrod to a ferrod in the customer dial pulse receiver when the dialing connection is established. In an intraoffice call supervision during talking is performed at the circuit junctor for both calling and called line.

### 5.1.2 *Basic Characteristics of Trunk and Service Circuits*

The basic switching design philosophy and transmission configuration of No. 2 ESS trunk and service circuits has followed the standards established by similar No. 1 ESS circuits.[7] The major circuit departure has been in the use of ordinary wire-spring relays rather than magnetic latching ones for trunk and service circuit control. These relays are controlled, as described in Section 4.1 directly by the central processor through peripheral decoders rather than signal distributors. In another departure, audible ringing tone is applied via junctor circuits and incoming trunk circuits. This eliminates a network connection to a tone circuit and the reservation of a network path. Significant processor real time is saved.

Regarding physical design, a new universal trunk and junctor frame has been designed that accepts junctor circuits, incoming and outgoing trunk circuits as well as operator trunk circuits. These circuits are permanently wired to 2-inch mounting plates. Each plate requires access to one peripheral decoder for relay control, and to eight ferrod sensors for scanner functions.

Tone circuits in No. 2 ESS are simplified and treated as lines. They respond to a connect signal from the circuit junctor or trunk circuit by connecting the assigned tone and restoring to normal when this connect signal is removed.

The power cross test made to customer lines prior to connecting sensitive service circuits to them is made a function of the network in No. 2 ESS. This permits not only savings from centralizing this function, but a substantial simplification of the service circuits that previously included such a test.

No. 2 ESS peripheral circuits have also been designed to permit operation with long customer loops using Uniguage design and to give coin lines dial tone before coin deposit for dialing the new universal emergency code (911), as well as emergency, ordinary assistance, information, and toll calls to operators.

## 5.2 *Trunk Circuits*

"Trunk circuits" are those control circuits that associate the switching network terminals with the transmission facility. In addi-

tion to voice frequency transmission elements, they contain means for transmitting and receiving call status signals (supervision) from or to other offices. "Trunks," on the other hand, are the entire channels from the switching network terminals of one office to the switching network terminals of another. They include trunk circuits, signaling equipment, transmission equipment as well as physical wires or carrier channels from one office to another.

Trunk circuits are classified in many ways. Considering methods of supervision, we have: loop (dc), reverse battery, high-low and E&M lead types. Considering signaling (address) language, we have: multifrequency and dial pulse types. And finally, considering direction of control, we have: one-way, which can be either incoming or outgoing; and two-way, which can be controlled from either connecting office.

The ESS trunk circuit philosophy of simplification and assignment of many of their functions to program control or service circuits has resulted in a significant reduction in the number of different types of trunk circuits needed in each office.

Trunk circuits can be divided into three categories; those to (*i*) central offices, (*ii*) switchboards, and (*iii*) test and repair desks.

A special case of the first category is a circuit connecting two customers of the same central office. This circuit is known as a junctor circuit because of its location in the center of the network fabric. It is to be distinguished from a junctor (wire junctor) which is similarly located but used in all other types of connections.

#### 5.2.1 *Junctor Circuit*

Figure 16 illustrates the junctor circuit. The circuit is controlled by three relays making possible eight circuit states which are illustrated in Figure 17.

With all three relays released (BYPASS) the circuit reduces to two metallic paths connecting the two associated network terminals. Since this is exactly the form of the wire junctor, this circuit can be used as such at times when an idle wire junctor between two networks is not available.

With relays A and B operated (TALK) the circuit permits the two local customers connected via the network to its terminals to carry on a conversation. Talking battery voltage is fed to the customers' sets through ferrod sensors which are used to monitor their switchhook status. To prevent the shunting of the ac talking signal by the

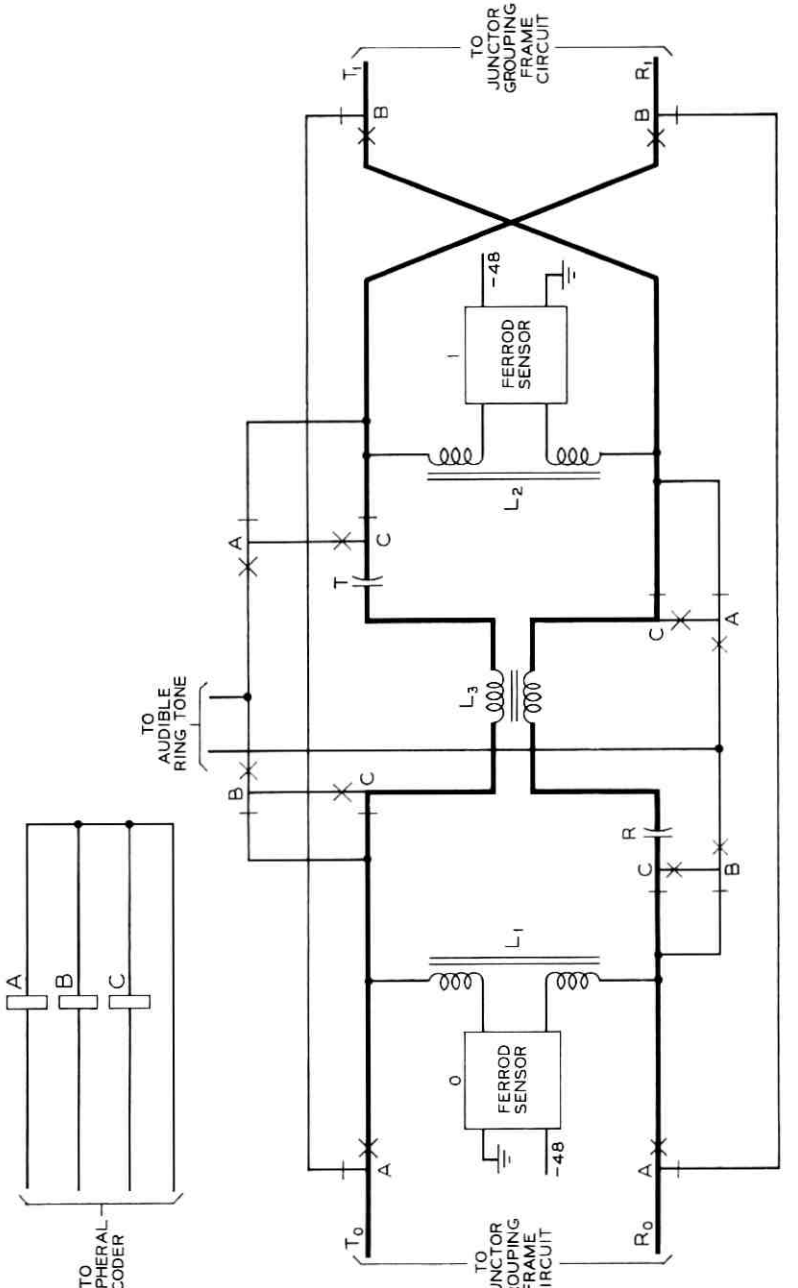



Fig. 16—Junctor circuit.

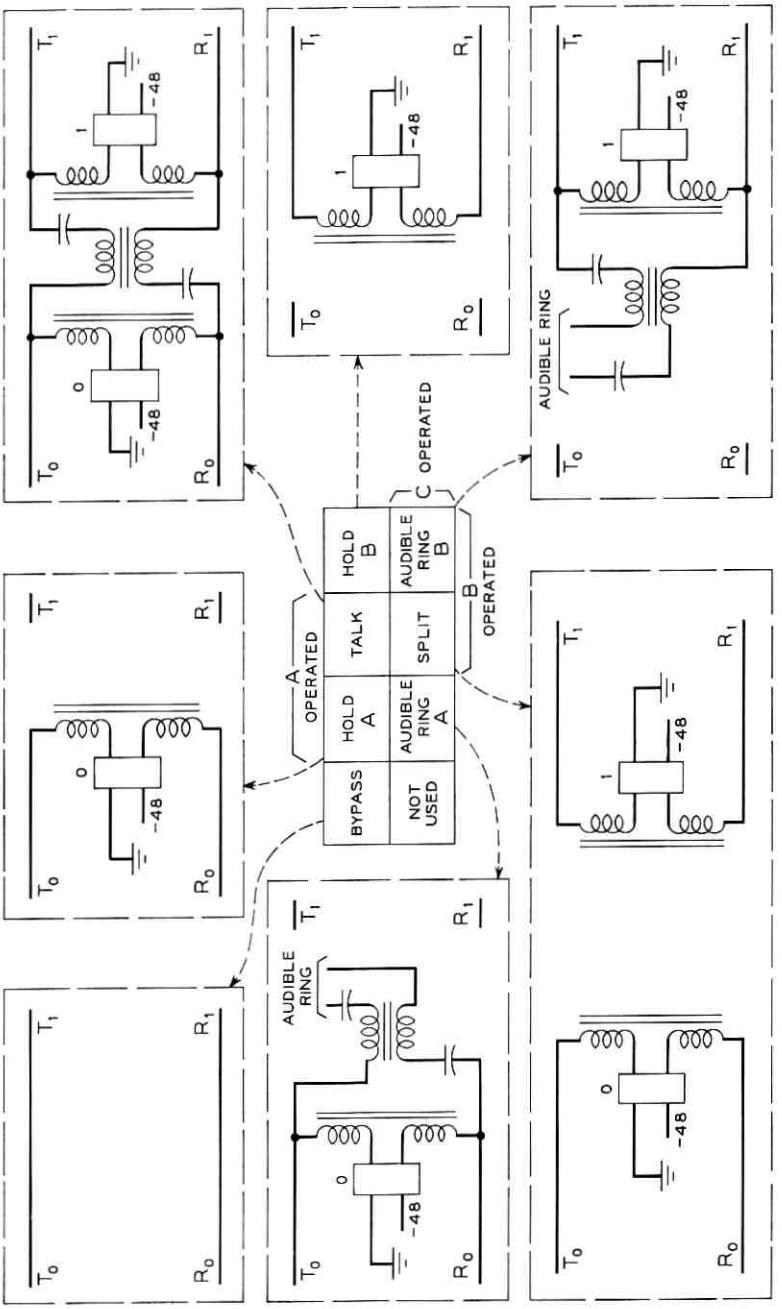



Fig. 17 — Possible junctor circuit states.

low impedance battery and ferrod sensors, series battery feed inductors are used.

With relay A or B operated (HOLD A or HOLD B) the circuit provides switchhook monitoring of either the customer connected to terminal A or the one connected to terminal B. This is an intermediate state in the processing of a call. The calling customer has finished his dialing and is waiting for the system to alert the called customer.

With relays A and C or B and C operated (AUDIBLE RING A or AUDIBLE RING B) the circuit provides switchhook supervision and audible ring tone to the calling customer. The junctor circuit is placed in this state when alerting of the called customer begins. Finally, with all three relays operated, the circuit provides for switchhook supervision of both customers connected to its terminals without permitting a talking path between them. This state is used, in some cases, when processing custom calling service calls.

5.2.2 *Outgoing Trunk Circuit*

Figure 18 illustrates the outgoing trunk circuit used with loop (dc) type supervision. This circuit together with the similar incoming trunk is the most commonly used circuit and great efforts were made to keep it simple. Connect and disconnect signals are transmitted to the distant office by closing or opening the dc "loop" toward it. This is accomplished by inserting in series with the outgoing terminals a low resistance ferrod sensor. This low resistance circuit causes current to flow, thus signaling the distant office. The distant office, in turn, responds by reversing the battery of the circuit. This is detected by the ferrod sensor since the associated diodes will permit current to flow through its windings. Three relays are used to control this circuit which assumes eight different states under program control. Figure 19 illustrates these states.

With no relay operated (IDLE) the circuit is open toward the network and offers an ac termination to the transmission facility (idle circuit termination). With C relay operated (BYPASS) the circuit is "bypassed"; that is, the transmission and supervisory elements are disconnected and the circuit reduces to two metallic paths. In this state, the circuit permits the direct association with the transmission facility, of an address transmitting circuit of a language (dial pulse or multifrequency) understandable by the distant office.

With B relay operated (HOLD) the circuit maintains a connect signal to the distant office while the address transmitter (which initiated this signal) is disconnected.

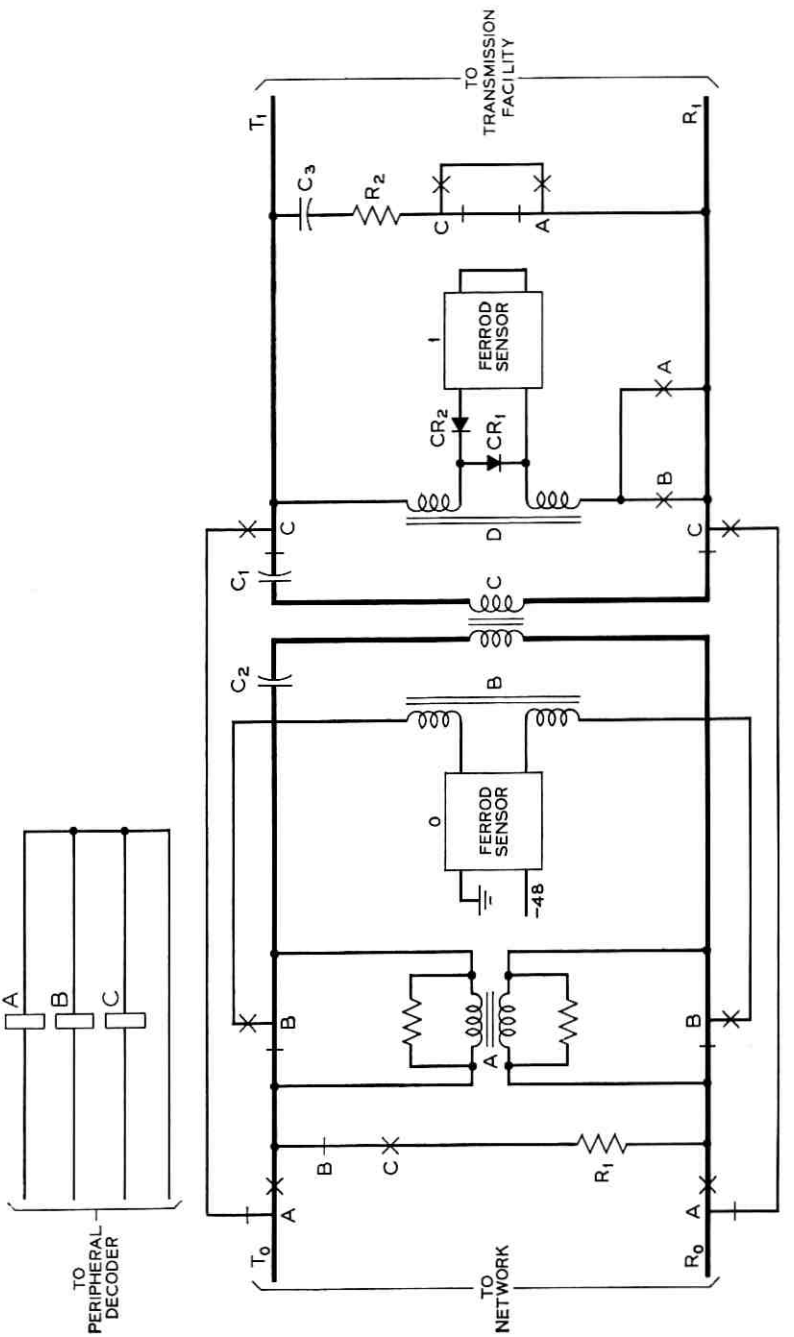



Fig. 18—Outgoing trunk circuit.

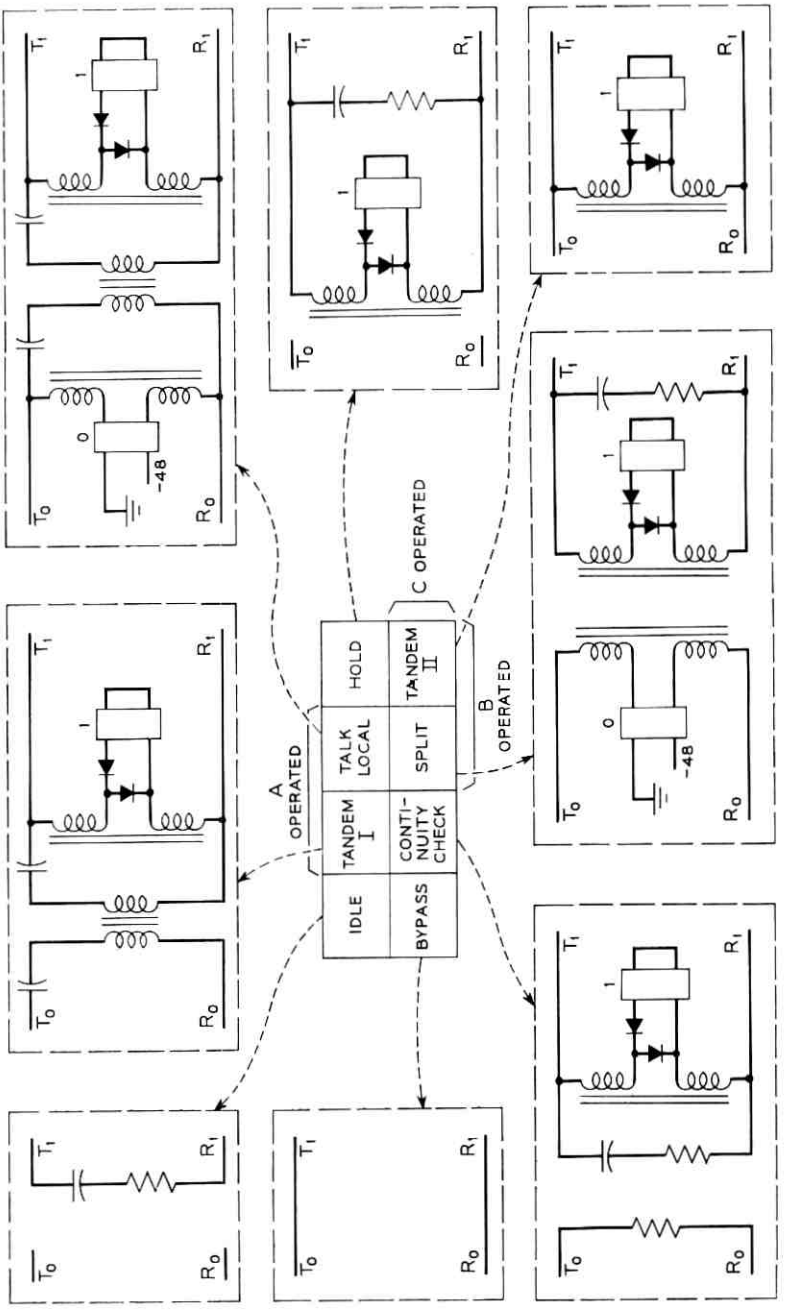



Fig. 19 — Possible outgoing trunk circuit states.

With relays A and B operated (TK LOC) the circuit provides for a local customer to talk to one in a distant office. Talking battery voltage is supplied to the local customer, and ferrod sensor scan points are connected to both parties for supervision.

When both the calling and called customers are in different distant offices and this third office is used as a tandem point, the respective offices are supervised by the ferrod sensors of the associated trunk circuits, the calling office at the incoming trunk circuit and the called office at the outgoing trunk circuit. Ferrod sensors used for local customers are not needed and are disconnected to improve transmission. Two talking states are provided for such calls, TDM I (A relay operated) and TDM II (B and C relays operated). The latter is used with operator incoming trunk circuits, which provide the transmission transformer and the former with incoming trunk circuits from other local offices.

Because in tandem states direct current does not flow through the network path, the integrity of the network path is checked by placing the incoming circuit in a local talking state and the outgoing in the CONT CHECK state. In this state a resistor is placed across the network terminals of the outgoing circuit causing current to flow through the ferrod sensor of the incoming circuit.

One final state, used in special applications, is the SPLIT state (all relays operated). In this state the circuit can supply talking battery voltage to a local customer and supervise both local and distant customers without permitting a conversation between them.

#### 5.2.3 *Incoming Trunk Circuit*

Figure 20 illustrates an incoming trunk circuit of the loop (dc) type. This circuit can be thought of as the connecting circuit at the distant ESS office working with the outgoing trunk circuit just described. Again, three relays are used to control the circuit and two ferrod sensors are used to pass call status signals to the processor. Figure 21 illustrates the eight possible states of this circuit.

With no relays operated (IDLE) the circuit presents an idle circuit termination to the transmission facility while a low resistance ferrod sensor monitors for a connect signal from the distant office.

With the A relay operated (BYPASS) the transmission facility is bypassed to an appropriate digit receiver connected through the network.

With C relay operated (AUDIBLE RING) audible ring tone is returned to the distant office while the called local customer is alerted.

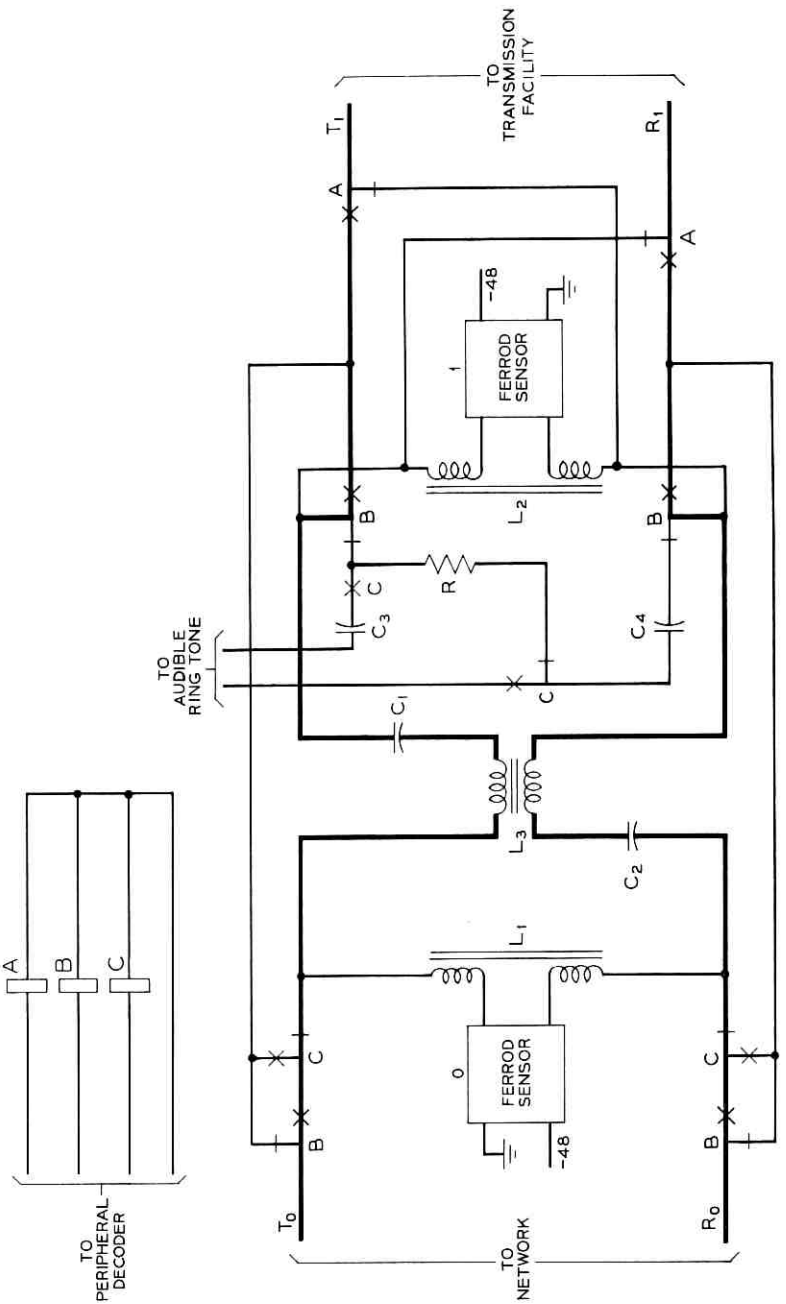



Fig. 20 — Incoming trunk circuit.

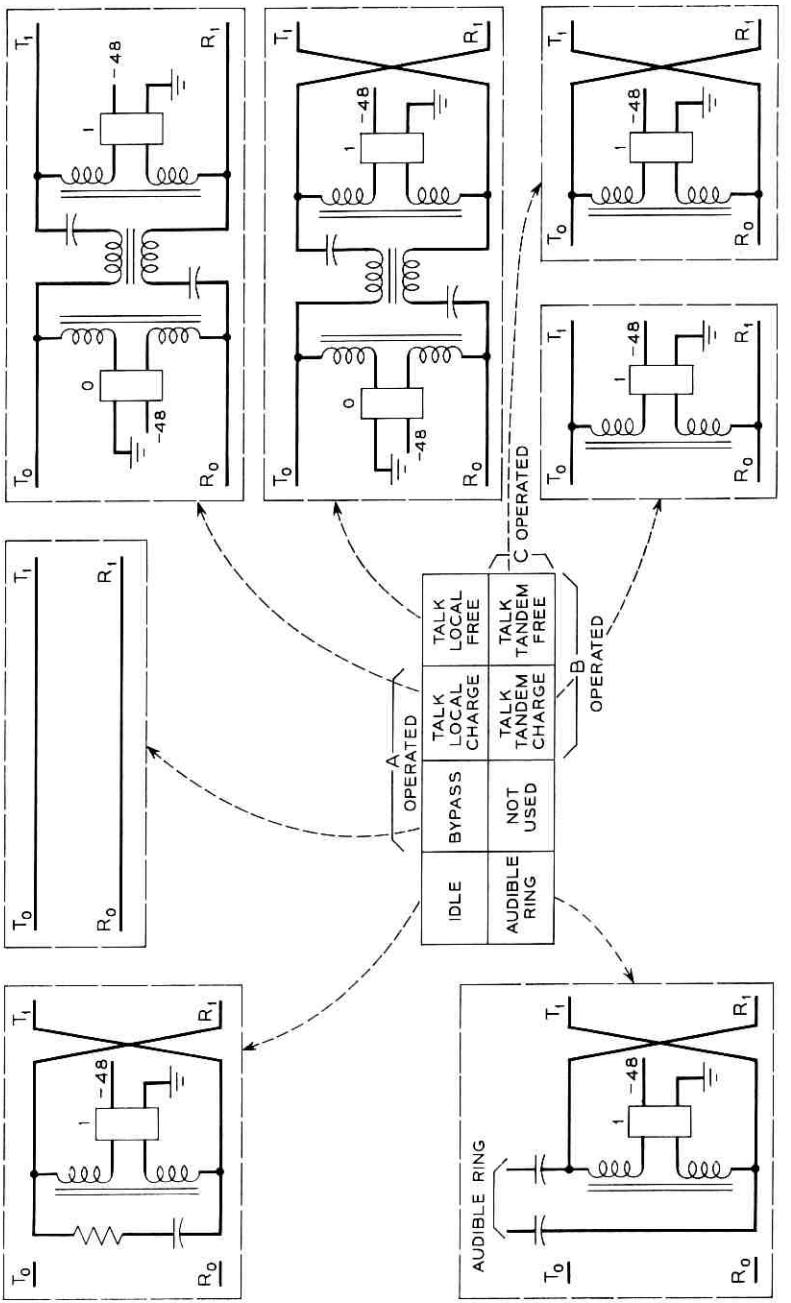



Fig. 21 — Possible incoming trunk circuit states.

When the called customer answers, the circuit is switched to the TALK LOCAL CHARGE state (A and B relays operated) if the call is to be charged or to the TALK LOCAL FREE if the call is free, such as a call to the telephone company business office.

The circuit provides two tandem states for calls to be routed to another office. TDM CHARGE (all relay operated) and TDM FREE (B and C relays operated) are used depending on charge information as just described.

#### 5.2.4 *Other Trunk Circuits*

A number of small variations of the above circuits have been developed for use with offices of special character. Step-by-step offices, for instance, can transmit digit information immediately after sending the connect signal without waiting for a confirmation signal that a digit receiver is available. In such incoming trunk circuits a relay is provided in place of a ferrod sensor to receive and shape the dial pulses before passing them to the input-output logic.

Another variation is an outgoing trunk circuit to centralized automatic message accounting offices where monitoring of the transmission facility is required at all times regardless of whether a connect signal has been transmitted. Circuits have also been developed for transmission facilities using separate leads for supervisory signals (E&M leads). For connecting to switchboard operators both inband and dc signaling circuits have been developed.

### 5.3 *Service Circuits*

Service circuits are used to perform specialized functions such as digit reception and transmission, alerting, informing, coin control, and others. These circuits appear on terminals of the switching network, as shown on Fig. 15, which they use for access to lines and trunk circuits. Service circuits, like trunk circuits, communicate with the system control via scan points and peripheral decoders. Their holding time is generally less than that of trunk circuits and their use is primarily confined to the setup period of a call.

#### 5.3.1 *Digit Receiving Circuits*

Address information, in the form of coded decimal digits can be received from either local customers or other central offices. Because of differences in codes and signaling methods, different digit receiving circuits are used in each case.

5.3.1.1 *Digit Receiving from Local Customers.* Local customers send address information either by means of dial pulses (generated by a rotary dial opening and closing a dc loop) or by *Touch-Tone*® telephone frequency tones. Because a local customer may have telephones of both types, the central office must be prepared to receive either type of pulsing. Two circuits, a customer dial pulse receiver circuit and a *Touch-Tone*® calling detector are associated with a customer line for this purpose. The former receives dial pulses and contains all the control features as well as facilities for dial tone and party testing. The latter is strictly an ac transistorized receiver with its input connected across the input of the customer dial pulse receiver.

5.3.1.2 *Digit Receiving from Other Offices.* Address information from other offices is received in the form of either a 2-out-of-6 multifrequency code or in dial pulses. It is not anticipated that the No. 2 ESS office will work with panel or other older design offices requiring facilities for reception of revertive or panel call indicator codes.

*Multifrequency*

The preferred method for sending digit information in modern telephone practice is by means of voice frequency tones coded to represent decimal digits on a 2-out-of-6 code (multifrequency pulsing). The distant office transmits a connect signal and the receiving office responds by connecting the incoming trunk circuit to a multifrequency receiver circuit which then signals the distant office that it is ready to receive the digit bearing MF pulses. The multifrequency receiver circuit is a transistor circuit with frequency discriminating ability which passes the digit information to ferrod sensor scan points. The central processor, by scanning these points, determines the digits sent which are then stored in the call store.

*Dial Pulse*

There are two types of dial pulsing that can be received by the No. 2 ESS office; pulsing from offices that use register type circuits which wait for a start signal before outpulsing and offices that are directly controlled by customer dials which do not wait for a start pulse but outpulse immediately following the transmission of a connect signal.

Pulsing from offices requiring a start pulsing signal is received in a simple relay service circuit called "trunk dial pulse receiver circuit." This is connected to the trunk circuit via the switching network in a manner similar to that used for multifrequency pulsing.

Dial pulsing, with no start pulsing signal, is received, primarily,

from nonregister type step-by-step offices. The office transmits a connect signal to the No. 2 ESS office and directs to it customer dialing. Part of the interdigital interval preceeding the first digit sent to the No. 2 ESS is used up at the step-by-step office in selecting a trunk to the No. 2 ESS office. Depending on traffic conditions, the balance of interdigital time left for the No. 2 ESS to select a receiver circuit is hardly sufficient. For this reason, pulsing of this type is received at the trunk circuit where pulses are collected by the system via scan points. A circuit similar to the loop incoming trunk circuit is used except that a special relay is used to receive the pulses. Because the input-output wired logic of the No. 2 ESS can scan for dial pulses at a nominal 10 millisecond rate, there is no need for special pulse stretching circuits.

5.3.2 *Digit Transmitting*

The problem of digit transmission is, of course, the inverse of the one just described. Address information is transmitted in coded decimal digits to other offices in the form of multifrequency pulses and dial pulses.

5.3.2.1 *Multifrequency Digit Transmission.* As described in Section 5.3.1.2, the most common method of digit transmission between central offices is by means of the 2-out-of-6 multifrequency code. Multifrequency pulsing is used with No. 5 crossbar, crossbar tandem, and No. 1 ESS offices. No. 2 ESS uses a service circuit that is connected to the transmission facility through the switching network and the trunk circuit. The trunk circuit is previously set at the "bypass" state which permits exchange of dc supervisory signals between the multifrequency transmitter circuit and the distant office. The absence of series and shunt circuit elements from the trunk circuit, when set at the bypass state, prevents any degradation or attenuation of multifrequency pulses.

The multifrequency transmitter circuit contains the necessary oscillators for generating the voice frequency signals which are keyed by relays under system control. This circuit also includes peripheral decoder controlled relays that transmit connect signals to distant offices as well as connections to the scanner circuit for receiving start pulsing signals.

5.3.2.2 *Dial Pulse Transmission.* Dial pulse digit transmission is used with step-by-step offices. Unlike the dial pulse sending circuits of crossbar systems which use self-contained pulse generating circuits, or

No. 1 ESS in which the central control signals the beginning and end of each dial pulse to each transmitter, No. 2 ESS dial pulse transmitters connect to a pair of duplicated common buses for dial pulse timing. Each pair of buses provides a source of two accurate timing signals. Signals on one bus indicate the beginning and signals on the other bus indicate the end of each dial pulse interval. These signals are gated into the transmitters by peripheral decoders for the time interval necessary to send each digit. This method minimizes pulse generating circuitry and program since only the beginning and end of each digit must be signaled to the transmitter.

#### 5.3.3 *Alerting and Informing Circuits*

Alerting and informing circuits are used to permit communication between man and machine. In the first category we include ringing circuits which alert a customer to an incoming call; and in the second, various tone and announcement circuits which inform the customer about the progress of his call.

5.3.3.1 *Ringing Circuits.* Ringing circuits are used to apply a 20 Hz ac signal superimposed on −48 volt dc central office battery. The ac voltage is used to ring the bell in the customer's telephone which is bridged across the line with a series capacitor. The central office battery is used to energize a relay in the ringing circuit when the customer answers. This relay when operated disconnects (trips) ringing from the customer's line and causes the connection of a supervisory scan point to it. The connection of the ringing circuit is then removed and that between calling and called customers is established through the network.

Two ringing circuits have been developed for No. 2 ESS. One, used with most lines, alerts single and two party lines as well as PBXs while the other rings four and eight party lines. The former connects to lines interrupted (2 seconds ON, 4 seconds OFF) ringing obtained from the ringing and tone frame. The latter is connected to a continuous ringing source at the ringing and tone frame, which it interrupts under peripheral decoder control to suit the needs of the various four and eight party, and rural lines.

5.3.3.2 *Tone and Announcement Circuits.* Tone circuits in No. 2 ESS are treated like customer lines by the program. The circuit consists of a relay bridged across a pair of network terminals which operates when a junctor circuit or a trunk circuit connects to it. This relay connects to these terminals the tone assigned to it which is obtained from the ringing and tone plant. Tones such as 60 ipm (interruptions per minute)

busy, 120 ipm overflow, and the "no such number" tone are normally supplied by these circuits.

For verbal announcements, two circuits have been developed that connect to the recorded announcement frame. One connects to the announcement immediately when seized, the other waits until the beginning of the announcement before connecting it to the customer.

### 5.3.4 *Coin Control and Applique Circuits*

Coin station functions in No. 2 ESS are performed only by the coin control circuit which is associated with the coin line via a switching network connection for the duration of such functions. The centralization of these functions in the coin control circuit has permitted a considerable simplification of trunk circuits as well as the elimination of the distinction between coin and noncoin types.

Applique circuits are used whenever specialized information must be given to or obtained from the central processor.

5.3.4.1 *Coin Control Circuit.* The coin control circuit is connected to a coin line momentarily to verify the presence of a coin, and to collect or to return a coin. The circuit is controlled by relays operated by the peripheral decoder circuit to apply the appropriate dc potentials to the coin line. Ferrod sensor scan points are used to monitor supervision as well as to verify coin presence and the flow of coin collect or return current.

5.3.4.2 *Applique Circuits.* There are three types of applique circuits. The peripheral decoder applique, the master scanner applique, and the interrupter applique circuits. The peripheral decoder applique circuit consists of a single relay controlled by the system through the peripheral decoder. The contacts of this relay are used to light lamps or to control other functions not normally performed by a specific circuit. The master scanner applique circuit consists of a single resistor that converts relay operations to current suitable to ferrod sensor saturation at the master scanner circuit. These signals are from various types of monitoring devices at the central office such as alarming key operations. The interrupter applique circuit supplies 60 and 120 interruptions per minute contact closures to various circuits for busy or overflow indications.

### 5.4 *Test Circuits*

Line, trunk, and service circuits are tested by specific test circuits that use the switching network for access. There are three basic hard-

ware entities that perform tests on trunk and service circuits: the trunk test circuit, the group of service test circuits, and the automatic line insulation test circuit. Tests may be initiated by call processing programs when faults are detected, routinely by the trunk test circuit's test programs, or by teletypewriter requests. Automatic line insulation tests may also be requested by the local test desk.

#### 5.4.1 *Trunk Test Circuit*

The trunk test circuit is mounted on its own frame in the maintenance area of the office. It is a manually operated circuit and maintenance craftsmen can obtain access to any of the trunks or service circuits through three access trunks to the switching network. The following test facilities are part of this frame:

(*i*) The Transmission Test Circuit permits testing lines and trunks by special terminations and tones.

(*ii*) The voltmeter circuit permits various leakage resistance measurements between trunk conductors or between one conductor and ground. It is also possible to test for the presence of foreign potentials and qualitative capacitance measurements.

(*iii*) The state change control permits placing any trunk or service circuit in any of its possible circuit states through peripheral decoder actions.

(*iv*) Test control permits taking circuits in and out of service and monitoring trunk circuit E&M leads.

(*v*) The impulse counter permits noise measurements.

#### 5.4.2 *Service Test Circuits*

For every service circuit in the No. 2 ESS office there is a corresponding test circuit. This test circuit uses the switching network for access and proceeds, under program control, to introduce various extreme conditions under which the circuit being tested is expected to perform. Following is a brief description of the various test circuits and the tests they perform.

Customer signaling is received by the customer dial pulse receiver and the *Touch-Tone*® calling detector circuits. To test these circuits, the system can connect them to a customer dial pulse receiver test circuit and a *Touch-Tone*® calling detector test circuit. The customer dial pulse receiver test circuit then transmits a series of marginal dial pulses to the customer dial pulse receiver being tested while the associated program checks its response by observing the digit receiving

circuitry of the input-output control. In addition, specialized functions such as dial tone transmission, toll diversion, and party test features are tested. The *Touch-Tone®* calling detector test circuit is similarly a precise *Touch-Tone®* telephone frequency transmitter that transmits a series of test digits designed to test the circuit performance at the extreme ends of each channel. In addition, checks are made for the rejection of signals slightly out of channel, presence of third frequencies, and proper timing response.

Multifrequency transmitters and receivers are tested by using one to test the other. Rather than connecting a pair of these circuits together directly through the network for this purpose, the connection is made through a third circuit, the multifrequency test environment circuit. This circuit has two appearances on the switching network, one for the transmitter and one for a receiver. A number of tests are performed on the service circuits by altering this artificial environment. One such test is the flat loss test where the pair of transmitted frequencies is considerably attenuated and the receiver is checked for response. In another test, the twist test, one of the frequencies is attenuated relative to the other and the receiver is checked. Tests are also made for double keying, for response to modulation products, and for proper timing.

The tone presence detector is a circuit that can detect audio frequency tones. It is used to test the output of the various tone and recorded announcement circuits as well as the performance of the conference circuits.

A common circuit tests both the coin control and the ringing circuits. The ringing and coin control test circuit simulates a customer station set or a coin set and under extreme conditions tests for application of ringing potentials, operation of the ring trip relay, as well as the application of coin test, coin collect and coin return potentials.

The proper operation of the ringer at the customer's set is checked by the station ringer test circuit. This circuit is connected at the request of a craftsman by dialing a special code usually at the time of installation of a set. Associated with this circuit a *Touch-Tone®* telephone station test circuit may be used for testing the accuracy of the *Touch-Tone®* telephone oscillators. As expected, this is a precise receiver tuned to the *Touch-Tone®* calling frequencies.

### 5.4.3 *Automatic Line Insulation Test Circuit*

The automatic line insulation test circuit is used to test customer lines for insulation defects. It applies a small voltage to the customer

line and measures the resulting current. It reports this to the system control via ferrod sensor scan points. There are three test modes for this circuit.

(*i*) Testing for leaks between tip and ring leads and between ring and ground. The first test usually indicates trouble in drop wire at the customer premises; the second, in open wire conductors.

(*ii*) Testing for leaks between tip and ground and from ring to ground. This test detects troubles in cable terminals.

(*iii*) Testing for leaks to battery from tip or ring leads. This test is used to detect defects in cable sheaths. Such defects permit leaks to battery on ring leads of other lines of the same cable because moisture is present.

Line insulation tests may be started automatically by the system at a predetermined time or by a testman at a local test desk.

REFERENCES

1. Spencer, A. E. and Vigilante, F. S., "System Organization and Objectives," B.S.T.J., this issue, pp. 2607–2618.
2. Browne, T. E., Quinn, T. M., Toy, W. N., and Yates, J. E., "Control Unit System," B.S.T.J., this issue, pp. 2619–2668
3. Freimanis, L., Guercio, A. M. and May, H. F., "No. 1 ESS Scanner, Signal Distributor, and Central Pulse Distributor," B.S.T.J., *43*, No. 5, part 2 (September 1964), pp. 2255–2282.
4. Lonnquist, C. W., Maganello, J. C., Skinner, R. S., and Skubiak, M. T., "Apparatus and Equipment," B.S.T.J., this issue, pp. 2817–2863.
5. Feiner, A., *The Ferreed*, B.S.T.J., *43*, No. 1, part 1 (January 1964), pp. 1–14.
6. Danielson, D., Dunlap, K. S., and Hofmann, H. R., No. 1 ESS Switching Network Frames and Circuits," B.S.T.J., *43*, No. 5, part 2 (September 1964), pp. 2221–2253.
7. Biddulph, R., Budlong, A. H., Casterline, R. C., Funk, D. L. and Goeller, L. F., Jr., "Line, Trunk, Junctor, and Service Circuits for No. 1 ESS," B.S.T.J., *43*, No. 5, part 2 (September 1964), pp. 2321–2353.

# Service Features and Call Processing Plan

By ROBERT J. ANDREWS, JOHN J. DRISCOLL, JOHN A. HERNDON, PHILIP C. RICHARDS, and L. RALPH ROBERTS

(Manuscript received April 4, 1969)

*In No. 2 ESS modern telephone features are provided by concise call processing programs, which make extensive use of subroutines. This paper discusses the features to be available with the introduction of the system. It describes the call processing program plan used to achieve efficient use of program storage and gives details of supervision processing, digit handling, peripheral equipment control and translation. This paper also describes how call processing programs control the progress of an intraoffice call, and how they are used in testing of trunk and service circuits.*

## I. INTRODUCTION

The No. 2 Electronic Switching System performs the functions of a local telephone central office under the control of a stored program acting through data processing, input-output, and two-wire switching equipment. Virtually all the actions of the system are determined by the sequences of instructions coded and stored in memory.

The No. 2 ESS stored program may be divided into two parts: the call processing programs, which provide telephone service and operational features, and the maintenance and administrative programs, which maintain an operational system in the presence of troubles and diagnose the faulty units. Thus, the purpose of the stored program is identical to the purpose of the central office itself, including the implicit function of assuring dependable service.

This paper deals with the service features of No. 2 ESS, the call processing programs which provide these features, and with testing lines, trunks, and service circuits which use the call processing programs. The maintenance and administrative programs, the circuits and the equipment for No. 2 ESS are described in other papers.[1–4]

## II. SERVICES AND OPERATIONAL FEATURES

The usefulness and value of No. 2 ESS to a telephone operating company will be determined by its service and operational features. The service features provide for the needs of telephone customers and the continual evolution of services throughout the life of the system. The operational features permit interfacing with transmission and signaling facilities, station equipment, and other switching systems. Additional operational features provide an efficient interface with operating company personnel by including facilities for day-to-day maintenance and administration,[1] for taking traffic and performance measurements and for recording data for computation of service charges.

### 2.1 *Service Environment and Performance*

No. 2 ESS is designed to provide service and operational features consistent with modern customer service needs and the environment in which medium size central offices may be expected to operate.[5]

The No. 2 ESS design anticipates a wide diversity of customer usage statistics from installation to installation. For example, 80 percent of Bell System central offices with between 1,000 and 10,000 lines have average busy-season busy-hour characteristics within the following ranges:

(*i*) Originating plus incoming calls per line 0.7–2.1
(*ii*) Originating plus incoming hundred-call-seconds per line 1.3–2.9
(*iii*) Originating traffic completing within the office 32%–92%

No. 2 ESS installations will be individually arranged or engineered using the modular attributes of the peripheral equipment design to provide a quality of service consistent with current operating company standards.[3] Table I shows the expected switching service performance objectives for No. 2 ESS.

The actual performance of an individual installation may occasionally fail to meet these standards when traffic volumes or patterns exceed the operating company forecasts which were used to engineer the equipment quantities, or if the equipment incurs physical damage.

### 2.2 *Telephone Services*

In general, No. 2 ESS provides the full complement of telephone services necessary in any modern telephone switching system. Although their implementation within No. 2 ESS is unique in many

TABLE I—SWITCHING SERVICE PERFORMANCE OBJECTIVES FOR NO. 2 ESS

| | |
|---|---|
| Dial tone speed (busy season—busy hour) | Probability $\leqq 0.015$ of delay $\geqq$ 3 seconds |
| Line to line connections | Blocking probability $\leqq 0.04$ |
| Incoming trunk to line connections | Blocking probability $\leqq 0.02$ |
| Call processing irregularities | Probability $\leqq 1 \times 10^{-4}$ |

respects, the customer operating procedures and resultant system responses have been made uniform and consistent with other switching systems. These services can be considered in two categories, standard and custom.

### 2.2.1 *Standard Telephone Service*

Standard service consists of all dialable calls from and to customer lines using dial pulse signaling and 20 Hz ringing. In No. 2 ESS, service has been implemented to be consistent with existing switching systems. The system has flexibility to adapt to evolving needs and provide future services economically.

The term "all dialable calls" refers to the ability to accept and derive appropriate routing for all telephone numbers used in current practice. These range from the single digit zero (operator), through three digit service codes such as 411 (directory assistance) or 911 (emergency), up to ten digit numbers for direct distance dialing. Included in the design are capabilities which anticipate modifications of the existing number plan, for example, single digits (one or zero) as prefixes and arrangements for international direct distance dialing.

In addition to determining the appropriate route or specific line for each valid number, No. 2 ESS provides standard signals, tones, and recorded announcements. These are used to ring station equipment in completing calls and for notifying originating customers or equipment of each call's status. The audible ringing, busy and reorder tones are returned to the originator as notification that the directed destination is being rung, is busy, or that traffic or equipment conditions preclude reaching the destination. These tones are distinct sequences of precision tones for potential automatic recognition by station equipment.

Additional standard services, such as coin station and PBX calls, generally require additional switching functions and special electrical interfaces.

Coin service requires additional control and signaling functions to test for coin presence and to collect or return the deposit. No. 2 ESS is arranged to provide two versions of coin service. One allows dial tone before an initial deposit and free completion of operator or emergency calls. The other requires an initial deposit before any call can be originated.

In serving manual PBX systems, No. 2 ESS uses basic loop supervisory signaling on the group of lines to the PBX and also hunts for an idle member of the group when the directory number of the PBX is called. For dial PBXs and similar customer systems or equipment, a "ground start" supervisory signaling mechanism is utilized instead of a "loop" method in order to electrically exchange "busy-idle" and "start dial" state information with PBX equipment.

### 2.2.2 *Custom Services*

Custom services as supplements to the traditional standard services will also be available in No. 2 ESS to allow customers the option of further improving the efficiency and utility of their service.

*Touch-Tone*® calling requires the system to accept originations from such stations in addition to originations from conventional dial stations. When a *Touch-Tone*® telephone originates a call, the No. 2 ESS connects signal receiving equipment capable of recognizing frequency pairs.

Speed calling service allows a customer to specify a frequently used number by dialing only one or two digits instead of the full seven or more. For each customer who subscribes to speed calling, the system retains a list of abbreviated codes and the unique number assigned to each. The system also accepts dialed instructions from customers to add to or modify their own lists.

Customers who subscribe to three-way calling can add a third party to an existing connection by alerting the switching system with a supervisory flash (momentary on-hook), and then dialing the number for the added party.

A customer with call waiting service will, when he is engaged in a call, be notified with a tone if an additional call is directed to his number. He can then use a supervisory flash to alert the system to hold his original connection and connect him to the new call. Successive flashes will result in alternating connections between the two calls.

Call forwarding service allows a customer to direct the switching system to forward all calls for him to another number until he deactivates the service. Care has been taken in planning this service to reduce potential annoyance to other customers. The switching system automatically places a call to the forwarded number at the time the service is activated to encourage a customer to both verify the intended destination and to exercise the courtesy of notifying the other customer of the impending forwarded calls. In addition, as each subsequent call is forwarded the original number is rung briefly as a reminder that the service is active.

### 2.3 *Operational Features*

The significant operational features provided by No. 2 ESS are listed in Table II and are organized in three categories:

(*i*) Customer—for interchange of information with customers via their station or PBX equipment.
(*ii*) Intersystem—for operation with other switching systems.
(*iii*) Administrative—to interface with operating company personnel or with specialized systems where a function has been automated.

The manner in which the program uses customer class of service information, dialed digits, and the operational characteristics of equipment to implement these operational features is described in the following sections.

## III. SMALL OFFICE PROGRAM APPROACH

No. 2 ESS is expected to be used primarily in those Bell System central offices where the telephone company is more concerned about cost than about maximum traffic capacity. As a result, substantial savings are obtained by emphasizing call storage and program storage economy rather than call handling capacity. Also, the use of a simple processing hierarchy leads to additional storage economy at some expense in system efficiency.

### 3.1 *Use of Subroutines to Reduce Program Storage*

The subroutine is widely used in No. 2 ESS to keep program size small, even though some extra real time is used in preparing data and transferring to the subroutine. Since additional program economy can be obtained by nesting subroutines (subroutines calling other subroutines), the No. 2 ESS processor is designed to readily per-

TABLE II—OPERATIONAL FEATURES

*Customer*

Supervision: loop and ground start plus sleeve lead function.
Pulsing: dial pulse and *Touch-Tone*® calling with precision dial tone.
Ringing: 20 Hz from ac-dc or superimposed equipment.
Prepay and dial tone first coin service.
Reverting call sequences.
Hotel-motel message register signals.
Line polarity reversals for PBX toll diversion.
Busy verification and ringback for operator or local test desk.
Party identification and foreign potential tests.
Permanent signal and partial dial recognition.
Announcements (6 channels).

*Intersystem*

Supervision: loop and E&M
Outpulsing: dial pulse and multifrequency (13 digits) with delay dialing, wink start or stop-go controls.
Inpulsing: dial pulse and multifrequency (13 digits) with immediate dial or wink start control.
Automatic number identification information to an adjacent switching center.
Alternate routing and foreign area translation.
Operator switchboard or traffic service position interfaces with inband coin control.
Directory assistance (information) interface.
Test desk interface.
Automatic intercept system interface.

*Administrative*

Maintenance personnel interface: teletypewriter channel and local control panel for: (*i*) equipment status, (*ii*) detail trouble diagnosis and clearance and (*iii*) making lines, trunks, and service circuits busy for maintenance.
Automatic fault recognition and diagnosis of major components with automatic system reconfiguration to sustain service.
Measurements of service and performance characteristics: internally assembled and recorded via teletypewriter channels.
Interface for external service observations.
Changes to customer service information via teletypewriter channel.
Automatic message accounting data: magnetic tape recording or station identity transmittal to a central automatic message accounting center.
Line load control: self monitoring.
Emergency system recovery facility to restore call processing capability in the presence of errors.
Audits of call store records to assure consistency.
Automatic line insulation testing: remote control via teletypewriter channel.
Trunk testing: manual dial access to individual trunks, automatically sequenced outgoing tests and terminations for incoming tests.
Line testing: dial access.
Office alarm system interface: attended or unattended.
Automatic overload controls.

mit using several levels of nesting. Thus, the lowest level subroutine does a very specific task in a very straightforward manner. Higher level subroutines have larger tasks to perform but these tasks are accomplished primarily by moving data appropriately and calling lower level subroutines for further action. This structure leads to a basic call handling program that consists mainly of calls to subroutines.

### 3.2 *Economical Use of Call Storage*

Call storage economy is achieved through the use of small call records and by assuring that only truly variable information is stored. Small call records require efficient packing of information. Several different items are packed in the same word, even though the packing and unpacking takes extra time. Translation data concerning participants in a call is not kept in the call record and must be obtained from program store whenever it is needed. In general, any data that can be reconstructed from other information is not retained in call store.

Programs examine the call records and take action as a result of any new information plus the information contained in the record. Although it is the program which takes action, it is convenient to refer to the record as performing the function, and this convention is followed here.

### 3.3 *Simple Processing Hierarchy*

Further program and call storage savings are achieved through the use of a simple processing hierarchy consisting of a main program level, a timed interrupt program level, and a demand interrupt program level. Although the demand interrupt takes precedence over the timed interrupt, there are virtually no priorities within a given level. Instead, all tasks are handled as they are encountered and each program does as much work as possible on one job before going on to the next. The demand interrupt occurs only from an indication of machine failure or from a manual request. The timed interrupt is used for only those tasks where a small delay may cause an error or a failure. All other tasks are handled on the main program level and in a random sequence without regard for the urgency of the task. A more frequent interrupt would be required were it not for a wired logic unit in the processor which does the scanning for customer dial pulses in place of the stored program. The function of

this unit is explained in more detail in Section 5.2. This simple hierarchy saves program storage and call storage because as a task occurs it is performed regardless of priority instead of being passed to a task distributor program which would have to record details of each task for later execution. The only effect of this random sequencing is a slight, but unnoticeable delay for a few calls.

## IV. BASIC CALL PROGRAM SYSTEM

### 4.1 *Main Program Loop*

The overall program control plan is shown in Fig. 1. The call processing functions are controlled by the main program loop. One hundred milliseconds is the normal time required to complete one pass through this loop. To handle calls as quickly as possible, a new main program loop is usually initiated following the particular timed interrupt which delivers most new supervisory reports to the supervision distributor.

The 100 millisecond loop time represents a compromise between the speed with which the system can react to new inputs for individ-

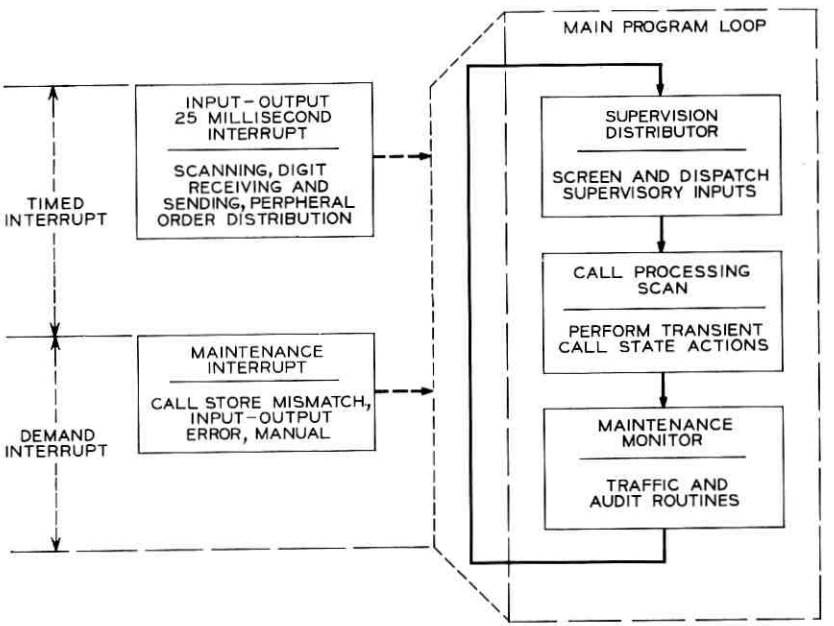



Fig. 1 — Program control plan.

ual calls and the efficiency with which it can process all calls. With a 100 millisecond loop interval, the amount of time wasted accessing calls that have no new information is less than 5 per cent while 100 milliseconds is the maximum delay before a new input is processed.

The supervision distributor is the first function in the loop. It distributes each new supervisory report to either the call record with which the report is associated or to a new call record if a supervisory report represents a new call.

The call processing scan accesses the transient call record associated with each nontalking call in the system. Any new information associated with a call is processed and an appropriate output response is generated.

The maintenance monitor is entered after all of the transient call records have been accessed. This program performs essential maintenance and administrative tasks including recording various traffic data, processing teletypewriter messages, and necessary audits to insure that the system is operating normally. Other maintenance tasks which may be deferred are handled as time allows.

### 4.2 *Input-Output 25 Millisecond Interrupt*

The timed interrupt used by call processing for input and output functions occurs every 25 milliseconds. The interrupt stops the main program, stores its program location and all machine registers in variable storage, and initiates those tasks required at that time. Although all of the tasks performed by the interrupt need to be performed more often than could be done by the main program loop, very few need to be done every 25 milliseconds. The various tasks are distributed among several interrupts at their required frequency as shown in Table III. This distribution of tasks attempts to equalize the amount of time required by each 25 millisecond interrupt. Notice that a few of the tasks must be executed every 25 milliseconds and these tasks determine the minimum time between interrupts.

#### 4.2.1 *Detection of Supervision*

The detection of supervision falls into several categories that require different frequencies of processing. The most frequent is associated with line service requests. A wired logic facility allows the system to detect one line service request at a time. Because one request must be processed before the next one may be detected, the line scanner check occurs every 25 milliseconds to avoid a momentary limitation on service requests. Program scans of trunk circuit and

TABLE III—FREQUENCY OF TASKS PERFORMED DURING TIMED INTERRUPTS

| Task* | See section | Input-output 25 millisecond interrupts (in milliseconds) | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 25 | 50 | 75 | 100 | 125 | 150 | 175 | 200 | 225 | 250 | 275 | 300 | 325 | 350 | 375 | 400 | 425 | 450 |
| Line scanner check | 4.2.1 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| Line hit list rescan | 5.1 | | X | | X | | X | | X | | X | | X | | X | | X | | X | |
| Fast trunk scan | 4.2.1 | | X | | X | | X | | X | | X | | X | | X | | X | | X | |
| Normal trunk scan | 4.2.1 | | X | | | | X | | | | X | | | | X | | | | X | |
| Trunk hit list rescan | 5.1 | | | | X | | | | X | | | | X | | | | X | | | |
| Tone digit check | 4.2.2 | | X | | X | | X | | X | | X | | X | | X | | X | | X | |
| Dial pulse digit check | 4.2.2 | | | | X | | | | | X | | | | | X | | | | | X |
| Sending check | 4.2.2 | | X | | X | | X | | X | | X | | X | | X | | X | | X | |
| Multifrequency digit turn off | 5.2.2 | | | | | X | | | | | | X | | | | | | X | | |
| Peripheral order distribution | 4.2.3 | X | | X | | X | | X | | X | | X | | X | | X | | X | | X |
| Teletypewriter input-output | 4.2.4 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| Automatic message accounting output | 4.2.4 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| Maintenance functions | 4.2.4 | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |

* Task is performed during interrupts marked X.

service circuit supervisory indications occur at either a 50 or 100 millisecond rate.

The 50 millisecond rate of the fast trunk scan is used for those trunks that can generate supervisory signals that last less than 100 milliseconds, such as an operator trunk which can generate an on-hook wink of between 70 and 130 milliseconds. In order to insure detection of this wink, these circuits must be scanned at least once every 70 milliseconds; so to be compatible with the system interrupt structure, they are scanned every 50 milliseconds.

Also scanned during the fast trunk scan are trunks from step-by-step switching systems which can present a seizure to the system followed almost immediately by customer dialing. The system must scan these trunks often enough to insure detecting the seizure before the first dial pulse comes from the customer. All other supervisory indications are scanned using the normal trunk scan at a 100 millisecond rate which represents a compromise between system processing efficiency and system response time to a given change of supervision.

#### 4.2.2 *Digit Receiving and Sending*

Digit receiving and sending involves the receipt of dial pulse, tone, and multifrequency digit information, and the sending of dial pulse and multifrequency digits. Each one of these functions has its own set of timing constraints which must be adhered to. The timing sequences involved in the digit processing facility represent a combination of all these timing constraints.

Since the minimum time between tone or multifrequency digits is about 80 milliseconds, the program does a tone digit check every 50 milliseconds. At the same time it must be capable of receiving dial pulse digits, since at the time a party requests service it may not be known whether he will send dial pulse or tone digits. Dial pulse reception requires the discrimination between a disconnect from the customer and a dial pulse. Further, the system must discriminate between the off-hook periods separating two dial pulses and the off-hook period following the end of a dial pulse digit. Neither the on-hook nor the off-hook portion of a dial pulse should exceed 125 milliseconds.

A disconnect signal or the off-hook time between two digits should always exceed 250 milliseconds. Therefore, the system does a dial pulse digit check every 125 milliseconds. It concludes that the subscriber has disconnected if he is on-hook for two successive dial pulse

digit checks without having been off-hook in between. It concludes that the subscriber has finished dialing a digit if he is off-hook for two successive dial pulse digit checks without being on-hook in between.

For sending operations, multifrequency digits require a 150 millisecond cycle and dial pulse digits require a 100 millisecond cycle. The greatest common multiple, 50 milliseconds, is thus used as the interval for performing a sending check for any sending operation which may be required.

#### 4.2.3 *Peripheral Order Distribution*

Distribution of orders to peripheral circuits during the timed interrupt is necessary to assure efficient use of peripheral control circuitry. Because sequences of orders are often involved in setting up or changing paths through the network, the total time required for sequences must not exceed the maximum time permitted for the temporary interruptions of connections. Therefore, peripheral order distribution occurs every 50 milliseconds and, where appropriate, circuits are interrogated for responses from previous orders at the same time.

This 50 millisecond rate is chosen to coincide with the operate time of the network control, which requires slightly less than 40 milliseconds to complete its most complicated action. Since it can take up to 8 milliseconds to complete peripheral order distribution during any one period, the 50 millisecond rate of sending these orders assures that the network controller used at the end of one series of distributions is available for use at the beginning of the next series.

#### 4.2.4 *Other Interrupt Functions*

Other functions performed during the timed interrupt include teletypewriter input-output, automatic message accounting output, and several maintenance functions. The teletypewriter input-output work is handled once every 25 milliseconds during the interrupt to assure that no characters typed by the craftsman are missed and that the output typing rate is constant. Automatic message accounting output data are distributed to an incremental tape recorder every 25 milliseconds to assure an adequate capacity with a maximum call rate system.

Maintenance functions handled during the timed interrupt are those routine tests which require extremely critical timing. Although routine maintenance is normally considered quite low priority, some tests which involve hardware response times require critical timing once the test is initiated. This timing is so critical that any time varia-

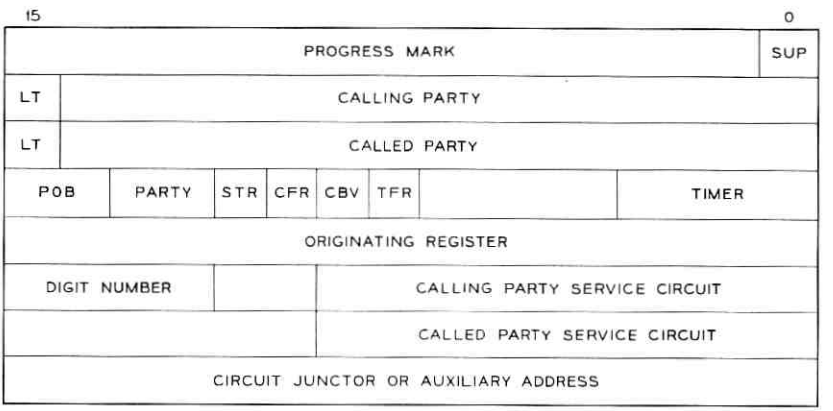
tion resulting from the amount of work done during the interrupt cannot be tolerated. Therefore, these maintenance functions are performed at the beginning of the interrupt if they are needed.

### 4.3 *Call Processing Control*

The primary call processing control in No. 2 ESS rests with the set of programs associated with the transient call records. A transient call record is associated with each active nontalking call in the system and consists of 8 call store words, 16 bits wide as shown in Fig. 2. The transient call record controls the progress of the call for the interval starting with dialing through ringing until an answer signal has been received and the connection is in the talking state. There are up to 256 of these transient call records in a No. 2 ESS, each independently handling its own telephone call. Since the transient call record handles calls in all possible states, it is not practical to have a fixed layout of data within the entire eight-word block. Therefore, Fig. 2 serves as an example of a typical layout.

#### 4.3.1 *Progress Marks*

The first word of the transient call record contains the address of the program used by the processor to reinitiate the processing of a call on each main program loop. This program address is called a




SUP = SUPERVISION FLAG
LT = 1 IF PARTY IS A LINE,
0 IF A TRUNK
PARTY = 2-PARTY LINE IDENTITY
STR = SECOND TRY IN PROGRESS
CFR = CONFERENCE IN PROGRESS
CBV = CHANGE SPEED CALL LIST
TFR = TRANSFER CALL
POB = PERIPHERAL ORDER BUFFER

Fig. 2 — Transient call record.

progress mark and it must be provided whenever the transient call record is in use.

For example, if a party is dialing, the program addressed by the progress mark will inquire if any new digits are present or if the customer has delayed too long between digits. During the main program loop each transient call record is accessed and control is transferred to the program indicated by the progress mark, and that program then inquires if there is any new information associated with the call. If there is nothing new, the same progress mark is left in the transient call record and the next transient call record is accessed. When there is new information, the program acts upon that information and generates any output data required. If the completion of this action changes the state of the call, a new progress mark is written into the transient call record to reflect the new state.

The process of moving from one transient call record to the next is handled by a special advance instruction in the No. 2 ESS processor. This instruction relies upon the fact that all transient call records are adjacent to each other and all of them begin at a location whose low three bits are all zeros. In addition, a particular machine address register is used by the transient call record programs for reading and writing data in the transient call record, so that it always points at some address within the eight words of the transient call record.

When the advance command is executed, the next transient call record is determined by first zeroing the low three bits of the address register and then adding eight to the resulting address. The register then points to the first word of the next transient call record. The processor reads the progress mark and transfers control to the indicated program. Since the advance command performs this entire function, it is not necessary to actually return to the main control program between the processing of two transient call records. The program is not concerned with the absolute location of the transient call record, so that data is accessed simply by modifying the low three bits of the address register.

#### 4.3.2 *The Transient Call Record Layout*

The second and third words of the transient call record almost always are used to hold the calling and called party's identification. The high bit of this word determines whether the party is a line or a trunk and the other 15 bits specify a line's terminal number or a trunk circuit's scan point number. The No. 2 ESS switching network is limited to fifteen 2,048 terminal networks or 30,720 terminals, and,

therefore, 15 bits can be used to define any line terminal. It has been determined that twelve 1,024 point scanners are adequate to handle all trunks and service circuits in a No. 2 ESS office, and fourteen bits are used for this purpose. The use of the second and third words of the transient call record is limited to defining the calling and called parties so that tracing programs can find the transient call record associated with a particular line to determine information about the state of a call or to set up a no test connection.

The layout of the remaining five words of the transient call record are somewhat less standardized. The fourth word usually contains a set of flag bits which indicate various substates that do not affect the normal flow of call processing, but permit the same programs to handle several different situations. In addition, this word contains four bits which are used for various timing functions and two bits which are used for sequencing the delivery of orders to the periphery. The fifth word contains the address of the originating register, which receives and stores digits from the customer and also handles the sending of digits to another central office.

The low ten bits of the sixth word are used to store the identity of any service circuit which may be connected to the calling party identified in the second word. Likewise, the low ten bits of the seventh word contain the identity of a service circuit associated with the called party identified in the third word. The eighth word contains the identity of a circuit junctor used to connect the calling and called party. When an originating register or service circuit or circuit junctor is not required, these words may be used to store other useful information which aids in processing the call. For example, during dialing, when no circuit junctor is required and no service circuit is associated with the called party, class information for the calling party is stored in the seventh word and an auxiliary address is stored in the eighth word. This address is used in the processing of dialed digits.

## V. PRIMARY PROCESSING FUNCTIONS

The programs used to process transient call records are called progress mark routines and they form the backbone of the call program structure of No. 2 ESS. All supervisory inputs are delivered to the transient call record, where they are interpreted by the progress mark routines. All new digits are examined and interpreted by the progress mark routines. These routines call various subroutines for purposes of

translating information, selecting equipment, providing timing, and sending orders to peripheral equipment to complete connections for customers. The following sections describe most of these functions in detail, showing the relation among various circuit functions, interrupt program functions, and progress mark routines.

Figure 3 shows the basic information flow of the call processing system and indicates the relation between circuitry and program. It shows how information comes into the system from peripheral circuits through both wired logic in the processor and programs executed during the timed interrupt. The information thus gleaned is passed on to the progress mark routines through several different paths. The progress mark routines which are executed at the base level examine this information using various subroutines, and pass orders to the interrupt program for distribution to peripheral circuits. When this distribution has been completed, the interrupt program notifies the progress mark routines so that a new state of the call may be recognized.

### 5.1 *Processing of Supervision*

#### 5.1.1 *Line, Trunk, and Service Circuit Scanning*

Two types of supervisory scanning are provided in No. 2 ESS. Line scanning detects requests for service from customer lines. Trunk and service circuit scanning detects changes in supervisory states associated with calls in progress and detects requests for service from incoming trunks. Trunk and service circuit scan points provide more changes of supervision than do the more numerous line scan points. These scan point changes may be of shorter duration and may require more rapid system response than line scan points. These characteristics lead to two entirely different methods for scanning.

No. 2 ESS line scanning takes advantage of the fact that a line's request for service is always an on-hook to off-hook change. Once a request has been served, the line ferrod is disconnected from the line, causing the scan point to go on-hook and remain on-hook for the duration of the call. When the ferrod is restored to the line at the end of a call, the customer is on hook so that the scan point continues to indicate on-hook. As a result, only those scan points which represent unserved customer requests for service are off-hook at any time. Once properly initialized, an input-output circuit in the No. 2 ESS processor automatically cycles through all of the line scan points row by row and looks for any lines that are off hook. When it finds an off-hook,

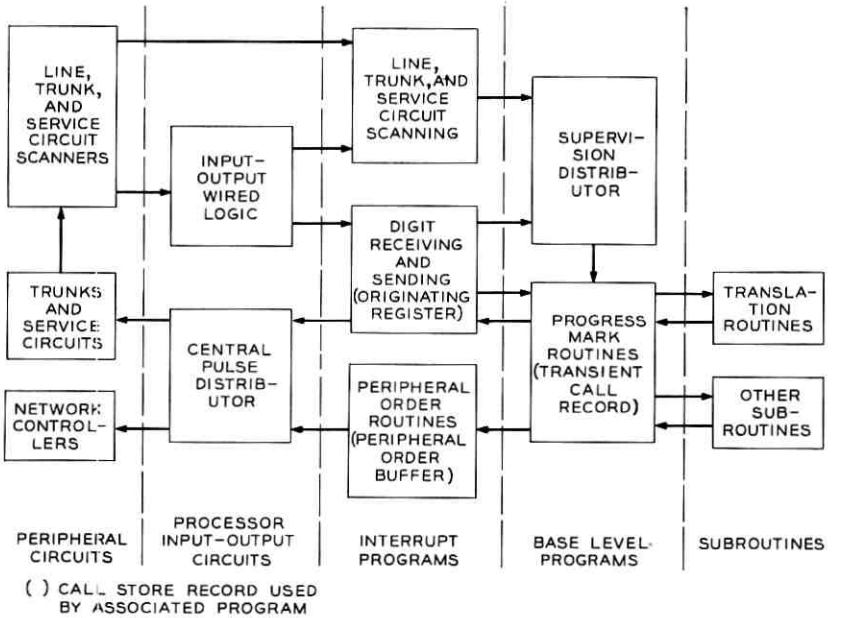



Fig. 3 — No. 2 ESS call processing information flow.

the circuit stops cycling and indicates the scanner row containing the line service request. Later the circuit is interrogated by program, the row identity recorded in a hit list for later use, and the circuit is reinitialized to start cycling from that point.

Since the significant supervisory changes from trunks and service circuits can be either off-hook to on-hook or on-hook to off-hook, it is necessary to scan these points by comparing the present state of the point with the last-look bit in call store which represents the previous state. This scanning is done in the input-output (timed) interrupt as described in Section 4.2.1.

For many reasons, supervisory changes lasting only a few milliseconds can occur and do not truly reflect a valid change in the state. On some occasions it is possible for the trunk scanning program or the line scanning circuit to detect these momentary hits. To reduce the number of hits passed on to the processing program, the address of the scanner row in which a change is detected, is stored in a hit list. 50 milliseconds later this same row is again scanned by program and if a change still exists, then a valid supervisory change is assumed to have occurred.

Table III shows the relative times of the line and trunk hit list rescans. The probability of a hit plus a valid change occurring in the same row at the time of the rescan is very low. The occasional hit that does get reported will be eventually ignored by the processing programs but it does require some additional processing time. Once a valid supervisory change has been detected, the scan point number is placed in one of three lists for eventual processing by the supervision distributor. There is a separate list for reporting line origination requests (origination hopper), trunk and service circuit off-hook reports (off-hook hopper), and trunk and service circuit on-hook reports (on-hook timing list). The latter two lists are also used by the supervision distributor for further timing of the supervisory changes.

The supervisory scanning programs include several defensive programming features which detect failures in the scanner circuits which may not be detected by other means. On the average, only about two scan point changes should occur in a 1024 point scanner during any single 100-millisecond program scan. Therefore, if many more than this occur, there is adequate reason to suspect a failure in the scanner and a more detailed diagnostic program is called to determine whether this occurrence is actually a failure or just a momentary peak in the number of scan point changes.

#### 5.1.2 *Supervision Distributor*

The supervision distributor program which functions in the main program loop has two primary tasks to perform. First, supervisory changes detected in the interrupt program are timed and compared with other supervisory reports to further eliminate momentary hits and to also discriminate between flash signals and valid on-hooks. This timing function is only associated with trunk and service circuit scan point changes since no on-hook changes can be reported for line scan points. The two lists associated with this timing function are the off-hook hopper and the on-hook timing list.

All valid on-hooks must last for at least 250 milliseconds before they are reported. Therefore, the on-hook timing list is checked every main program loop for all scan point changes that have been in the list for longer than the 250 milliseconds. Any off-hook scan point change that follows an on-hook change by less than 250 milliseconds is a trunk flash. The flash is detected by matching all of the scan point numbers in the off-hook hopper against all of the scan point numbers in the on-hook timing list before the off-hooks are further

processed. When a match is detected, a flash report is processed for that scan point rather than an off-hook.

The second task of the supervision distributor is to determine the state of the circuit which generated the report. There are three possible states that can occur: (*i*) no call in progress, (*ii*) transient or nontalking state, and (*iii*) stable, or talking state. The state of each circuit is recorded in a two-word call store record, called the terminal memory record associated with that circuit. If the call is in a transient state, the address of the transient call record associated with the call is stored in the terminal memory record as shown in Fig. 4.

Two bits are also stored in the terminal memory record to indicate its particular function in the transient call. To report a change, the supervision distributor accesses the transient call record indicated in the terminal memory record and checks the contents of the progress mark word. If the low bit of this word is a 1, the program changes that bit to a 0, writes it back into the progress mark word and

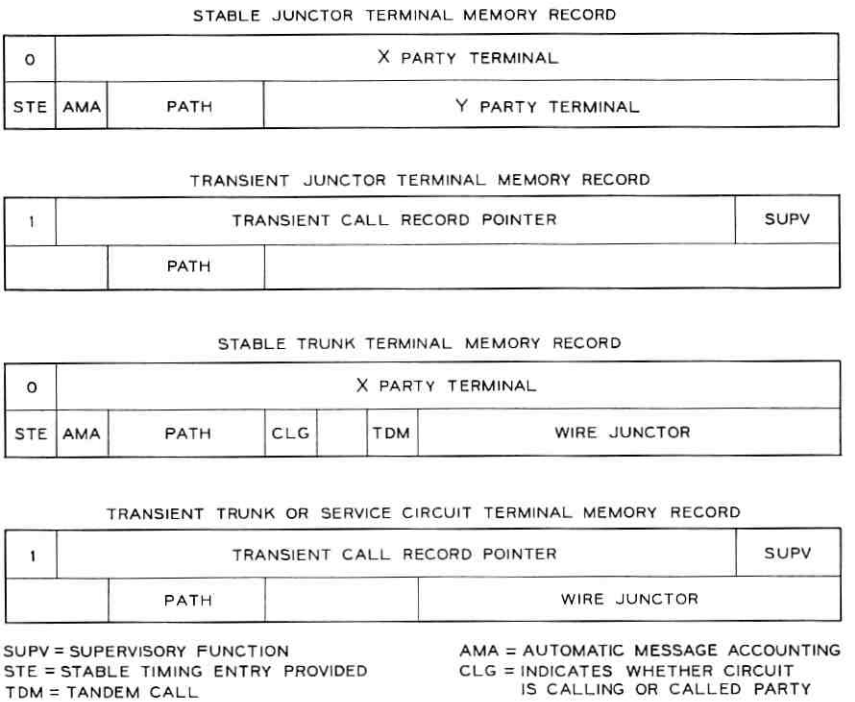



Fig. 4 — Terminal memory record formats.

also writes a five-bit code elsewhere in the transient call record to indicate the specific details of the supervisory change. If the low bit of the progress mark is a 0, supervision cannot be reported to the transient call record and the program moves the change into a queue for later distribution to the transient call record. This procedure assures that only one supervisory report will be delivered to the transient call record at a time, and permits the transient call record to block any supervisory reports simply by using a progress mark with its low bit 0 during those intervals when it would be impossible to properly interpret the supervisory report.

In other instances, if the terminal memory record indicates that the state of the circuit is idle or stable, the supervision distributor selects a transient call record and places the change report and the circuit identity in it. It also inserts a progress mark to address a program which will appropriately decode the supervisory change either as a request for service or as a disconnect signal in the case of a stable call. The terminal memory record is also updated to indicate its new transient state and the identity of the transient call record.

Line service requests are handled by the supervision distributor through the line origination hopper. Since lines have no terminal memory records, and since the only changes in this list represent requests for service, the supervision distributor attempts to select a transient call record for each service request in this hopper. Any unsuccessful attempts will leave the request in the hopper for further attempts. If a transient call record is selected, the line's terminal equipment number is stored in the second word and a progress mark is placed in the first word. The program addressed by the progress mark will later attempt to connect the line to a digit receiver.

Flashes from trunks are detected by the supervision distributor, but flashes from customers are detected by progress mark routines which time the on-hook period after the change is reported to the transient call record. This method of handling customer flashes is used because only a few customer classes of service permit flashes and there are only a few states where a flash is treated differently from an on-hook followed by an off-hook. Only in those instances is timing necessary to determine if the customer is flashing.

### 5.2 *Processing of Digits*

All digit receiving and sending functions in No. 2 ESS use an originating register and a set of associated programs under direction

of the progress mark routines (see Fig. 3). The originating register layout as shown in Fig. 5 consists of eight 16-bit call store words. The lower four words of the register contain space for storing up to 16 digits, each digit being represented by a four-bit binary coded decimal code. The upper four words of the originating register contain control data for the interfaces among the wired logic digit handling facility, the timed interrupt digit handling programs, and the main program level progress mark routines. The system has a capacity of up to 128 originating registers, but only one is associated with a particular call.

### 5.2.1 *Receiving*

There are three types of digits that must be received—dial pulse digits from lines or trunks, tone digits from lines, and multifrequency

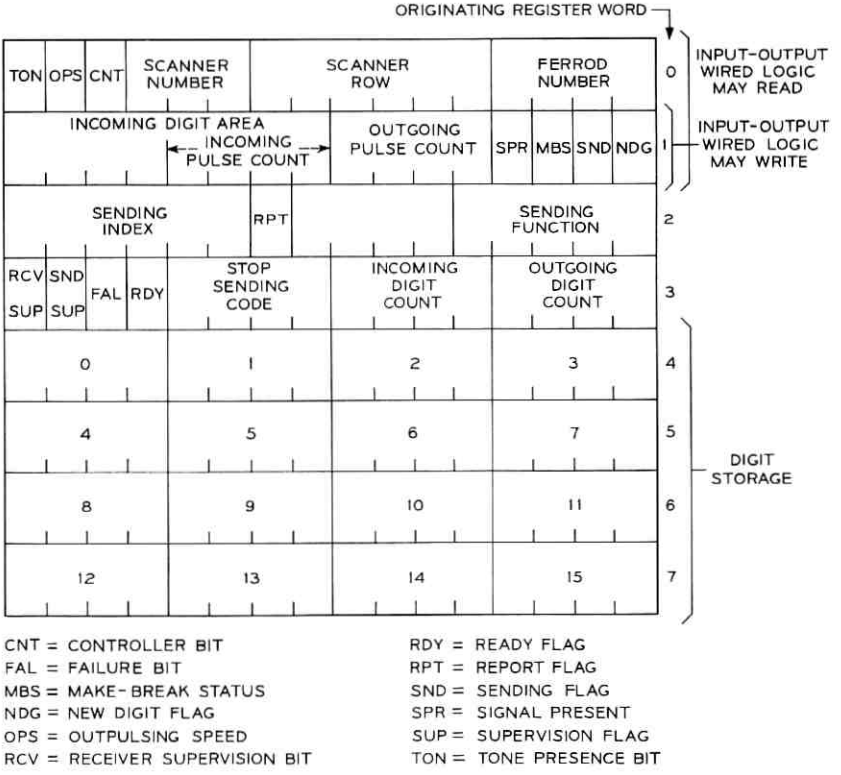



Fig. 5 — Originating register.

digits from trunks. For dial pulse digits, the originating register is used to count the pulses detected by a scan point. In the case of tone or multifrequency digits, the receiving circuit indicates through a set of scan points which tones are present and through a signal present scan point that a proper combination of signals has been present for a minimum time. The receiver circuit may be arranged to permit both tone and dial pulse digits to be recognized, thus permitting parties on a party line to have both rotary dial and *Touch-Tone*® telephones.

A wired logic facility in the No. 2 ESS processor scans the originating registers at a 10-millisecond rate. Each time an originating register is accessed by the wired logic, the first word is read and the indicated row of 16 scan points is examined. In the case of dial pulse reception, the status of a single scan point is compared with the make-break status bit, and if a change from off-hook to on-hook has occurred, the bit is updated and the low bit of the second word, the new digit flag, is set to 0. If a change from on-hook to off-hook has occurred, the make-break status bit is updated, the new digit flag is again set to 0 and the incoming pulse count is incremented by 1.

In order to detect a completed digit, or a disconnect, the interrupt processing program scans through all the originating registers every 125 milliseconds and sets the new digit flag to a 1. If this flag remains 1 for another 125 milliseconds, and if the make-break status bit indicates an off-hook, the program assumes that a new digit is completed. If the make-break status bit indicates on-hook, then the program concludes that the line has disconnected.

When a digit is completed, the program removes it from the incoming digit area, stores it in an appropriate digit slot of the originating register and then zeros the incoming digit area. As long as the incoming digit area is all zeros and the digit receiver scan point used to supervise the call is off hook, the wired logic continues to zero the new digit flag, thus preventing the program from examining the incoming digit area when it is empty.

Tone and multifrequency digits are recognized by scanning and comparing the signal present scan point with the signal present bit. When a new signal is present, the states of six scan points representing the multifrequency digit or eight scan points representing the tone digit are recorded in the incoming digit area and the new digit flag is set to a 1. Since multifrequency and tone digits may come as often as once every 80 milliseconds, the program examines the originating registers once every 50 milliseconds. If a new digit is present, it is

removed from the incoming digit area, converted to a binary coded decimal representation and stored in an appropriate digit slot in the last four words of the originating register.

The position for storing a completed digit is determined by the four-bit incoming digit count in the fourth word of the originating register. The program that moves a new digit in place determines its location from the incoming digit count which is then incremented to provide a pointer for the next digit. The program also sets the transient call record ready flag in the fourth word of the originating register. Approximately every half second the transient call record controlling a dialing call checks the originating register for the ready flag. If the flag is not set a time check is made to avoid tying up common equipment. If the flag indicates that a digit is present, the progress mark routine checks the incoming digit count and if enough digits have been received, the progress mark routine will then perform any appropriate action.

For example, after three digits have been received, the progress mark routine will attempt to determine the routing of the call and if it is a local call, after seven digits have been received, it will attempt to set up the appropriate ringing connection. Whether or not the transient call record actually interprets the digits, it resets the ready flag in the fourth word so that it will not have to check the originating register again until another digit is received, thus minimizing the amount of time spent checking until there is new information available. In the case of receiving multifrequency digits, the transient call record ready flag is not set when each digit is received; rather, it is set only when the start digit is received by the wired logic indicating that all of the digits are present. This arrangement is provided because it is not practical to translate the digits as they are received since the total number of digits can be variable.

### 5.2.2 *Sending*

There are only two types of digits which may be sent, dial pulse and multifrequency. To insure a high degree of precision, the timing functions needed to send these digits are under the control of the processor. Timing for output dial pulses is provided by a wired logic facility which distributes start and stop pulses over a timing bus to the dial pulse sending circuits.

Sending is initiated from a transient call record by a progress mark routine after a sender circuit has been connected to a trunk. At that point a sending index is inserted into the third word of the originating

register along with the sending function. The progress mark routine places the identity of the first digit to be sent in the outgoing digit count area and the last digit to be sent in the stop sending code area.

The sending index points to a call store list that contains the scan point number and peripheral decoder* address of the sender circuits. The scan point number is needed by the interrupt program to determine if a start sending signal has been received. In the case of dial pulse outpulsing, when the start sending signal is received, the interrupt program moves the digit indicated by the outgoing digit count into the outgoing pulse count area of the originating register and then increments the outgoing digit count by 1. The program also sends a peripheral decoder order to operate relays in the sender to start sending dial pulses. Every 100 milliseconds the wired logic decrements the outgoing pulse count by 1 until that count reaches 1. At that point it sets the sending flag so that the interrupt program may inform the sending circuit to stop sending pulses.

After turning off the dial pulses the interrupt program loads a constant into the outgoing pulse count to time the interval between digits so that the sending flag will be raised when it is time to send the next digit. The sequencing of functions required to time digit and interdigit intervals is controlled by the sending function identified in the third word of the originating register. Since both the timing of the dial pulses and the timing of the originating register scans are under control of the processor, it is possible to guarantee that the starting and stopping of dial pulse digits occurs during the off-hook portion of the dial pulse cycle so that no short or long pulses occur.

In the case of multifrequency sending, the entire digit is sent to the sending circuit by the interrupt program and the wired logic facility is only used to time the length of the digit. All multifrequency digits except the key pulse digit are sent in synchronism for all originating registers and are turned on for 75 milliseconds and then off for 75 milliseconds. (The key pulse digit, the first digit sent, must remain on for 100 milliseconds.) While sending, the interrupt program accesses the originating register every 150 milliseconds, reads the next digit and sends peripheral decoder orders to the sender to turn on the pair of tones representing the digit. The number of the sender is then placed in a special list. 75 milliseconds later this special list is examined by the interrupt program and peripheral decoder orders are sent to all senders in the list to turn off any multifrequency tones they are send-

* A peripheral decoder is a shift register circuit used to receive serial messages from the processor and operate relays in trunk and service circuits. (See Ref. 3 for details.)

ing. The timing for multifrequency digit turn off is shown in Table III.

In order to send the proper number of digits, the program, before it sends a new digit, compares the stop sending code and the outgoing digit count to see if it has reached the last digit. If not it checks to see if another digit has been received as indicated by the incoming digit count. This procedure is used to permit overlap outpulsing. When the last digit has been sent the ready flag is set to indicate that fact to the transient call record.

For program simplification, the stop sending code is used throughout the processing of dialing by the progress mark routines to indicate the total number of digits expected. This arrangement permits the interpretation of digits to be handled independently from any prefixing digits that may have been received, since the position of the digits to be interpreted is determined relative to the stop sending code rather than from an absolute position within the originating register.

### 5.3 *Peripheral Order Programs*

Network connections are set up and torn down by means of peripheral order buffer programs which execute during the timed interrupt. Using information recorded in 16 word buffers, these programs send orders to network controllers, peripheral decoders, and scanners in order to perform actions needed by base level call and maintenance programs. Because a network controller requires 40 milliseconds to execute an order from the central processor, peripheral order buffers are examined every 50 milliseconds by the peripheral order buffer execution program. This execution frequency guarantees that all network controllers are idle at the beginning of the peripheral order buffer execution program and makes efficient use of them.

The sixteen call store words provided for a peripheral order buffer allow capacity for storing information (scan points, peripheral decoder address, paths) for up to three circuits. The most typical peripheral order buffer action disconnects a terminal from one circuit and connects it to another. The majority of these actions can be handled by one peripheral order buffer. Actions involving more than three terminals require the use of more than one peripheral order buffer. Other actions provided by the peripheral order buffer execution programs include:

(*i*) Simple disconnects: Trunk, service circuit, or junctor circuit cut through contacts are opened and the line ferrods of any lines involved in the call are restored by closing the line cut-off ferreeds.

(*ii*) Simple connects: A network connection is set up between two

terminals and circuits associated with one or both terminals are placed in an initial state.

An action taken by the peripheral order buffer execution program is accomplished by executing a sequence of generalized tasks which include sending a network order, sending a peripheral decoder order, scanning a ferrod, or delaying before another task is attempted. Use of the orders may be illustrated by the specific orders for the action which connects dial tone to a line. (See Ref. 3 for details of circuit operation). The orders are:

(*i*) Perform a false cross or ground check and then connect the digit receiver terminal to one terminal of the specified wire junctor.

(*ii*) Perform a false cross or ground check and connect the other terminal of the wire junctor to the line, opening the line ferrod cut off contacts (delay 50 milliseconds).

(*iii*) Perform a power cross test on the line (delay 50 milliseconds).

(*iv*) Operate the cut through contacts in the digit receiver by means of a peripheral decoder order to connect the digit receiver to the line.

(*v*) Delay 50 milliseconds (1 peripheral order buffer cycle) to allow relay operating transients to decay.

(*vi*) Scan the digit receiver line supervisory ferrod to check by means of an off-hook transition that there is network continuity.

(*vii*) Place the digit receiver relays in the dial tone state.

The peripheral order buffer execution programs actually execute other tasks than those illustrated, but the main body of the peripheral order buffer actions are composed of the tasks illustrated.

Two methods are used by the program to execute the 300 different peripheral order buffer actions currently required. Most actions do not occur often so a method has been devised to allow a small program to handle these actions at the expense of some system real time. These slower actions are described by data words stored sequentially in memory which are decoded one at a time by the program to determine the sequential tasks needed to complete each peripheral order buffer action. The sequence needed for a particular action is defined by storing an index to the sequence table in the peripheral order buffer when it is selected and initialized. The sequence is terminated by an end code.

In addition to sequence tables, trunk state tables are used by the peripheral order buffer programs to determine which trunk relays must be operated to place the circuit in a desired state. These tables also indicate how ferrods are assigned for a trunk, and describe to the peripheral order buffer execution program which ferrods to scan for

continuity checks, and other scan tasks. When the peripheral order buffer is initialized, a trunk state table pointer is stored in the peripheral order buffer for each trunk. The state table is then indexed from the starting location defined by the pointer to determine how the circuit relays must be operated to reach a particular trunk state. Use of the circuit state table allows only one task sequence to be used to describe a given peripheral order buffer action for all trunk types. This technique also yields good flexibility for the incorporation of new trunk types or unforeseen peripheral order buffer actions.

A second method is used for performing frequently needed peripheral order buffer actions. This method uses direct program code rather than sequence tables and is used primarily for network actions needed for setting up intraoffice, incoming, and outgoing calls. Also included in this category are all simple disconnects and actions which may be completed exclusively by means of peripheral decoder orders. By saving the time required to address and decode the contents of words in the sequence table, the real time consumed by these frequent actions may be minimized. Combination of the two methods of executing peripheral order buffer actions permits compromises between system real time and program size.

A second body of routines called the peripheral order buffer loading program is used to handle peripheral actions. This program is a collection of subroutines which interfaces with base level call and maintenance programs to load the information into a peripheral order buffer which is needed by that buffer's execution program. When called, the loading program selects an idle peripheral order buffer and, because the transient call record rarely contains information which is directly usable, performs translations upon trunk scan points and other information recorded in the transient call record to obtain peripheral assignment information. The terminal memory records used by a transient call record are also addressed to retrieve path information which must also be interpreted before it can be used by the peripheral order buffer execution program.

When a base level transient call record program requires a peripheral order buffer action, the type of action as well as the lines, trunks, and service circuits to be used in the action must be specified to the loading routine. A calling sequence is used which consists of the instruction which transfers control to the loading program entry point, one or more data words, and a failure and a success return. The data words of the calling sequence are decoded by the loading program to determine what action is requested.

Several important features are provided by the peripheral order

buffer execution programs which have not previously been mentioned. These include blocking unwanted supervisory reports caused by network and circuit actions, minimizing open intervals on the transmission paths of established calls and retrying failing network orders.

Supervisory signals are generated by setting up and tearing down network paths which are important for checking network continuity but which serve no other useful purpose. An example of such a signal is the off-hook which results from connecting a line to a circuit junctor or a trunk. The line is known to have previously been off hook and an off-hook report adds no new information. If the scanning programs are allowed to report the off-hook, however, approximately 1 millisecond of real time is required to discard it. For this reason the peripheral order buffer execution program changes the state of the last look bit kept for the ferrod in call store when a continuity check reveals that it has changed state. The scanning program, which executes after the peripheral order buffer execution program, detects no change of state.

Open intervals in the transmission path are caused by peripheral order buffer disconnect-connect actions after a connection has been established. When such an action involves a line, an advance network controller reservation technique is used to limit the open interval to 150 milliseconds. Since the connect action consists of setting up two half paths in the network, a half path is first connected for the circuit to which a line is being connected. The network controller needed to connect the line is then reserved by marking its status bit busy in memory. The cut through relay of the circuit from which the line is being disconnected is opened and the execution is terminated until the next peripheral order buffer cycle.

During the next peripheral order buffer execution cycle (50 milliseconds later) the reserved network is used to connect the line to the new circuit and 50 milliseconds later the cut through relay of this second circuit is closed by a peripheral decoder order. Failure to find an idle network controller for use in connecting the line causes the peripheral order buffer execution to delay opening the line's transmission path until an idle controller becomes available.

The peripheral order buffer execution program works in conjunction with a "working mode program" which executes just prior to the peripheral order buffer execution program. By scanning the F, S, and T scan points of the network controllers, the working mode program is able to detect if any network controllers failed network orders during the previous peripheral order buffer execution cycle. When a

failure indication is found, it is the function of the working mode program to record the failure and cause all orders sent during the previous peripheral order buffer execution cycle to be retried. A second failure causes the failing peripheral order buffer to be located by means of a special peripheral order buffer execution cycle in which orders sent during the previous cycle are formed and examined to determine which order used the failing controller. Location of the failing order and the peripheral order buffer from which it was sent allows the order to be examined for correctness and triggers maintenance actions which attempt to find a system configuration which can communicate the failing order to the network controllers. A series of up to three retries can occur and success at any stage allows the peripheral order buffer to continue its sequential tasks.

### 5.4 *Translation*

Translation data occupies a large percentage of program storage in all No. 2 ESS offices. With less than one module (16,384 words) required in the smallest offices, storage for translation information must be capable of growing to more than eight modules (131,072 words) for larger offices. Efficient use of translation storage is necessary to achieve low system cost.

The simplest and fastest translator would be one which uses the number to be translated (terminal equipment number, directory number, trunk scan point number, and so on) to directly index storage to obtain the desired data. This approach is wasteful of storage in almost all cases. In a small office (for example, 4000 lines) only $\frac{1}{5}$ to $\frac{1}{4}$ of a maximum sized translator would be filled and rarely would it be completely filled.

To achieve translation storage efficiency and ease of growth in No. 2 ESS, the major translators have been divided into blocks of translation words, called subtranslators. A portion of the number being translated is used to address an entry in a master table index and retrieve the starting address of the desired subtranslator. The remaining portion of the number is used to index to the desired entry in the subtranslator. The word located by this procedure may be either the output data or a pointer to the expanded data for the more unusual cases.

In No. 2 ESS even parity is used on data words and odd parity is used for program instructions. Encoding parity in this way allows the processor to detect and prevent erroneous use of translation words as instructions or instructions as translation data. When a parity

error is detected, a duplicate processor is switched on line in an attempt to continue service. Should the error occur in translation data, it may cause two or more processor switches and require system emergency action to recover its processing capability.* A primary cause of this type of error is improper accessing of storage resulting from the use of incorrect translator indexes.

Protection against such errors is built into most of the translation subroutines. To implement this protection, additional information describing each subtranslator is included in the master table index. These data describe both the number of words per subtranslator entry and the maximum index value for the subtranslator. Each use of the translator can then be checked to insure that the index used to access the subtranslator remains within its boundaries. Errors detected by this check cause a teletypewriter printout and cause error indications to be returned to the calling program.

Another potential source of errors in translation data is service order changes which must be introduced into the system by the telephone company. Errors introduced into certain areas of the translation information such as the master table index can be catastrophic; for this reason, no facility for altering such storage is provided via teletypewriter. These critical items may be changed when necessary through use of the office data assembler program to recompile translations.[6]

Several strategies are invoked in an attempt to guarantee the integrity of service order information typed into the machine for lines and trunks. One example of these strategies forces service orders for line changes to include both the equipment number and directory number of the line. The redundancy in the input data allows simple but powerful checks to be made before using the change data. Insuring that service order information is independent of program store addresses and internal storage formats also helps to minimize errors in input information.

#### 5.4.1 *Originating Translation*

The line origination translation provides a conversion from the line equipment number to the line class, service data and billing number. Each line in the No. 2 ESS office has a terminal equipment number which is used to refer to that line during the processing of its calls. The class data derived from the line origination translation includes

* System emergency actions are program actions which attempt to eliminate the call or calls causing system errors.

a major class indicating the type of service such as individual, two-party, PBX, and so on; a screening class which is used to determine routing and charging for calls dialed by this line; and various class bits which indicate the type of dialing and the priority of service. This translation is used when a digit receiver is selected, when a billing entry is prepared for automatic message accounting, when various special service requests occur, and when special control functions are required.

The first step in the translation shown in Fig. 6 involves the use of the high six bits of the equipment number to select the terminal equipment number translator associated with the network on which the line appears. This translator has a one-word entry for each terminal equipment number, and the low nine bits of the number indicates this entry. In a majority of cases, this translator contains an abbreviated class indication for the line and the line's billing number (directory number). The five-bit abbreviated class represents the most common combinations of service features in the office. It is expanded in another translator to indicate all of the class of service information including the major class, the screening class and the various class bits.

A sizable number of lines, however, cannot be represented by an abbreviated class because they require more information than a single word can contain. In these cases, the single word associated with the line equipment number contains a pointer used to locate two or four words of data which further expand the class information for that line. These individual class of service expansions are laid out in the same format as the abbreviated class expansions to assure that the resultant output is independent of the translation method. The presence of the pointer is indicated by an all zero abbreviated class.

Two party lines are an example of a situation where additional data words are required. Each party has a separate class and billing number which must be listed as a separate entry. Each entry is then treated as though it came from the terminal equipment number translator. Another example of the variation from the normal pattern is the PBX line where the class of service applies to the entire group of PBX lines. In this case, the pointer indicates to which PBX group (PBX number) the line belongs and its member number within the group. As a result, only one specification of class information is needed for the entire PBX group. The grouping is also used by the machine to mark a PBX line busy as soon as it originates to prevent it from being selected for a call to the PBX.

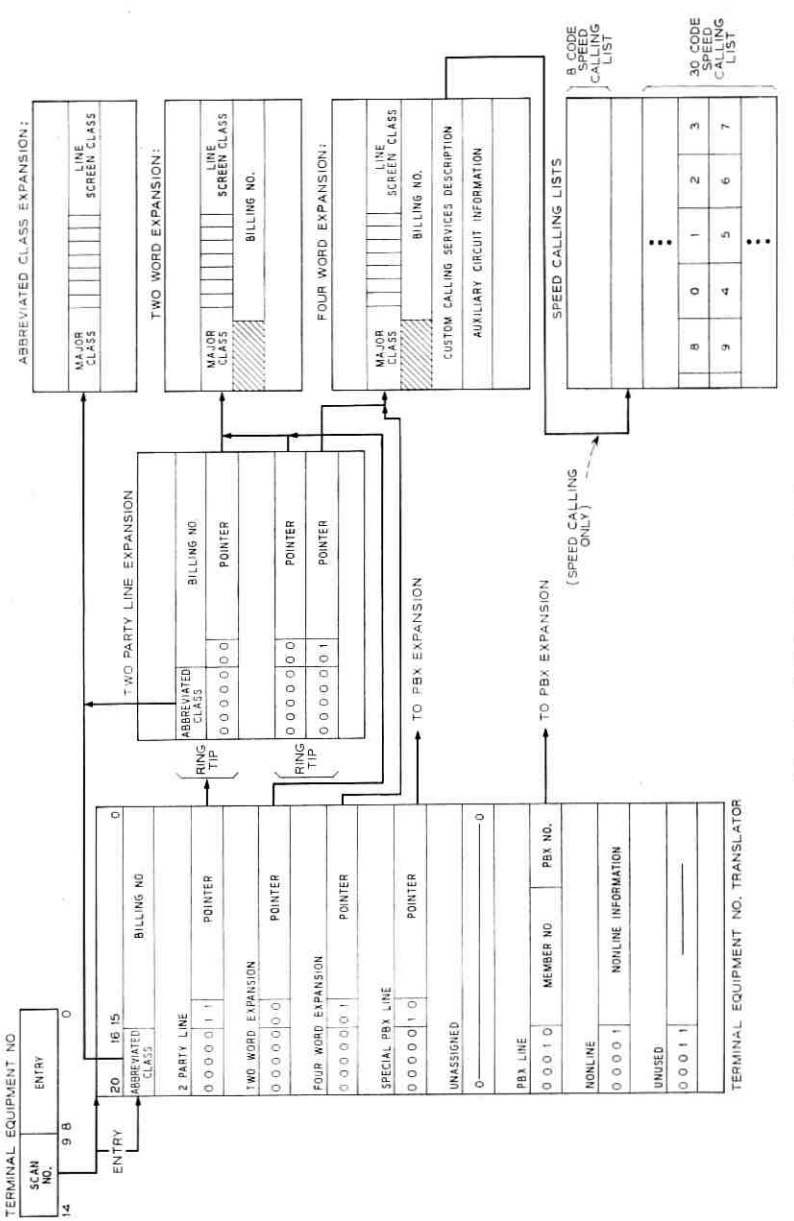



Fig. 6—Originating translation.

5.4.2 *Three-Digit Translation*

The function of three-digit translation is to convert the dialed digits and the originating customer's class of service (obtained from the originating translation) to applicable routing and charging data. The translation must decide if a call is permissible, and how to complete and charge it. The translation is complicated by the varying charges on coin, toll, and message register operations, and the discrimination required for extended area service and wide area telephone service (WATS).

The three-digit translation could be implemented using a program store table for each three-digit code with an entry in every table for each class of service. Such an arrangement, however, would require several hundred thousand program words which is clearly uneconomical. The objective of this translator is to provide the same amount of information using 32 screening tables each with 32 entries, amounting to a total of 1,024 program store words. Several stages of compression of the input data are necessary to permit the use of such small screening facilities.

The three digits dialed by a customer are converted into a ten-bit binary number. As indicated by Fig. 7 the high nine bits select one of 400 words in the code point translator and the low bit selects one of two entries within the word. The resulting eight-bit number defines an entry in the code group expansion table. Many three-digit codes dialed by the customer can be treated identically and, therefore, they use the same expansion.

There are several types of code group expansion entries. The first is used when it is desired to determine the amount of traffic for a specific office code. This preroute peg count entry contains the location of the counter and a pointer to another code group expansion entry which is used for routing the call. The second is a conflict entry which provides two pointers to other code group expansions, one is used if the three digits are determined to be an area code; the other is used if the three digits are determined to be an office code. This facility will eventually be required when the same three digits are used for both office and area codes.

A third type of entry in the code group expansion points to a foreign area translator which provides for further translation of the fourth, fifth, and sixth digits after a customer dials an area code of a nearby number plan area. This foreign area translator facility permits direct routing to specific office codes in the foreign numbering plan area. As

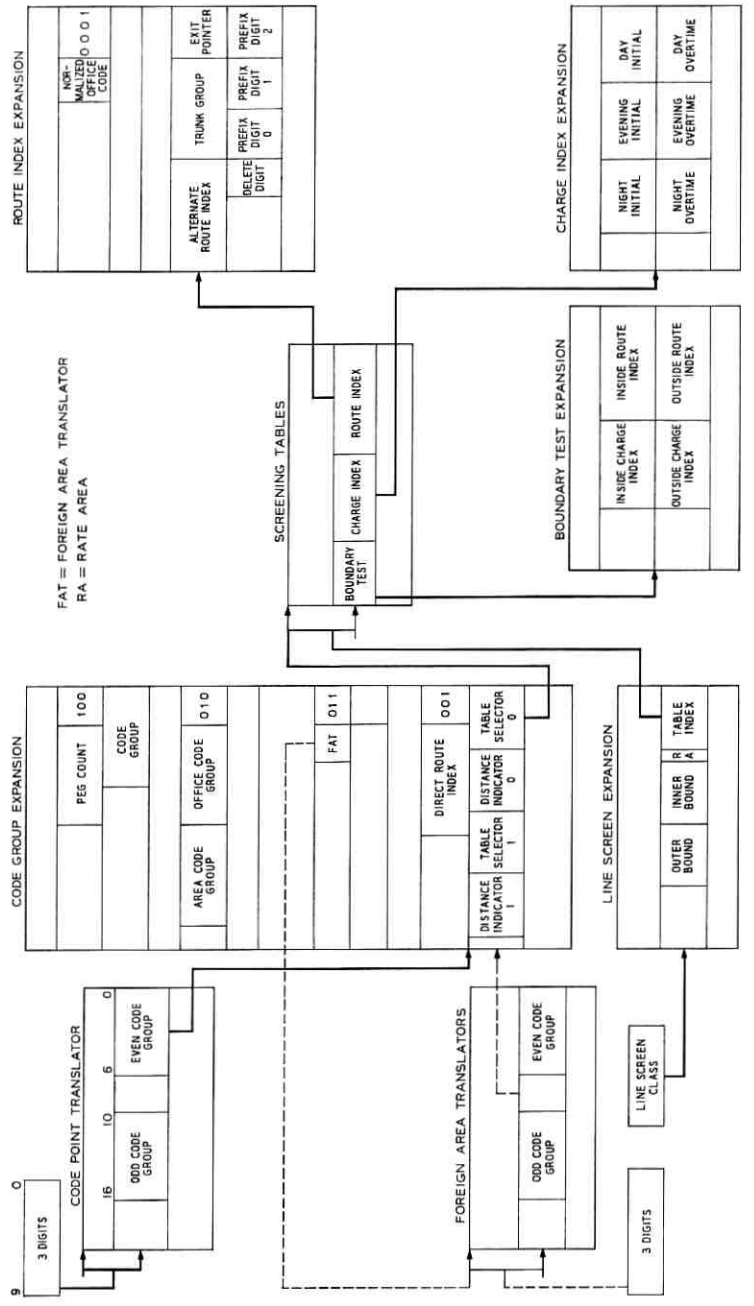



Fig. 7 — Three digit translation tables.

indicated, the foreign area translators have exactly the same format and output as the code point translators and are indexed with the fourth, fifth, and sixth digits dialed by the customer.

The final and most prevalent type of entry is one which indicates the screening table that should be used for this call. If the charge treatment for different three-digit codes is the same but they are routed over different trunk groups, a different direct route index is provided in each of the code group expansion entries, but they point to the same screening table. If the screening process results in a route index equal to zero, then the direct route index is used.

Since it is fairly common for a central office to serve two different rate areas, two screening table selectors have been provided in the code group expansion. The selector to be used in a particular call is determined by the rate area bit in the line screening class expansion for the originating line. Notice that two direct route indexes are not required because regardless of the rate area involved, the direct route to the office having a particular three-digit code should be the same. Also provided for each rate area in this code group expansion is a distance indicator.

The line screen expansion has an entry for each line screen class, and each entry represents a different call treatment. The line screen class is obtained from the originating translation and it is independent of the calling lines major class. Only one type of entry is used in the line screen expansion. It contains an index to the screening table which was previously selected by the code group expansion. It also contains the rate area bit indicating the rate area of the line along with an inner and outer boundary specifier.

The screening table is selected by the code group expansion and the entry is selected by the table index from the line screening expansion. There is only one type of entry in these tables, and each contains a route index and charge index which are to be used for routing the call and providing charging information. If the route index found is zero then the direct route index specified by the code group expansion is used. Also contained in this entry is a pointer to the boundary test expansion to be used for this call.

The route index expansion provides one type of entry which specifies the trunk group that should be used for completing the call, and it provides an alternate route index if the trunk group specified is all busy. In addition it indicates through an exit pointer how many digits are to be expected along with any modification necessary in the dialed information, such as prefixing and deleting digits. A second

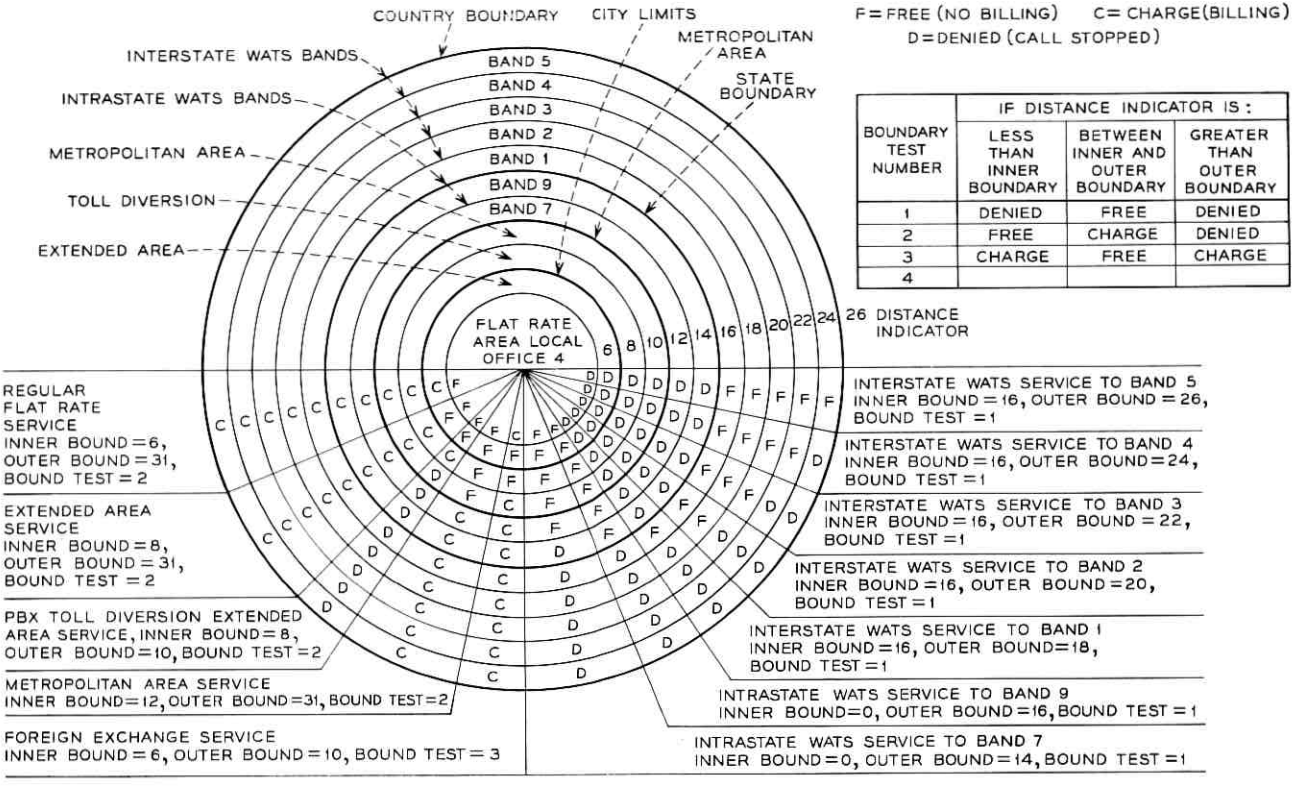
type of entry is provided for intraoffice calls. It contains a normalized office code for later use by the directory number translator, and the three digits of the actual office code.

The charge index expansion is used to determine the appropriate rate to be used in charging for the call. Entries in this table include those for coin and message register calls. Figure 7 shows a typical entry.

A boundary test expansion has been included in this three-digit translator to provide additional screening capability without using a large number of additional program store words. This feature takes advantage of the fact that special charge treatments such as extended area service, and WATS do not require different routing patterns but only different charging techniques and that the details of the charging are determined by the automatic message accounting center during the processing of automatic message accounting tapes and not at the time the call is placed. It also takes advantage of the fact that there is a good correlation between the treatment of various three-digit codes and their distance from the central office, and that a particular customer class of service need have only two boundaries where the method of charging for a call changes.

The boundary test results are obtained by making an arithmetic comparison between the distance indicator obtained from the code group expansion and the inner and outer boundaries obtained from the line screen expansion (Fig. 7). If the distance indicator falls between the inner and outer boundaries, the route and charge index as specified in the screening table are used. Otherwise, the route and charge index as provided by the indicated entry in the boundary test expansion are used. The entry in the boundary test expansion is determined by the boundary test pointer in the screening table. Within that entry, the inside route and charge indexes are used if the distance indicator is less than the inner boundary while the outside route and charge indexes are used if the distance indicator is greater than the outer boundary. If either the route or charge index resulting from the boundary test expansion is zero, then the route or charge index specified in the screening table is used.

An example of the compression of information gained from use of the boundary test is shown in Fig. 8. The upper portion of the circles defines an assignment of distance indicators to 11 code groups. The lower portion of the figure defines 12 line screening classes and assigns inner and outer bounds and boundary test indexes to each of them.




F=FREE (NO BILLING) C=CHARGE(BILLING)
D=DENIED (CALL STOPPED)

| BOUNDARY TEST NUMBER | IF DISTANCE INDICATOR IS: | | |
|---|---|---|---|
| | LESS THAN INNER BOUNDARY | BETWEEN INNER AND OUTER BOUNDARY | GREATER THAN OUTER BOUNDARY |
| 1 | DENIED | FREE | DENIED |
| 2 | FREE | CHARGE | DENIED |
| 3 | CHARGE | FREE | CHARGE |
| 4 | | | |

Fig. 8 — Boundary test diagram.

If no other screening classes were provided for the central office illustrated in the example, then only one screening table would be required and only three entries within that table would be needed. Without the boundary test facility, this example would require 11 screening tables with 12 entries each or a total of 132 words.

### 5.4.3 *Directory Number Translation*

Translation of directory numbers is the process of associating either a terminal number and terminating class or an error treatment indicator with each directory number which can be received from a line or trunk. For most lines in an office a single 21-bit word is sufficient, but lines subscribing to the call waiting, call forwarding, or the series completion feature require more than one word to contain all terminating line data. For them, a two-word expansion is needed and a pointer is used in the terminating translator to locate the expansion. Formats for entries in the terminating translator are shown in Fig. 9.

To afford reasonable breakage and ease of directory number assignment, the directory number translator has been subdivided into 100 word blocks called terminating translators. A terminating translator is indexed with the binary equivalent of the tens and units digits of the directory number to obtain the desired entry. For each office

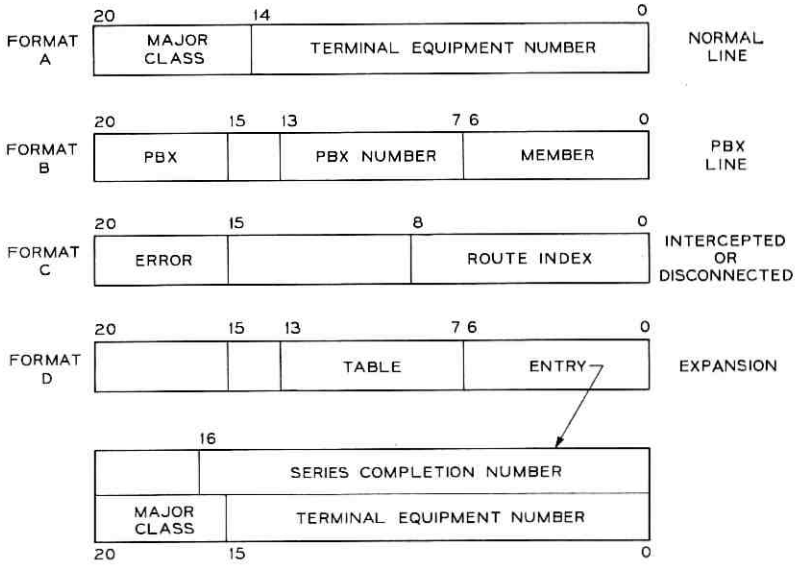



Fig. 9 — Terminating translator entry formats.

code used, 100 of these tables may be provided. Access to the terminating translators is gained through an index called a normalized office code which is obtained from the three digit translation for intraoffice calls or from trunk translation information for incoming calls. For each normalized office code there is an entry in the master table index which contains the address of a number group translator. The number group translator contains 100 entries, each of which is a pointer to a terminating translator. Indexing the number group translator with the binary equivalent of the thousands and hundreds digits of the directory number locates the entry which contains the location of the corresponding terminating translator. The entire directory number translator is depicted in Fig. 10.

#### 5.4.4 *Trunk and Service Circuit Translations*

Trunk and service circuit translation subroutines are used to obtain peripheral circuit assignment information. Outputs from the translator consist of the terminal memory record address, terminal equipment number, trunk group and member number, peripheral decoder enable address, and scan point numbers.

Although both scan point numbers and trunk group numbers are used as inputs to the trunk and service circuit translations, only the use of scan points is described here.

The manner in which storage is accessed in translating a universal trunk frame scan point number is indicated in Fig. 11. The depicted subtranslator contains two words for each of the 128 circuits on a universal trunk frame bay and two additional words used to derive the terminal memory record address and peripheral decoder enable for these circuits. The uniformity of universal circuit scan point and peripheral decoder assignments allows simplicity of the translator and compression of the information stored.

There is no pattern to the manner in which the scan points and peripheral decoders are assigned to miscellaneous trunks and service circuits. Two additional words describing these assignments must, therefore, be stored for each miscellaneous circuit to allow these extra degrees of freedom. A subtranslator accessed for micellaneous scan points is shown in Fig. 12.

## VI. DETAILS OF AN INTRAOFFICE CALL

A better understanding of how the call processing functions are controlled to provide telephone service may be obtained from a detailed description of an intraoffice call.

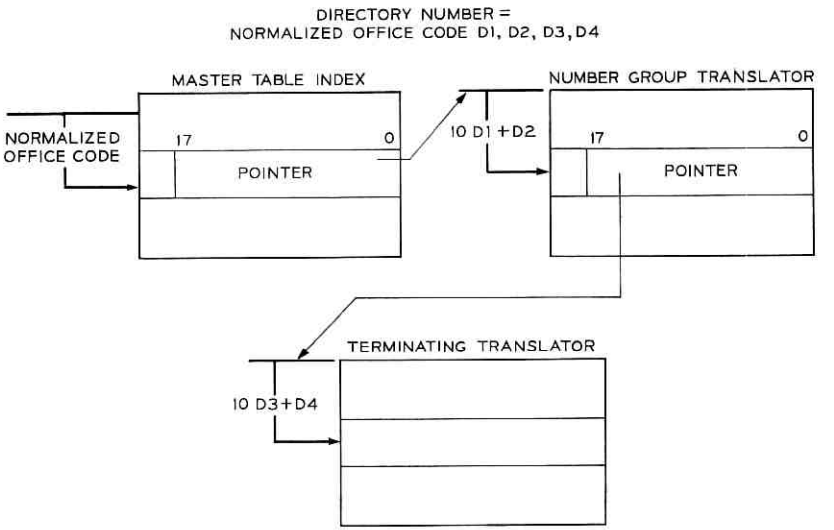



Fig. 10 — Directory number translator.

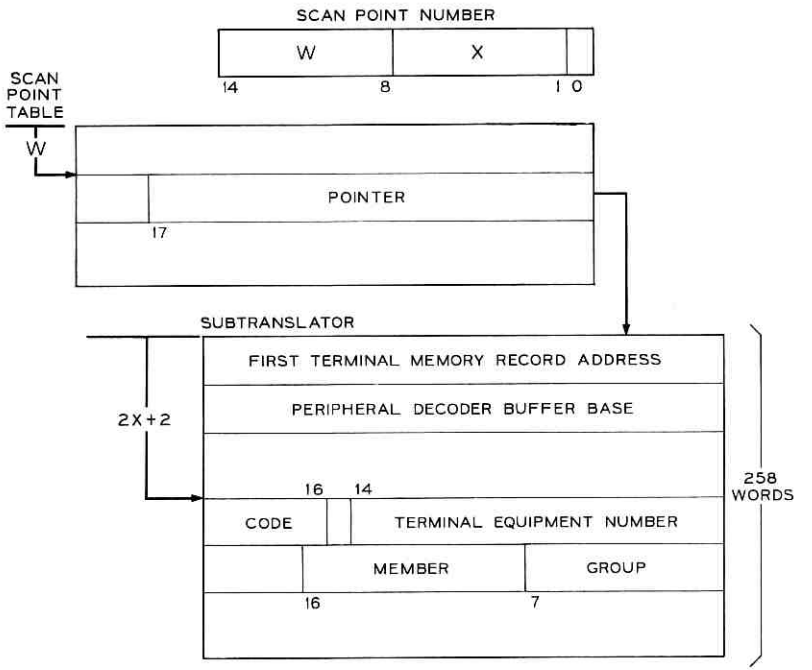



Fig. 11 — Universal trunk and junctor translator.

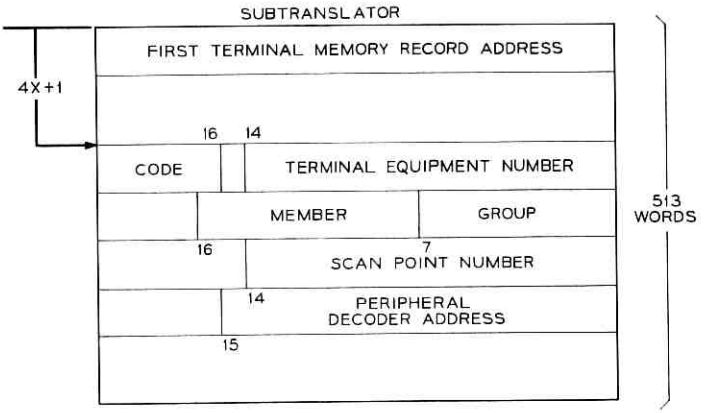



Fig. 12 — Miscellaneous scan point subtranslator for 128 circuits.

When a subscriber goes off hook the ferrod associated with his line is saturated. This fact is detected by the line scanner, and the interrupt program performs hit timing before placing the line terminal equipment number in the originating hopper (see Section 5.1.1). Later the supervision distributor detects the nonzero entry during its processing of the hopper, and searches for an idle transient call record. When one is found, the terminal number of the line is written into the second word of the transient call record, and the line origination progress mark is recorded in the first word. The line terminal number is then cleared from the origination hopper. If an idle transient call record is not available, processing of the origination hopper is terminated and the line number is allowed to remain in the hopper to await an idle transient call record.

At the conclusion of the supervision distributor program's execution, the call processing scan is begun. During this scan the transient call record containing the originating line is encountered and control is passed by means of the progress mark to the line origination program. This program's first action is to obtain a translation of the line number to determine the originating treatment for the line. An idle originating register must be found to receive digit information from the customer, and its address is recorded in the fifth word of the transient call record (Fig. 2). The line originating class data is then examined to determine whether a dial pulse or combined receiver is needed to receive digits from the line. The circuit group number of the desired receiver is supplied to a subroutine which selects a digit receiver of the type required by the customer.

The selection subroutine is responsible for locating an idle digit receiver and reserving a network path for use in connecting the line to the receiver. When an idle receiver is found, it is made busy in call store and the circuit group number is used by the service circuit translation programs to obtain its terminal equipment number. The network map contained in call store is then examined, and an idle path from the calling line to the receiver is selected. The links corresponding to the idle path are made busy in the network map and information describing the path is placed into word 1 of the receiver terminal memory record (see Fig. 4). At this time the terminal memory record is also placed into the transient state, and the address of the transient call record is written into the first word of the terminal memory record. A ten-bit circuit number is placed into the sixth word of the transient call record to identify the selected receiver. In the event that the selection program is unable to locate a path to the selected receiver, the receiver status bit is made idle and a second receiver is selected. A path selection is then attempted for the second receiver. If no path is found on the second try the failure is communicated to the line origination program.

All equipment identities needed to connect dial tone to the calling line are now stored in the transient call record. To achieve the connection a subroutine is called which selects a peripheral order buffer and loads it with all circuit and path information required by the peripheral order buffer execution program. The loading program translates the digit receiver ten-bit number contained in the sixth word of the transient call record to obtain the peripheral assignments for the circuit. No further actions can be taken for the call until the connection is made and so the peripheral order buffer loading program stores a new progress mark in the transient call record to await completion of the connection.

This progress mark causes a program to check during each base level transient call record scan if dial tone has been given. It also provides for queueing in the event that all peripheral order buffers were found busy by the peripheral order buffer loading program.

The final action of the dial tone peripheral order buffer is to record the scan points of the digit receiver in the originating register so that digit collection may be initiated by the wired logic digit receiving circuits. This action must be done after dial tone is returned in order to insure that relay operating transients are not mistaken for dial pulses. Because as much as 100 milliseconds may elapse before the base level program becomes aware of the successful connection,

the scan point numbers must be recorded in the originating register during the interrupt so that dial pulses will not be missed. The successful connection is reported to the transient call record by the peripheral order buffer loading program, and the progress mark is changed to address the digit reception program.

The digit reception program is written in a general way so that it may be used to collect digits for all types of calls and still allow good flexibility in the interpretation of the digits. Once each 500 milliseconds this program examines the ready bit (Fig. 5) contained in the originating register addressed by the fifth word of the transient call record. A zero in this bit indicates that no new digits have been received and tests are then made to determine if partial dial time-out has occurred. If the ready bit is set to 1, a new digit has been received. A comparison is then made between the incoming digit count contained in the fourth word of the originating register and a digit number stored in the fifth word of the transient call record. If the incoming digit count is found to be less than the digit number, the ready bit is zeroed and no further action is taken. If the incoming digit count is equal to or greater than the digit number, control is transferred to the address contained in the eighth word of the transient call record. A user of the digit reception program may then initialize the digit number to a number of digits which need to be collected before translation is necessary and designate the location where control is expected when that number of digits have been received. An additional feature of this program allows the user to specify by means of a bit in the transient call record whether the digit reception program should dispose of dialing time-out indications in a normal manner or whether the user program should be informed of a time out.

During the course of a call, the first incoming digit is examined as soon as it has been dialed for the purpose of providing the 1 and 0 prefixing features. Of course an initial 0 can also signal an operator call. For lines subscribing to custom calling features, two digits must also be examined so that requests for the service may be detected. The digit reception program normally must wait for three digits, however, before the type of call being placed can be discovered. A subroutine is used to translate the first three dialed digits, and the returns from this subroutine inform the digit reception program of the type of call being placed. An intraoffice call indication from the translation subroutine causes the digit reception program to await the completion of seven digits before the called number is fully determined. For this

case the translation subroutine supplies a charge index and an indicator called a normalized office code which designates that one of a maximum of six office codes has been dialed. Both the charge index and normalized office code are recorded in the seventh word of the transient call record.

Reception of the seventh digit causes the digit reception program to call the directory number translation program to decode the full number (see Section 5.4.3). The normalized office code is supplied to the directory number translation program and is used to index the proper number group translator. The translation returns a terminal equipment number and a called party terminating major class to the digit reception program. The terminal equipment number is used to locate the scanner last look bit to determine if the called party is busy. This bit has a zero value when the line is idle and assuming this to be the case, the last look bit is set 1 to busy the line.

A talking path is necessary for the final line-to-line connection, and a path hunt subroutine is used to search the call store network map for this path. In order to guarantee a low probability of blocking on all connections, the identity of the A link used for the line side of the digit receiver connection is passed to the path hunt program. In searching for a talking path the subroutine attempts to reuse this A link. It should be observed that if this tactic is not employed, at most three A links are available for use in the talking connection. This condition, of course, lowers the probability that a path can be successfully found.

The path hunt subroutine selects a circuit junctor for use in connecting the two lines because the lines must be supervised at the junctor during the talking interval. When a path has been found, the terminal memory record of the circuit junctor is made transient and path information is recorded in the second word of the terminal memory record. The identity of the A link reserved for the called line side of the talking connection is returned so that it can be shared with the path used for ringing the called party.

Selection of a ringer includes an attempt to give immediate ringing to the called party. By consulting information recorded in the call store by the ringing and tone frame maintenance program, the ringer group is selected which will ring the station soonest. When a path is found for use in ringing the called line, it is recorded in the ringer terminal memory record and this record is placed in the transient state.

Calls which require timing during the talking interval (message

register, local coin) must be identified in a call store list called a stable call timing list during that interval. Before proceeding to ring the called line an idle two-word entry called a stable timing entry is located and reserved for the call. Failure to locate the stable timing entry when it is required causes the call to be intercepted.

Automatic message accounting calls are recorded by means of a nine-track magnetic tape machine using a triple entry format. If the call requires automatic message accounting billing the initial entry is formed and placed in a call store buffer for later transmittal to the tape machine.

Ringing is supplied to the called party by execution of a peripheral order buffer loading program which selects a peripheral order buffer and loads information in it necessary for connecting the ringer. Successful execution of this routine causes another peripheral order buffer to be loaded to disconnect the digit receiver and connect the calling line to the circuit junctor selected for the final talking path. Audible ringing is then returned from the junctor circuit to the calling party. The base level call program awaits completion of each of these peripheral order buffer actions by executing a call to the peripheral order buffer loading program during each transient call record scan as described previously. Completion of the second peripheral order buffer action causes the originating register, the digit receiver, its associated terminal memory record and path memory used for the dialing connection to be idled in memory. The ringing progress mark is then stored in the first word of the transient call record to await further customer action.

In order to conserve system real time, the ringing program executes only one instruction during each transient call record scan while awaiting a response to ringing. This instruction simply skips past the ringing transient call record and causes the following transient call record to be processed. As described in Section 5.1.2, the supervisory entry point into the ringing program is taken when the supervision distributor reports supervision to the transient call record. When this program branch is reached, bits stored in the transient call record by the supervision distributor are decoded to determine the source of the change in supervision (calling or called party) and the new supervisory state.

Indication of an off-hook condition of the line supervisory ferrod in the ringer circuit causes the program to call the peripheral order buffer loading program to select and load a peripheral order buffer with information necessary to disconnect the called party from the

ringer and connect him to the circuit junctor. The peripheral order buffer execution program also changes the state of the junctor circuit relays so that audible ringing is removed from the calling party and the two parties are connected in a talking state.

Successful execution of the peripheral order buffer actions allows the ringer circuit, and the ringing path to be idled in memory. A new progress mark is then stored in the first word of the transient call record causing a one-second period to be timed as a check for valid answer. During the one-second period on-hook reports from the called party are ignored so that on-hooks generated by switch hook bounce will not be erroneously interpreted. A disconnect received from the calling line, of course, terminates the call.

After the called party has remained off-hook for one second, charge guard timing is begun by storing a new progress mark in the first word of the transient call record. This timing period extends for two seconds and is provided for insuring that the connection is truly established before a charge is made for the call. An on-hook report received from the called party during the "charge guard" interval initiates timing as if the called party had terminated on an established call.

Upon completion of the two-second charge guard timing interval, charging is begun. For automatic message accounting calls, an answer entry is formed and placed in the automatic message accounting buffer. Coin or message register charge timing is initiated by addressing the previously selected stable timing entry and changing its format so that the stable timing entry can start timing of the initial charge interval. The transient call record is then cleared and the circuit junctor terminal memory record is used to record the identities of the calling and called lines. Bit 15 of the first word of the terminal memory record is also set equal to 0 to indicate that the call is stable. No further action is required for the call until one of the lines takes further action or further charging action becomes necessary because of time out of the stable timing entry.

To terminate the call one of the parties must hang up. The on-hook condition causes the junctor circuit ferrod used to supervise the on-hook line to indicate the on-hook transition when it is next interrogated by the trunk and junctor scanning program. After 50 milliseconds of hit protection timing, this program reports the on-hook scan point number to an on-hook timing list which is routinely examined by the supervision distributor. The on-hook report is allowed to remain in the on-hook timing list for 250 milliseconds before further action is taken, in order to guarantee the on-hook condition,

During a subsequent execution of the supervision distributor, examination of the on-hook timing list indicates that time out has occurred. The terminal memory record corresponding to the circuit junctor is then accessed via a translation of the scan point number and discovered to be stable. This terminal memory record state causes the supervision distributor to select an idle transient call record for handling tear down of the call. The calling and called line numbers are recorded in the second and third words of the transient call record and a disconnect progress mark is written into the first word. Additional information is placed in the transient call record to allow determination of whether the calling or called party terminated. The circuit junctor terminal memory record is then made transient by the supervision distributor.

During the call processing scan the disconnect program is accessed via the progress mark and the on-hook party is determined. Should the calling line be the first to disconnect, he is disconnected from the circuit junctor by means of a peripheral order buffer action and the called party is retained in the transient call record awaiting his on-hook signal. Receipt of the calling party disconnect also causes the following actions:

(*i*) An automatic message accounting disconnect entry is recorded in the automatic message accounting buffer if required.

(*ii*) Coin or message register timing is terminated and the associated stable timing entry is cleared if employed.

(*iii*) A coin collect is performed on the calling line if applicable.

(*iv*) The calling line last look (status) bit is set to 0.

The ensuing hang up by the called party results in another peripheral order buffer action to disconnect him from the circuit junctor. The circuit junctor, its associated terminal memory record, the path, and the line last look bit are then idled in memory and the transient call record is cleared. When the called party hangs up prior to calling party disconnect, actions similar to the above are taken, but an 11 second period is timed prior to release of the connection. A calling party disconnect during the time out period causes the call to be terminated without further timing.

## VII. TESTING

Call processing programs, automatic progression tests and manual progression tests are all used to detect trunk and service circuit abnormalities. The diagnostic programs take advantage of the many

functions that are common to those required by the normal call processing routines. In this way large program and call store economies are realized. For example a transient call record is used to control the test sequence. This permits the call store area used by the diagnostic program to be released when the test is completed. It also allows the test programs to use call processing routines for selecting circuits, sending peripheral orders, and outpulsing.

### 7.1 *Trunk and Service Circuit Testing*

Trunk and service circuit maintenance facilities in the No. 2 ESS have three main objectives: (*i*) to automatically detect faulty circuits and remove them from service as soon as possible after a trouble condition arises, (*ii*) to provide a teletypewriter message indicating the action taken and a trouble number to pinpoint the trouble, and (*iii*) to provide facilities to aid in manual testing and repair.

The main phases of trunk and service circuit maintenance (detection, diagnostic, and repair) are shown in Fig. 13. Trouble detection may result from automatic or manual detection tests or troubles may occur during call processing.

In the normal processing of calls, checks are made at key points for abnormalities that may exist. Timing is performed at places where supervision, such as a start pulsing signal, is expected from another office. Continuity and foreign potential checks are made whenever a new network path is set up. Checks are built into every scanner and peripheral decoder order. Whenever one of these checks detects a trouble condition that could have been caused by a trunk or service circuit, diagnostic programs are called in to test the suspected circuits.

Diagnostic tests for trunks and service circuits are designed to insure that the circuit can carry out its normal functions. The circuit being tested is connected through the network to other service or test circuits and a sequence of "test steps" is performed which produces pass or fail results. Data obtained from the test is used to produce a trouble number which consists of the "test step" and the results of the test, such as ferrod responses. This approach permits one number to serve both as the dictionary number for the trouble locating manual and as raw data.

Outgoing trunks are tested by placing a test call to a "test line" in the far end office. Tones and supervisory signals are transmitted by the test line as a part of the test sequence and to indicate the results of tests made on the trunk at the far end. A tone detector is connected

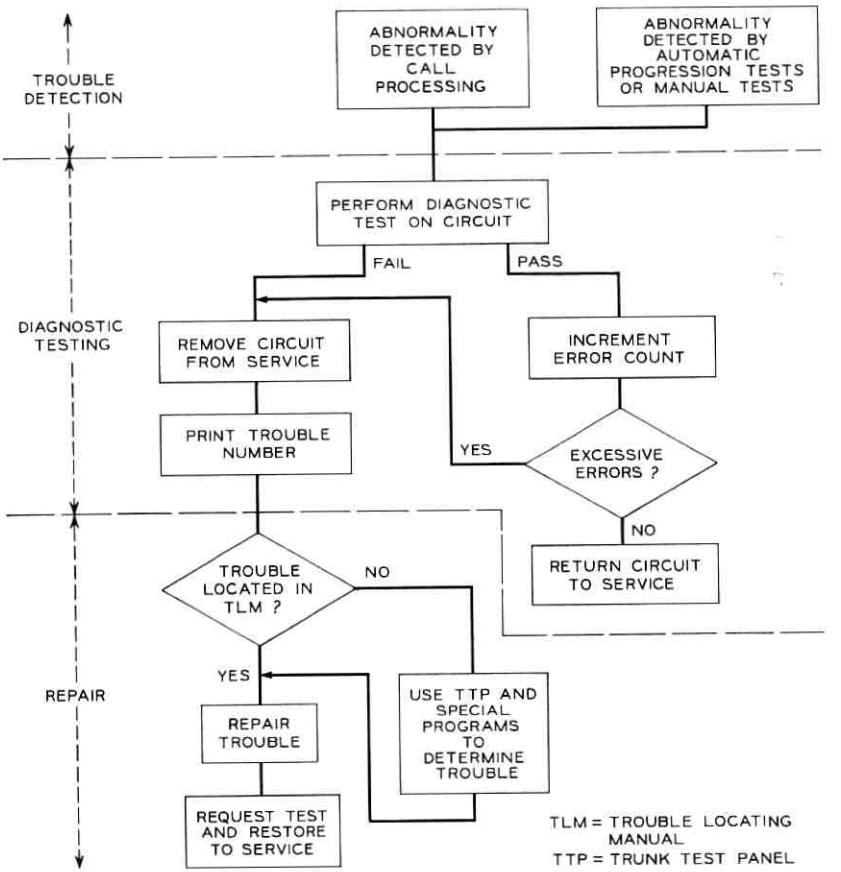



Fig. 13 — Trunk and service circuit maintenance plan.

to the near end of the trunk to detect these tones and busy or reorder conditions. The test program monitors the tone detector and the trunk supervisory ferrods for these responses and compares the results to "all tests pass" data that are stored in memory.

The philosophy of testing service circuits differs somewhat from that of trunk testing. Since trunk tests involve other central offices, testing techniques are dictated by the test facilities available in those offices. However, service circuits are contained wholly within one office and complete freedom exists in the design of such tests.

Two basic approaches are used in service circuit testing. One method is to connect the circuit to a specially designed test circuit through the

switching network. Marginal signals are generated by the test circuit to insure that the service circuit can function properly under any condition that may be encountered in normal service. Both circuits are sequenced through a series of states, and their ferrods are scanned in each test state. The results are then matched against "all tests pass" data stored in memory. This approach is used to test customer dial pulse receiver circuits where the test circuit simulates marginal customer dial pulses that might be transmitted over "worst case" customer lines.

The other method used in testing service circuits is to use the complement of the circuit being tested to generate the input signals. Multifrequency transmitters and receivers are tested in this manner by selecting a transmitter (receiver) at random to test a particular receiver (transmitter). A test circuit is connected into the network path between the two circuits to introduce transmission loss and distortion. Over a period of time all combinations of transmitters and receivers are tested as pairs and nonworking or marginal circuits are detected.

If the tests described above were carried out in response to errors detected by call processing programs, then a test failure results in the circuit being removed from service; an "all tests pass" is treated in the manner described in Section 7.2. If the tests were carried out in response to automatic progression testing or manual testing then a failure on the first pass through the tests is treated as a trouble detection and the test is repeated. The results of the second try are treated in the same manner as a failure from a call processing test request.

Safeguards are included to prevent the automatic removal of too many circuits in a group. Two successive failures in an automatic progression test causes a test of the test circuit. The number of circuits that can be automatically removed from service is limited to a small percentage of that group.

A manual test of a trunk or service circuit is initiated via the maintenance teletypewriter or the trunk test panel. Teletypewriter requests may specify a particular circuit, group, or all circuits to be tested. A diagnostic test is always performed on a circuit when the craftsman attempts to restore it to service.

### 7.2 *Error Analysis*

When a trouble condition is encountered by call processing programs and the subsequent test of the circuit yields an "all tests pass,"

an error is recorded for that circuit. Errors may be caused by overload conditions in the far end office, dirty relay contacts, transient noise, or by a marginal condition that is not detected by the diagnostic program. The general scheme of error analysis is to detect circuits that are more "error prone" than other circuits of the same type. Error counts are kept on a limited number of circuits at a time. The error count is compared with an "excessive error count" which depends upon the total errors accumulated by all the other circuits in the same "error analysis group." An excessive number of errors results in the circuit being removed from service and a teletypewriter message is printed to inform the craftsman. If a circuit does not accumulate enough errors to be removed from service, it is replaced in the error analysis list by another circuit that is suspected of high error rate.

### 7.3 *Growth Testing*

Errors in the installation of new circuits or in the translation data associated with that circuit not only result in the circuit not working but may also affect other circuits. This seriously degrades service and may cause symptoms of troubles in other parts of the system such as scanners or the central pulse distributor.

Special programs have been provided which check the translation data associated with a circuit and perform an "installation test" of the circuit. The regular diagnostic program is used for testing service circuits; trunks are tested with the outside loop specially terminated to permit testing before assignment to a regular trunk group and to isolate circuit troubles from those that might be caused by outside facilities.

### 7.4 *Line Testing*

In No. 2 ESS, line testing facilities are concentrated at the local test cabinet and the local test desk. The cabinet is located at the central office while the desk may be at some remote centralized location. Continuity, leakage, foreign potential, and ferrod tests may be made from either the desk or cabinet. Permanent signal testing will be, as far as possible, under the control of the desk. The trunk test panel may also be used to test lines when necessary, although it is intended to be used primarily for trunk and service circuit testing.

Automatic line insulation tests, which check for excessive leakage and foreign potential, are conducted periodically on all idle lines in the office. Test failures are reported to a teletypewriter at the local test desk. This test may also be requested by the craftsman via the

local maintenance teletypewriter or the local test desk teletypewriters.

A line test facility that can be used by an installer from the customer's premises or by a craftsman at the local test desk is the station ringer test circuit. It is used to test for station ringer ground and to test *Touch-Tone*® telephone dials. Tones from the test circuit indicate proper operation or failure.

## VIII. CONCLUSION

No. 2 ESS provides the switching functions needed by a modern central office by means of an electronic data processor under the control of a set of call processing programs. This paper enumerates the switching function provided and discusses the organization and structure of the call processing programs used to minimize the program size. It describes in detail the primary processing functions and gives an example of an intraoffice call to illustrate the use of these functions to provide telephone service and to provide circuit testing capability.

The large variety of telephone services offered by No. 2 ESS is provided by a program of modest size. The system is expected to be economically competitive in the small and medium size central office field, and the program structure has made a major contribution to this competitive position.

## REFERENCES

1. Beuscher, H. J., Fessler, G. E., Huffman, D. W., Kennedy, P. J., and Nussbaum, E., "Administration and Maintenance Plan," B.S.T.J., this issue, pp. 2765–2815.
2. Browne, T. E., Quinn, T. M., Toy, W. N., and Yates, J. E., "Control Unit System," B.S.T.J., this issue, pp. 2619–2668.
3. Digrindakis, J., Freimanis, L., Hofmann, H. R., and Taylor, R. G., "Peripheral System," B.S.T.J., this issue, pp. 2669–2712.
4. Lonnquist, C. W., Manganello, J. C., Skinner, R. S., Skubiak, M. T., and Wadsworth, D. J., "Apparatus and Equipment," B.S.T.J., this issue, pp. 2817–2863.
5. Spencer, A. E., and Vigilante, F. S., "System Organization and Objectives," B.S.T.J., this issue, pp. 2607–2618.
6. Barton, M. E., Haller, N. M., and Ricker, G. W., "Service Programs," B.S.T.J., this issue, pp. 2865–2894.

# Administration and Maintenance Plan

By HUGH J. BEUSCHER, GEORGE E. FESSLER, D. WAYNE HUFFMAN, PETER J. KENNEDY and ERIC NUSSBAUM

(Manuscript received March 12, 1969)

*The No. 2 Electronic Switching System has been planned with dependability, maintainability, and operational ease representing a major portion of the total system design concept. The major characteristics that fall under these headings are redundancy, trouble detection, trouble recovery, diagnosis, repair, man-machine interface, data protection, and operational and administrative procedures for growth and change. All these factors play a large role in both the hardware and software portions of the system design. This article reviews the highlights in each of these areas and describes how they interact in the overall system across both the control complex and peripheral system areas.*

## I. OBJECTIVES

A successful local switching system must be dependable as seen from the customer's viewpoint and economical to maintain and operate as seen from the telephone company's view. This performance must be achieved under the stress of a variety of central office environmental constraints—physical (temperature, humidity), electrical (noise, commercial power interruption), traffic (overload), office growth (addition of lines, trunks, networks, stores, and so on), and human errors. The design of a switching system such as No. 2 ESS must then take into account all of these factors in achieving the proper balance between equipment first costs and annual operation and maintenance costs, while also meeting customer service standards.

### 1.1 *Dependability*

The customer measures a system's dependability in service continuity and accuracy of call handling (dialing, billing, and so on). The design objectives are, therefore, continuous high quality service

24 hours a day for a 40-year life. Specific reliability objectives are that total system down-time (time during which customer service is noticeably degraded) should not exceed two hours during its 40-year life and that, on the average, not more than 0.01 percent incorrectly handled calls should result from system troubles and errors. Furthermore, from a customer's service viewpoint, under trouble conditions it is preferred that service degradation peaks be avoided—a few calls handled incorrectly occasionally or a few very brief outages per year causes less customer inconvenience than less frequent but longer duration service difficulties and interruptions.

In order to achieve this degree of dependability, attention must be paid both to the operation and administration of the system when it is functioning normally, and to its maintainability in the face of trouble.

### 1.2 *Operation and Administration*

While processing calls, the switching machine must also monitor its own performance in terms of traffic measurements of calls handled in service, and of calls aborted, because of shortages of switching or trunk facilities. In addition, plant measurements of aborted calls and associated trouble conditions are necessary to provide an indication of the machine's health. This data, in hard copy, direct-reading simple formats, provides the basis of much of the craftsmen's actions in the day-to-day operation of the system.

In a stored program switching system, reassignments and additions of trunks, for example, may involve hardware changes, translation memory changes, and updating of administrative forms. Other classes of change items include customer line assignments (frequent but relatively simple) and the addition of major growth items such as networks and stores (less frequent but more complex). Experience has shown the importance of carefully human engineering these procedures since human intervention is often the source of system trouble rather than conventional component failures.

### 1.3 *Maintainability*

The actual process of trouble detection, system recovery, diagnosis, repair, and service verification is broadly classed in the maintainability category. In order to achieve the required dependability, the design must stress:

(*i*) Use of long-life components and adequate circuit margins.

(*ii*) A redundancy plan sufficient to keep the system operational in the presence of component failures and human intervention for administration and growth.

(*iii*) Trouble detection mechanisms (hardware and software) for all service effecting parts of the system, either automatic, routine, or, at worst, manual.

(*iv*) Rapid recovery of the call processing function and protection of the calls in progress in the face of either hardware, software, or human intervention (procedural) difficulties.

(*v*) Diagnosis and isolation of troubles (both hard faults and transients) either by automatic (programmed) machine analysis of data or by manual means.

(*vi*) Repair and verification procedures which allow for requested repetitive testing of trouble items, simple mechanical procedures, and automatic recheck on service restoral.

(*vii*) In general, a high level of human engineering in all displays, controls, teletypewriter communications, and operational procedures in which the craftsman interacts with the machine—hopefully with a view to keeping these items as similar as possible between different switching systems.

### 1.4 *Centralized Maintenance and Administration Facilities*

The inherent reliability of solid state components and good circuit design result in a significantly lower trouble rate than that experienced by comparable electromechanical systems; however, the complexity, high speed of operation, loss of the ability to audibly and visually observe apparatus operation has tended to increase the skills required by plant craftsmen to restore system operation and locate those troubles which the machine cannot handle automatically. In addition, the low trouble rate experienced does not give craftsmen sufficient practice to maintain their proficiency.

To overcome these possible problems and to take advantage of the great potential for annual cost savings, facilities and procedures should be provided to allow remote administration and maintenance control. Such facilities allow a small group of well-trained craftsmen at a centralized point to provide administrative and maintenance coverage to a number of ESS type offices almost equal to their being physically in each office. The amount of trouble hunting experience, at this point, should assure continued craftsman proficiency. Many actual repair operations, running of cross connections, and other tasks

requiring physical contact with the equipment may be taken care of by semiskilled craftsmen, either dispatched from a central point or, if the work load is adequate, by assigning such craftsmen to the office on a schedule.

The centralization of such facilities can also allow specialization of functions in convenient locations. For example, the plant service center would have direct access to a teletypewriter for making customer line change translations to any of several offices. And the traffic department could have access to data from several offices at one central location.

In summary then, the objectives of dependability imply the need for a high level of maintainability and operational ease, and the demands for economy can be substantially aided by centralizing these functions.

## II. GENERAL ADMINISTRATION AND MAINTENANCE PLAN

The dependability, adminstration, and maintainability objectives, when applied to stored program switching systems, define the need, in computer terms, for an on-line real-time, high availability machine. To achieve this economically requires careful initial systems planning in basic redundancy configurations, in man-machine interface capabilities, and in hardware-software tradeoffs. Though difficult to allocate precisely, it is estimated that approximately two-thirds of the total program is dedicated to a system of maintenance and administrative programs that are used to administer system redundancy, control detection, and diagnostic routines, make performance measurements, and provide for communication with the craftsman. It is the need to keep the switching system operational during periods of growth and change of customer services (major hardware, program and translation additions and changes), the need to keep calls being processed during switches to standby equipment (preservation of transient data), and the requirement of providing simultaneous on-line communication with a number of craftsmen, that adds extensively to the program structure and makes maintenance more than simply a matter of diagnostics.

### 2.1 *Basic Redundancy Plan*

As described in Ref. 1, the entire control unit (program control, program store, call store, input-output and peripheral bus system) is considered as a single entity which is duplicated. Two control units,

plus the maintenance center, comprise a control complex. Since the component count and failure rates are sufficiently low, no reconfiguration within a control unit is necessary, thus simplifying both the hardware connections and controlling programs. The maintenance center is unduplicated except for the existence of multiple teletypewriter channels (eight maximum) which serve a variety of different functions. In the peripheral equipment, network and scanner controllers and supplementary central pulse distributors are duplicated since failure here can effect large numbers of lines. Network fabric, trunks, junctors, and service circuits are traffic-engineered items and thus contain inherent redundancy.

In order to provide rapid recovery from troubles and effective continuity of service, the processors normally are run in synchronism. This enables the off-line processor to keep its registers and call store contents continuously up to date and thus constantly available to take over on-line functions. This synchronous mode of operation involves providing all inputs (both normal scanning and peripheral maintenance responses) to both processors, while deriving outputs only from the on-line machine. Since network and scanner controllers do not have the long-term memory functions associated with processors, their duplicate mates are operated either in a traffic load shared state or in an idle stand-by condition.

### 2.2 *Man-Machine Interface*

The major communication between the switching machines and the craftsman is by teletypewriter. In addition, audible alarms and visual displays are used to alert the craftsman to trouble conditions which are subsequently more fully reported on a teletypewriter channel. Manual controls are available on the maintenance center for performing special tests on out-of-service equipment and for taking restart action when the system has lost its "sanity" to the point where it can no longer interpret teletypewriter input commands.

#### 2.2.1 *Teletypewriter Facilities*

The system program is organized and the hardware is arranged for a maximum of eight teletypewriter channels. Each of these channels can be programmed to produce only certain classes of messages and to accept only a limited class of requests. Any one of these channels can operate a remote teletypewriter by the use of a data set. A typical installation might include four teletypewriters:

(*i*) Channel 1—local maintenance

(*ii*) Channel 2—remote maintenance

These channels report all system maintenance activity (troubles detected, diagnosis results, plant registers, and the like) and accept *all* system input messages (maintenance and other).

(*iii*) Channel 3—service order

This channel operates a remote teletypewriter at the plant service center which is used to input changes in customer line information (class of service, features, directory and billing numbers, and so on).

(*iv*) Channel 4—traffic

This channel provides traffic data according to a defined schedule (items such as trunk group usage, call rate, and dial tone delay). Specific data can be requested and the schedule can be changed by input messages on this channel.

All channels have built-in maintenance checks on each message and each one is arranged to provide automatic message transfer to an alternate channel in case of failure. Alternate backup channel definitions can also be changed easily.

#### 2.2.2 *Display and Manual Controls*

In addition to teletypewriter information, the maintenance center has quick reference visual displays of items such as system alarm status, processor on- off-line status, and processor, trunk, and peripheral unit trouble indications. It also has a dynamic display of the characteristic program loop (program address register).

Manual controls available to the craftsman can be divided into three basic categories:

(*i*) The most frequently used in normal operation are associated with a trunk test panel facility. Trunk testing arrangements provide for switched access to trunks and for measuring dc and ac signaling, transmission, and noise parameters. To use them, a craftsman activates special panel keys, a *Touch-Tone*® telephone key set, and a teletypewriter.

(*ii*) Facilities on the maintenance panel provide various tests on both the on-line and the off-line processors. Available off-line facilities include the ability to step through programs or whole routines, to interrogate and load registers and call store locations, and to preset condition (address) traps by use of the comparator. On-line functions include dynamic visual register and store displays, and

preset condition program interrupts which can provide snapshot teletypewriter dumps. In addition, controls are provided to vary margins (threshold) in both off-line call and program stores.

(*iii*) Manual restart controls are provided as a final backup when more normal communications fail. Included are such items as forcing and locking either control unit on line, and initiating memory (call store) clearing operations to restart the system when it loses "sanity."

### 2.2.3 *Documentation*

The use of all these man-machine facilities is built on a hierarchy of documents with which the craftsman must be familiar.

The first is the *Input Message/Output Message Manual* which defines all possible teletype messages which are programmed into the machine and lists all acceptable input requests and the expected system response to them. Figure 1 shows a typical input message entry. This one is used to update the system calendar and each variable field is fully defined in the manual.

```
                                                        UB SY:DAT

1.  INPUT MESSAGE FORMAT

        UB SY:DAT:mon day year b!

2.  EXPLANATION OF MESSAGE

        Used to enter the current date into the system

        UB =Utility Base level request.  The request will be
            performed immediately.
        SY = SYstem
        DAT = DATe

        mon = month of the year (1-12 decimal)
        day = day of the month (1-31 decimal)
        year = year - decimal, last two digits
        b = day of the week - decimal 0-6. Sunday is 0

3.  SYSTEM RESPONSES

        OK = message was OK. It was accepted and the work requested
             has been accomplished.
             If the message came from paper tape, the tape re
             be turned on.

        NG = the message was not accepted (N
             action or data fields were
             system state may not
             proceedures  or
             the
```

Fig. 1 — Sample *Input Message Manual* description: system calendar update.

When the output message gives specific diagnostic data, this *Manual* points to a *Trouble Locating Manual* (see Fig. 2) which translates the data field of the message (*Trouble No.* column) into a specific set of suspect circuit packs. Brief remarks are incorporated to describe the trouble area functionally. Should this information prove inadequate (that is, replacement of packs does not clear the trouble), subsequent sections of the *Trouble Locating Manual* are arranged to give a detailed functional description of the test and an interpretation of the digits in the trouble number. Repair procedures might then involve reference to the more basic maintenance documents, including program listings and schematic drawings. Experience indicates that better than two-thirds of the troubles should be cleared through use of only the simple trouble number translation.

A series of *Bell System Practices* are provided as basic training documents and give overall system descriptions in addition to detailing all operational and administrative procedures the craftsman must perform. These documents contain extensive cross references.

```
BELL TELEPHONE LABORATORIES, INCORPORATED

  TROUBLE NO.        EQUIP LOC - TYPE              REMARKS
       (CONT)
                     I0-132-30A   A440
                     I0-132-28    A440

                     I0-132-19    A440
                     I0-032-38    A408
                     I0-032-36    A403

  3812               FAILURE OF Z CPD TRANSLATION FROM THE EA

  3812 000001        I0-030-46    A403
                     I0-030-47    A401
                     I0-030-48    A438
                     I0-030-43    A403

  3812 000002        I0-030-46    A403
                     I0-030-47    A401
                     I0-030-48    A438
                     I0-030-43    A403

  3812 000003        I0-030-46    A403

                          (CONTINUED)
ESS NO.2 - ********************************* ISS **     TLM-*****-D513
```

Fig. 2 — Sample *Trouble Locating Manual.*

### 2.3 *Programming Organization*

In order to control all the maintenance, communication, and administration functions described above, the program structure is organized into a hierarchy of tasks performed at base level times and interrupt times. In addition, an initialization, or restart, procedure is provided under certain circumstances, resulting in a break in the continuity of program flow.

#### 2.3.1 *Base Level Programs*

All deferable, or low priority, maintenance tasks are handled at the end of the normal call processing transient call record scan.[2] Items covered here regularly include processing the waiting list of incoming and outgoing teletypewriter messages, including such functions as timing, format translation, and distribution of messages to client programs.

The base level maintenance monitor program determines which additional tasks are to be performed. The normal sequence of these tasks may be modified by any maintenance activity that has taken place since the last transient call register scan. For example, if a check circuit output has automatically switched out and inhibited a processor, this fact is taken into account here and diagnostics on the off-line take preference over lesser periodic routines. Similarly, manual requests may have a higher priority. One or more of the following functions is performed at this time:

(*i*) Execution of any manual (teletypewriter-inserted) requests such as demand tests or make-busy functions.

(*ii*) Updating of off-line call stores after an interval of non-synchronized processor operation.

(*iii*) Diagnostics on the processor and on peripheral units as a result of calls from trouble recovery routines or manual requests.

(*iv*) Short-term periodic routines performed on a schedule, including items such as processor trouble detection programs and initiation of trunk tests.

(*v*) Long-term periodic routines intended to exercise those circuits not used in normal operation, primarily check circuits and on-line/off-line switching facilities.

(*vi*) Miscellaneous functions, including error count tabulations in the call store (such as plant registers), directed scans of various ferrods assigned to maintenance functions, and control of maintenance center displays.

### 2.3.2 *Input-Output Interrupt Level Programs*

In addition to the scanning, digit receiving, and outpulsing functions that call processing handles during the periodic 25-millisecond input-output interrupt routine, maintenance functions requiring close timing are also executed here. Items covered here include:

(*i*) Trunk and service circuit tests requiring precisely timed actions are executed first to avoid stagger resulting from variable execution time in the various parts of the interrupt program.

(*ii*) The network controller's maintenance ferrods are checked for proper operation based on the previous interrupt's actions. Failures result in peripheral order buffer retries and eventual call teardown if no working mode is found.[2] The base level routine is notified of the troubles in order that diagnosis may be initiated later.

(*iii*) All teletypewriters are scanned for new inputs and new characters are outputted to active teletypewriters.

### 2.3.3 *Maintenance Interrupt Level Program*

The maintenance interrupt has the highest priority. It is initiated by processor mismatches as detected by the call store comparison in the maintenance center, by some peripheral unit and input-output errors, and by manual request. All three sources come in at the same priority level and block each other until their respective tasks are complete. These error signal interrupts immediately initiate trouble recovery programs and after the appropriate recovery actions (for example, switch of on-line/off-line control unit configuration or scanner controller) the problem is passed on for further resolution to the lower priority base level programs.

### 2.3.4 *Initialization Restart*

If the processors switch their on- off-line configuration while the off-line is out of synchronism, or if they go into a multiple switching mode from which they cannot recover, the program is restarted from a fixed location to provide an orderly return to the beginning of the call processing monitor cycle. The initial source of the trouble may be either hardware or software difficulties. A count of the number of restarts incurred during a given time is used to progressively clear out the call store until the system recovers its "sanity." This strategy involves clearing or releasing various transient and maintenance only locations of call memory while preserving most stable talking path

records. An initialization restart of this type may also be manually initiated, including a complete call clearing capability when necessary.

## III. ADMINISTRATIVE FUNCTIONS

A particular telephone office is defined to the call processing programs by a series of parameters and translation tables in program store that describe that office's network traffic characteristics, trunking facilities, routing and charging constraints and all individual subscriber definitions. The initial office traffic and trunking engineering is performed as a result of the operating company's analysis of needs and results in ordering the proper equipment frames for initial installation. All program store translation and parameter contents for initial service are processed by means of an Office Data Assembler program in a general purpose regional computation center.

As time passes, the typical office evolves and grows, and individual subscriber definitions change as people move and new services are offered. In order to respond to these changes, a series of administrative (recent change) programs are resident in the No. 2 ESS machine to permit a virtually continuous memory updating capability. Subscriber changes are generally originated by way of the plant service center, while network and trunking modifications are based largely on the traffic measurements performed by the processor on its daily calling rate pattern, and on operating company projections. In the case of major equipment growth additions, translations are changed by means of a new Office Data Assembler run.

In addition, another class of administrative functions known as plant measurements are maintained for each office. These involve both service measurements to reflect actual effects on service as seen by the customer (for example, total customer receiver time-outs) and performance measurements to reflect the basic health of the machine in terms of such items as failure and error counts of various pieces of equipment. These measurements are useful in directing attention to areas where additional maintenance effort appears justified.

### 3.1 *Traffic Measurements*

Traffic measurements are made throughout all phases of the call processing programs and are recorded in call stores. Data are put out through a dedicated teletype channel on assigned quarterly, hourly, daily, and weekly schedules, or on demand. Various combinations of the three basic types of measurements (peg counts, usage,

and overflow) are performed in such areas as networks, junctors, service circuits, trunks, and office calls. Their usefulness can perhaps best be described by some examples.

In the network, usage counts are maintained for each concentrator for use in load balancing and line assignment. In the junctor area, usage counts are kept on wire and circuit junctor groups for load balancing between networks and for intraoffice-interoffice call rate measurements. Trunk measurements are made for each group with various combinations of peg count, usage, and overflow in outgoing, incoming, and two-way categories. Usage counts are also made on subscriber items such as the various custom calling services.

When these data indicate the need for relatively minor reconfigurations without major hardware additions, translations are changed by local recent change procedures in the No. 2 ESS machine. In other cases, more elaborate processing is required at a regional computation center.

### 3.2 *Recent Change Procedures*

The types of items that fall in the recent change category include service orders (subscriber additions and changes), trunk additions, service observing, addition of new routes, and changes in office code treatment. The service orders are usually remotely entered from a plant service center; the other items are performed locally by the craftsman from the maintenance center teletypewriter. Again, an example will perhaps best describe the process.

The original office record input forms (Fig. 3) indicate that directory number 736-0056 is vacant. A new line is to be added. The information to be inserted includes this chosen vacant code together with its assigned terminal equipment number, associated billing number, desired features (dial transfer and add-on conference) and line class code (1FR single party, flat rate, residence). This information is keyed in on the service order teletypewriter channel as shown in Fig. 4 in the standard universal service order code input language. This information is processed by the No. 2 ESS, checked for validity, converted into its binary equivalent, and stored in a special call store recent change buffer.

The call store buffer, with a capacity of 512 updated program store words per module of permanent magnet twistor translation storage (16,384 words), is searched by the call programs for changes each time its associated permanent magnet twistor translations are ac-

DIRECTORY NUMBER RECORD CONVERSION
NO. 2 ESS

ESS 2100
TEL CO 002 (3 4 5)

OFFICE CODE 736 (17 18 19)

BASE & CONTROL NO. 77716 (6 7 8 9 10 11)

HUNDRED 00 (20 21)

ESS UNIT 736

FORM CODE 14 (15 16)

| DIRECTORY NUMBER (22 23) | TERM EQUIPMENT NUMBER: NET (25 26) | CONC GROUP (27) | CONC (28) | SWITCH (29) | LEVEL (30) | RATE AREA (31) | PARTY NO. (32) | CLASS INFORMATION: LINE CLASS CODE (33 35) | EQUIPMENT: LCA (36) | TES (37) | TTC (38) | (39) | (40) | FEATURES: (41) | (42) | AD2 (43) | ADI (44) | ADO (45) | CWT (46) | DTR (47) | SUPPL INFO: PAGE (48 51) | ITEM (52 53) | REMARKS (54 80) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 50 | 05 | 2 | 5 | 1 | 2 | 1 | | 1FR | ✓ | | | | | | | | | ✓ | | | | | |
| 51 | | | | | | | | | | | | | | | | | | | | | | | |
| 52 | 02 | 6 | 3 | 5 | 1 | 0 | | 1MR | | | ✓ | | | | | | ✓ | | | | | | |
| 53 | | | | | | | | | | | | | | | | | | | | | | | |
| 54 | 01 | 1 | 2 | 1 | 3 | 0 | | 1MB | | | ✓ | | | | | ✓ | | | | | ✓ | | | |
| 55 | | | | | | | | | | | | | | | | | | | | | | | |
| 56 | | | | | | | | | | | | | | | | | | | | | | | |
| 57 | 03 | 1 | 7 | 2 | 3 | 1 | 2 | 2MR | | | | | | | | ✓ | | | ✓ | | 0003 | 05 | |
| 58 | | | | | | | | | | | | | | | | | | | | | | | |
| 59 | | | | | | | | | | | | | | | | | | | | | | | |
| 60 | | | | | | | | | | | | | | | | | | | | | | | |
| 61 | | | | | | | | | | | | | | | | | | | | | | | |
| 62 | | | | | | | | | | | | | | | | | | | | | | | |

Fig. 3 — Sample office record input form.

```
A RC 50/      <--ADMINISTRATION RECENT CHANGE SERVICE ORDER.
TYP NEW/      <--TYPE NEW LINE.
IN 736 0056/  <--TELEPHONE NUMBER.
TEN 022200/   <--TERMINAL EQUIP NUMBER (NW#,GROUP,CONC.,SW.,LEVEL)
BIN 736 0050/ <--BILLING TO NUMBER.
FEA DTR ADD/  <--FEATURES: ADD-DIAL TRANSFER
FEA ADØ ADD/  <--FEATURES: ADD-ADD ON CONFERENCE.
LSØ 1FR1 ØK   <--LINE CLASS CODE - SINGLE PARTY, FLAT RATE, RESIDENCE
```

Fig. 4 — Sample recent change teletypewriter service order input message.

cessed. As changes and additions accumulate in the buffer, an output indication is given on the teletypewriter that the maximum capacity is being approached and that action should be taken to update the permanent magnet twistor magnet cards. This is accomplished by means of the single card writer contained in the maintenance center frame.[1]

The contents of the recent change call store buffer are analyzed by program and translated into the affected set of permanent magnet twistor cards (128 words per card) which are identified in a teletypewriter message upon request. These selected cards are individually inserted in the single card writer and the entire card is magnetized by a program that copies the old on-line translation plane from permanent magnet twistor with all appropriate call store buffer recent change entries incorporated. When all affected cards have been magnetized, they are inserted in the off-line program store, and automatically verified against on-line program store plus call store change buffers. If successful, the on-line/off-line processors are switched, the procedure is repeated, and the call store buffers are cleared to allow for accumulation of the next group of changes.

Depending on office size, rate of change, and local office manning practices, this translation updating procedure may be performed perhaps every two weeks. In larger offices, more automatic translation updating procedures would be provided if the change rate warranted it. Throughout the process, office record forms must be kept up to date in order that future changes do not create conflicts.

### 3.3 *Major Translation Change Procedures*

Certain translation changes require a simultaneous change of large blocks of data, plus extensive validity tests and error checking on the input data. In some cases, even with reasonable amounts of data, the translator structure is sufficiently complex to make the relatively simple recent change program inadequate. The changes which fall in these categories are situations which require major office equipment growth (network frames, trunk frames, storage additions, and so on),

major revisions in routing, screening, and charging translations, and general reorganization of existing translator origins as additional storage is added.

In all of these cases, the problem can usually be anticipated well ahead of time and so the rapid response characteristic of the recent change procedure is not essential. As a result, use is again made of the Office Data Assembler program as shown in Fig. 5. Used here in its update mode, the office data assembler program accepts new punched input forms, validates and error checks them, and incorporates the data into the existing office translators. To ensure that the actual No. 2 ESS office translator being updated is consistent with the administrative forms, the Office Data Assembler operates on an actual dump of the program store contents, and is capable of providing a new set of adminstration records.

## IV. CONTROL COMPLEX MAINTENANCE

The control complex, as stated earlier, consists of two control units and a maintenance center. Only two configurations of control complex equipment are possible: either one of the control units on-line

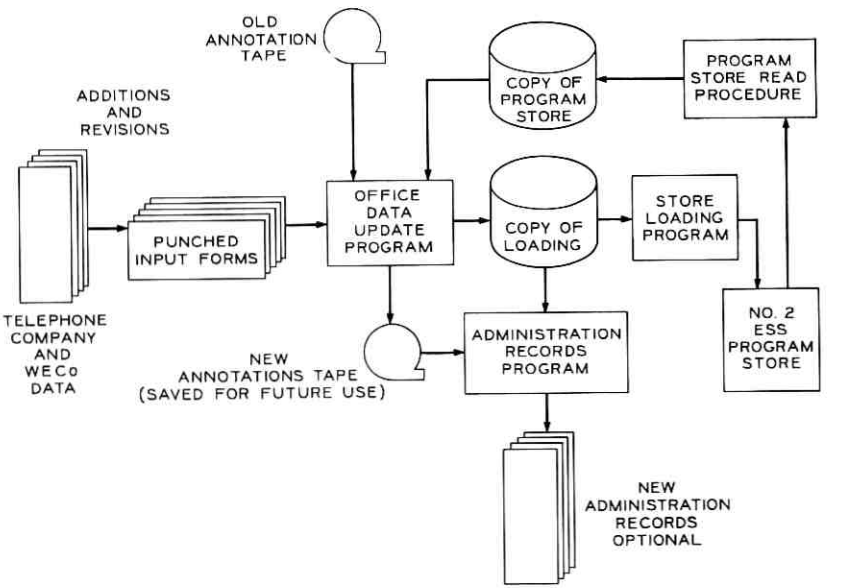



Fig. 5 — Initial loading and growth processing of office data.

(controlling peripheral equipment) and the other off-line. Normally, the two control units are running in command synchronism with only the on-line control unit actually performing a controlling function.

Problems occur in control complex operation when either a solid or marginal circuit fault occurs. Both maintenance circuits and programs are used to detect a trouble condition, to recover a working system, and finally to either determine that a transient error occurred or to diagnose what circuit fault exists. Figure 6 summarizes how troubles are detected and what happens after a trouble is found.

Maintenance circuits and programs can handle a wide variety of different faults. For example, programs and circuits used in fault diagnosis are designed to handle faults commonly encountered in the high-speed transistor resistor logic used in No. 2 ESS. These faults include: (*i*) open and shorted semiconductors, (*ii*) open resistors, and (*iii*) open connector contacts on pluggable circuit packages.

Programs and circuits which recover a working system after a fault occurs can handle not only these faults but also faults which cause

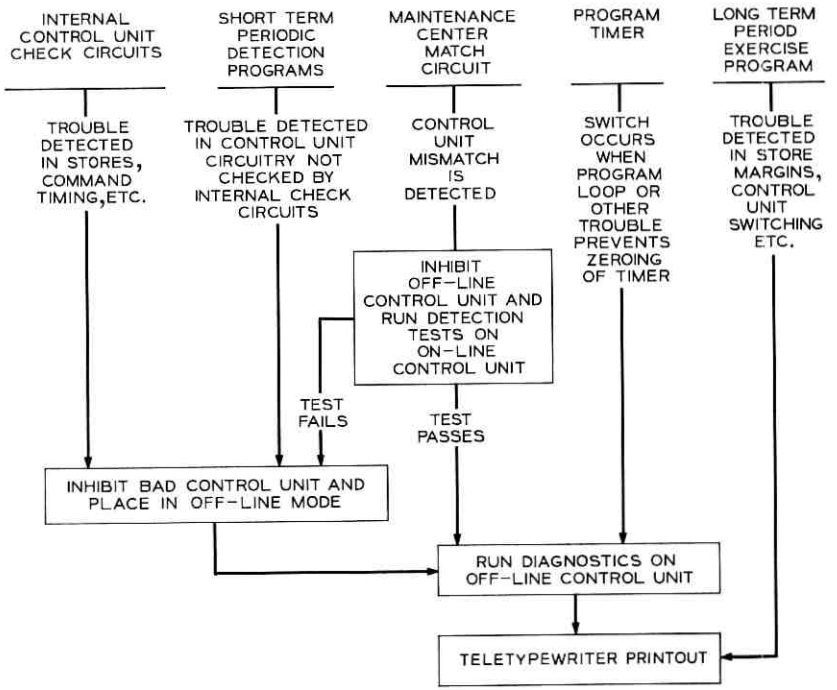



Fig. 6 — Control complex maintenance.

marginal circuit operation or faults such as wire opens or wire crosses which should not occur frequently.

### 4.1 *Trouble Detection Methods*

A combination of check circuits and program tests are used to find circuit troubles as shown in Fig. 6.

#### 4.1.1 *Check Circuits*

Typical internal control unit check circuits detect problems such as store access faults and automatically attempt to recover a working system by switching control units if a fault is detected in the on-line control unit. Check circuits are used when program tests cannot detect a fault fast enough or reliably enough. For example, the call store access check circuit detects faults such as shorted access diodes which cause marginal store operation. This class of faults cannot be solidly tested by any type of program check.

A maintenance center check circuit compares the call store input registers in the two control units when call store operations are performed in synchronism. A fault or transient error in almost any part of either control unit quickly results in a call store input register mismatch since almost all tasks performed in both the program control and input-output involve call store writing. A check circuit mismatch signal interrupts the program currently being run and causes a trouble recovery program to be called in which attempts to find the faulty control unit and place it in the off-line mode. This trouble recovery program is described in Section 4.2.

Each program control contains a program timer circuit which is designed to backup other detection methods. Normally, an on-line control unit program zeroes a counter in both program timers at least once every 300 milliseconds. If, however, a trouble condition exists such as a program loop which prevents a timer from being zeroed (within 320 milliseconds for the on-line and 640 milliseconds for the off-line), the timer will time-out and automatically produce a switch. The new on-line control unit is automatically forced to run an initialization restart program which attempts to establish a working system.

#### 4.1.2 *Program Detection*

Short-term periodic program tests detect the same troubles found by the mismatch trouble recovery program since exactly the same program tests are run. These tests are continually run interleaved with

call processing, and detect faults within approximately five seconds, compared with microseconds in the case of the comparator circuit. These tests provide trouble detection even when the control units are not being matched and provide a backup to the comparator check circuit. In fact, if very rapid fault detection were not required, it would not be necessary to have a comparator circuit.

Detection program testing of the input-output presents a special problem since this unit is normally operating independently of the program control. The program control has to stop the input-output, save, test, and restore input-output registers, and restart the input-output each time an input-output detection test is run in order to prevent interference with normal input-output operation.

Long-term periodic exercise programs perform tests on circuitry not normally checked by other detection means. For example, both the off-line program and call stores are placed in marginal modes and tested for correct operation. This is accomplished by applying high and low values of threshold voltage to store readout amplifiers at the same time words are read out and checked for correctness. This check attempts to force store problems to show up before they can affect actual system operation. In addition to store margin tests, the complete diagnostic test sequence as well as a test of control unit switching is performed to force troubles to show up in circuits not exercised in normal system operation.

An additional periodic check on correct system operation is performed by the base level maintenance monitor which checks the system state once each program scan by looking at certain key flip-flops. If an abnormal state is found, a trouble has occurred and diagnostics are called in. For example, diagnostics are called in if the off-line control unit is found inhibited and not running programs when it is supposed to be running in synchronism with the on-line control unit.

### 4.2 *Trouble Recovery*

After a trouble is detected, automatic circuits and, in some cases, trouble recovery programs are used to obtain a working system. The physical action taken to restore a working system is very simple as a result of the simple split redundancy. If a fault is found in the on-line control unit, control units are switched and the new off-line control unit is inhibited from running programs. If, on the other hand, a fault is found in the off-line control unit, the only action

taken is to inhibit the off-line control unit. These actions are automatically initiated when faults are detected by control unit check circuits or by programs. After trouble recovery action, the base level maintenance monitor will automatically call in diagnostics when the off-line control unit is found inhibited.

### 4.2.1 *Mismatch Detection Programs*

A special mismatch trouble recovery program is run in the on-line control unit after a mismatch to determine if the on-line control unit contains a fault. This program first inhibits the off-line control unit and then calls in detection tests. These detection tests are identical to those run during short-term periodic detection. However, all the tests are run at once instead of being interleaved with call processing in order to minimize test time. Tests are run in a sequence which attempts to test as much circuitry as possible as quickly as possible (all tests are run within 100 milliseconds). As a result, functions tested first are those which exercise the most circuitry. Actually, a large amount of circuitry can be assumed to be good at the beginning of the mismatch detection test, since many failures are detected by internal control unit check circuits rather than by the maintenance center match circuit.

If a solid fault in the on-line control unit is detected by a mismatch detection program, a control unit switch is automatically generated which inhibits the new off-line control unit and uninhibits the new on-line control causing an initialization restart program to be run in it. The new on-line control unit will be slightly out of date since no new inputs were recorded in its call store while mismatch detection programs were being run; however, it will not contain erroneously processed information. Shortly after the switch, the base level maintenance monitor will call in diagnostics when it finds the off-line control unit inhibited.

If the on-line control unit successfully passes all detection tests, diagnostic tests of the off-line control unit are called in. These tests attempt to determine if an off-line control unit fault caused the mismatch. Diagnostic test failure results in formation of a teletype printout which attempts to pinpoint the location of the fault. Successful completion of all diagnostic tests indicates a transient error condition caused the mismatch. If this is the case, control units are placed back in synchronism and one is added to a call store word containing the number of all test pass conditions.

#### 4.2.2 *Peripheral Unit Trouble Recovery Programs*

Certain faults in the input-output unit do not result in control units mismatching but instead cause peripheral units to be accessed improperly. Detection of these faults by certain internal input-output unit check circuits cause peripheral unit trouble recovery programs to be called in. These programs, described in detail in Section V, attempt to recover a working system by retrying peripheral orders plus switching peripheral and control unit equipment if necessary.

### 4.3 *Diagnostics*

Diagnostic programs are automatically called in by the base level maintenance monitor after trouble detection and recovery has been completed or can be manually requested via the teletypewriter. The objective of diagnostic programs is to produce a teletypewriter printout which isolates a fault to as small an area as possible. The following paragraphs describe circuitry and programs which are used to achieve this.

#### 4.3.1 *Maintenance Circuitry Provided for Diagnostic Testing*

Special control unit circuitry is provided to allow diagnostic tests to resolve the location of faults to a relatively few number of circuit packs. External maintenance commands allow the on-line control unit to control and monitor actions performed in the off-line unit. As shown in Fig. 7, the contents of on-line control unit registers can be gated to off-line registers or vice versa. Also, control functions such as starting and stopping the off-line control unit can be performed. This can be accomplished by resetting (to start) or setting (to stop) the off-line inhibit flip-flop. Figure 7 shows examples of two different external commands. One command executed in control unit 0 causes a register in that unit to be gated to a register in control unit 1 via the program gating busses. The other command starts control unit 1 by zeroing the inhibit flip-flop.

In a typical diagnostic test, the on-line control unit uses external commands to test off-line circuitry as follows:

(*i*) Stop the off-line control unit, initialize command timing and gate a program command directly into the program store output register.

(*ii*) Execute the command present in the off-line program store output register.

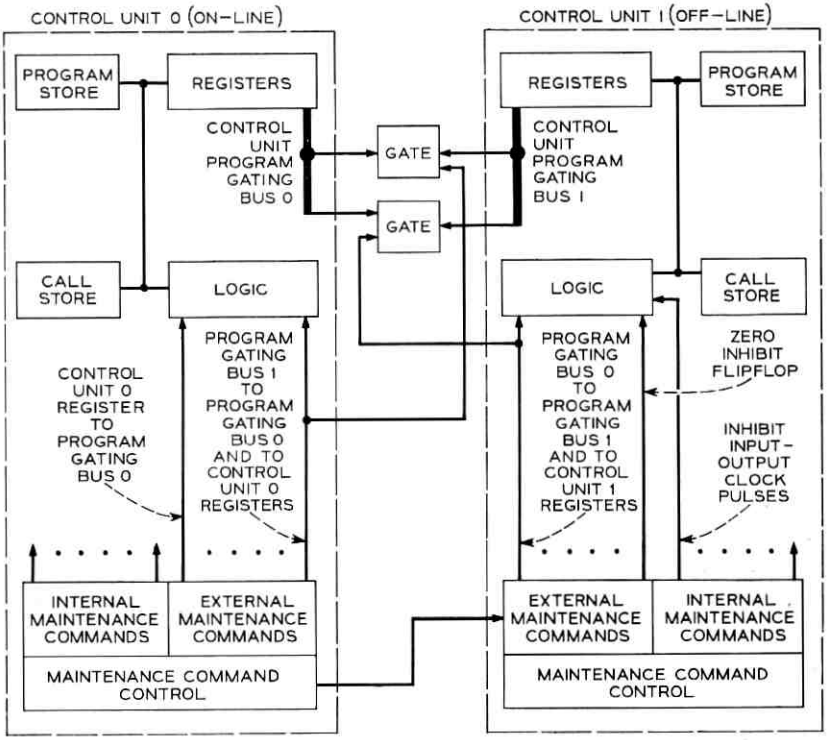



Fig. 7 — Typical diagnostic program commands.

(*iii*) Look directly at an off-line register to see if the command was executed correctly.

This testing method allows off-line control unit circuitry to be tested by the on-line control unit without having to rely on the correct operation of a large amount of off-line circuitry, including the off-line program store.

Certain internal maintenance commands executed by the off-line control unit are also used to aid diagnosis. These commands can either be executed under direct control of the on-line control unit as outlined above or as part of a program sequence controlled by off-line internal command logic (the command is read out of the off-line program store). In this second case, external commands are only used to establish an initial off-line program starting address, to start the program sequence running, and to look at test results after off-line testing is completed.

Internal maintenance commands are used to test areas such as the input-output unit which are not accessable to normal internal commands. For example, a typical input-output diagnostic test uses an internal maintenance command to set a flip-flop which prevents clock pulses from enabling input-output gates. This stops all input-output operations and allows the program control to execute additional internal maintenance commands to test specific input-output functions.

Commands which gate to and from maintenance center registers are also used by the on-line control unit to both control and monitor actions performed in the off-line unit. For example, outputs of off-line check circuits can be observed by looking at the maintenance center error register.

#### 4.3.2 *Diagnostic Programs*

It is important to order diagnostic tests so that a command or circuit used to test another command or circuit has itself been previously tested. If this rule is followed, the only circuitry under suspicion if a given test fails is the circuitry currently being tested and only a single printout indicating what test failed and how it failed is required to provide diagnostic information. Good test access from the on-line to off-line control unit allows tests to be ordered in this manner in most cases.

Figure 8 is a simplified flowchart of the diagnostic sequence. First, on-line to off-line access is tested both via the maintenance center and external commands. Success of these test blocks insures that sufficient on-line to off-line access is available to test off-line circuitry in detail. The next test blocks use this access to test various parts of the off-line control unit including command logic, the program store, and the call store. Near the end of the diagnostic sequence, enough off-line circuitry has been tested so that diagnostic programs can be run entirely in the off-line control unit. Tests such as input-output diagnostics are run in this manner under control of the on-line control unit. In many cases, these same tests are also used in periodic and mismatch detection.

The entire diagnostic sequence is always called whenever control complex diagnostics are called in automatically. This insures that faults are caught by the proper diagnostic test and that a meaningful printout is produced. No attempt is made to use the results of trouble recovery programs in order to shorten the time required for diagnostic testing since diagnostic test time (about 30 seconds) is an insignificant part of the total time required to repair a fault.

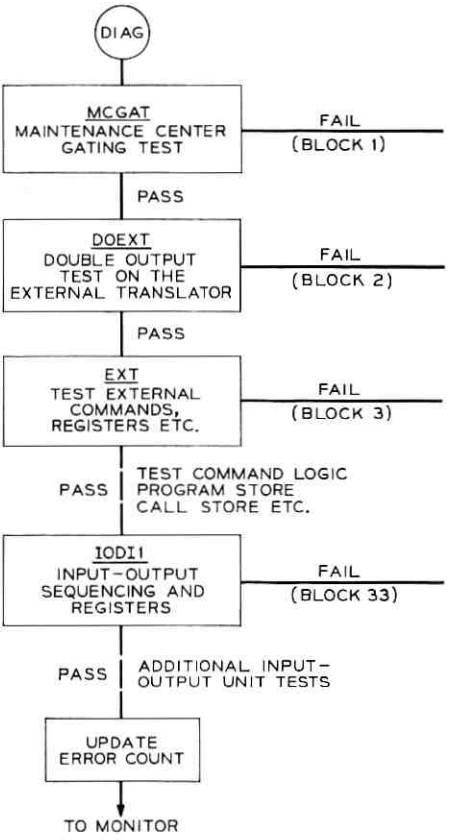



Fig. 8 — The diagnostic sequence.

In certain cases, after a given diagnostic test failure, up to two other diagnostic tests further down in the diagnostic sequence may be automatically requested in order to improve resolution. In all cases, if more than one test fails, the diagnostic printout is generated by the failing test closest to the end of the diagnostic sequence.

As shown in Fig. 9, a test block is subdivided into test segments. Numbers associated with test blocks and test segments are used to uniquely identify the failing test in a teletypewriter printout. A typical diagnostic printout shown in Fig. 10 contains a block number and segment number identifying the test that was in progress when the printout was formed. It also contains a data word indicating exactly how the test failed; for example, what bit of a particular register is bad. This information is used by the craftsman in referenc-

ing the translation section of the trouble locating manual in order to find replacement circuit packages.

A failure in a circuit containing state memory may in the worst case leave the circuit in any one of $2^N$ states if $N$ memory elements are present. If no attempt is made to initialize this state memory before a diagnostic test is run, up to $2^N$ different diagnostic printouts may be generated. In general, good on-line to off-line communications allows setting of state memory to a single consistent state. This means only one printout will be produced for a given fault independent of the control unit state at the time of failure. For certain faults, it is not possible to establish a single initial state before diagnostic testing is started. A fault of this type may produce a different printout for each possible initial state. In these cases, an attempt is made to list all possible printouts in the trouble locating manual.

Often, in order to obtain good diagnostic resolution, it is desirable to perform combinational checks of logical rather than sequential operation, since combinational checks only depend on the present inputs and not on the past machine states. However, much of the control unit normally operates in a synchronous sequential manner:

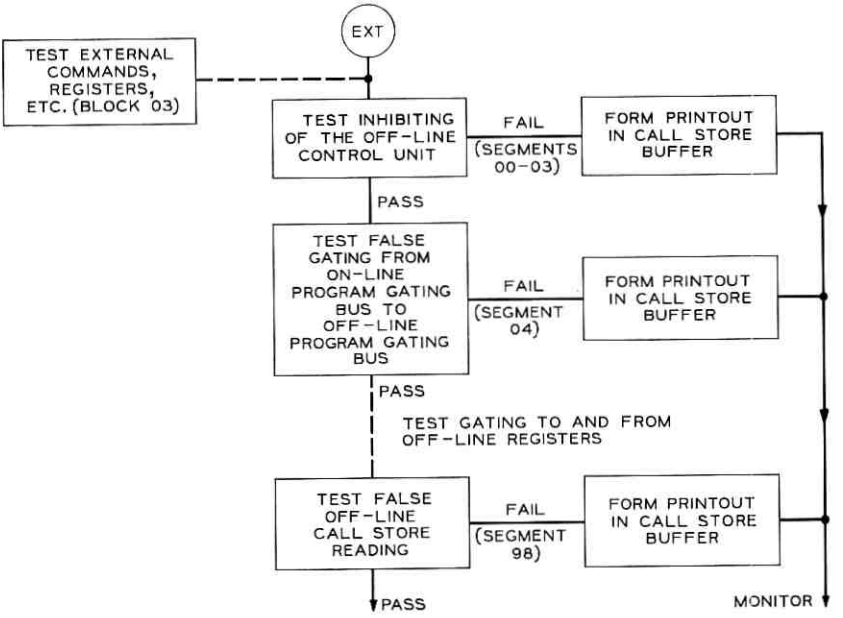



Fig. 9 — A diagnostic test block.

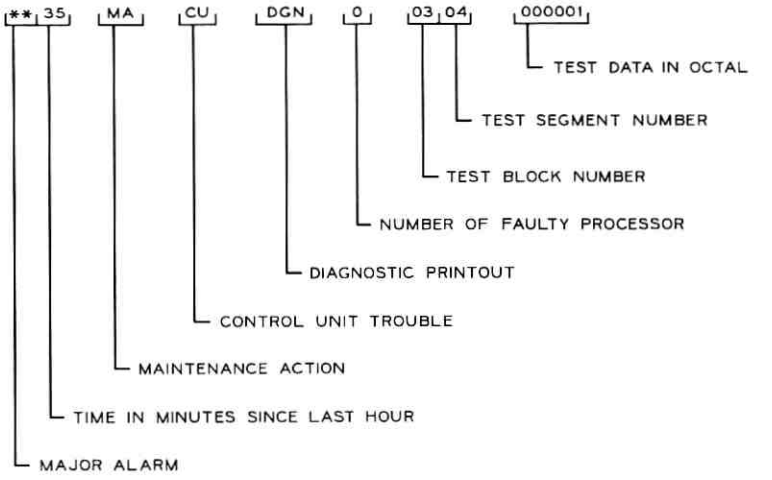



Fig. 10 — A typical diagnostic printout.

operations performed during clocked intervals are dependent not only on present inputs but also on past machine states. An operation performed during such a clocked interval can be checked combinationally if the state memory remembering the past machine states can be gated to by program and if the machine can be prevented from cycling through the sequence. Figure 11 is an example of circuitry in the input-output unit which is checked in this manner. The following steps are used to check gate SETB.

(*i*) Set a control flip-flop to prevent clock pulses from enabling input-output logic gates or resetting flip-flops.

(*ii*) Set the A flip-flop and clear the B flip-flop by direct gating via the program gating bus.

(*iii*) Enable clock signal POON for one clock interval by executing a special internal maintenance instruction.

(*iv*) Check to see if the B flip-flop is correctly set via program gating bus access.

Failure of the B flip-flop to be set results in a generation of a diagnostic printout which gives the block and test segment number of the failing test plus a test result data word.

The program timer which is an asynchronous seqential circuit is also checked in a similar manner except that, instead of stopping the

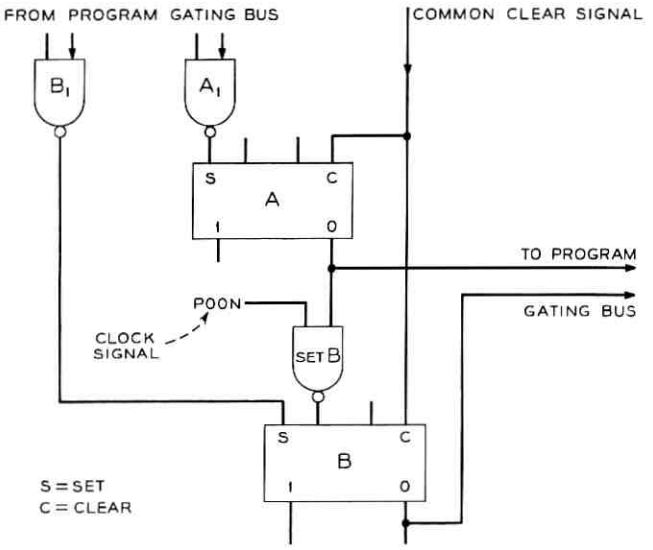



Fig. 11 — Combinational testing.

clock, leads from a control flip-flop are used to cut feedback paths and prevent sequential operation while testing is being performed.

### 4.4 *Error Strategy*

A control complex error is defined as a trouble which is detected by either circuit or program means and then disappears when diagnostic testing is performed. Errors can be produced by such things as a fault causing a marginal circuit condition or by noise which changes a 1 to a 0 or vice versa. Errors had to be considered when designing control complex maintenance programs in order to prevent them from adversely affecting call processing and to insure that a repeated error caused by a marginal circuit fault results in some attempt at corrective action.

#### 4.4.1 *Mismatch Detection*

Mismatch detection tests attempt to minimize the effect of errors or standby faults on call processing by turning on input-output work (both circuit and program) very soon after the initial mismatch. Figure 12 shows detection test ordering and actions taken at various times after the initial mismatch. Note that the off-line control unit is inhibited and a test of the input-output is performed immediately

after mismatch. If this test is successful, input-output digit scan functions are resumed. After an additional short test of program control logic (approximately 5 milliseconds after mismatch), the input-output 25 millisecond interrupt is allowed to resume.

Allowing the input-output unit and the 25 millisecond interrupt program to resume operation very soon after the mismatch, prevents incoming information from being lost when no solid fault is present in the on-line control unit.

### 4.4.2 *Repeated Errors*

The base level maintenance monitor keeps a count of the number of times automatically initiated control complex diagnostics return an all-tests-pass indication. If this count exceeds a fixed threshold in a ten-minute interval, further tests are initiated and no further automatic attempt is made to return to synchronization. Thus, some pro-

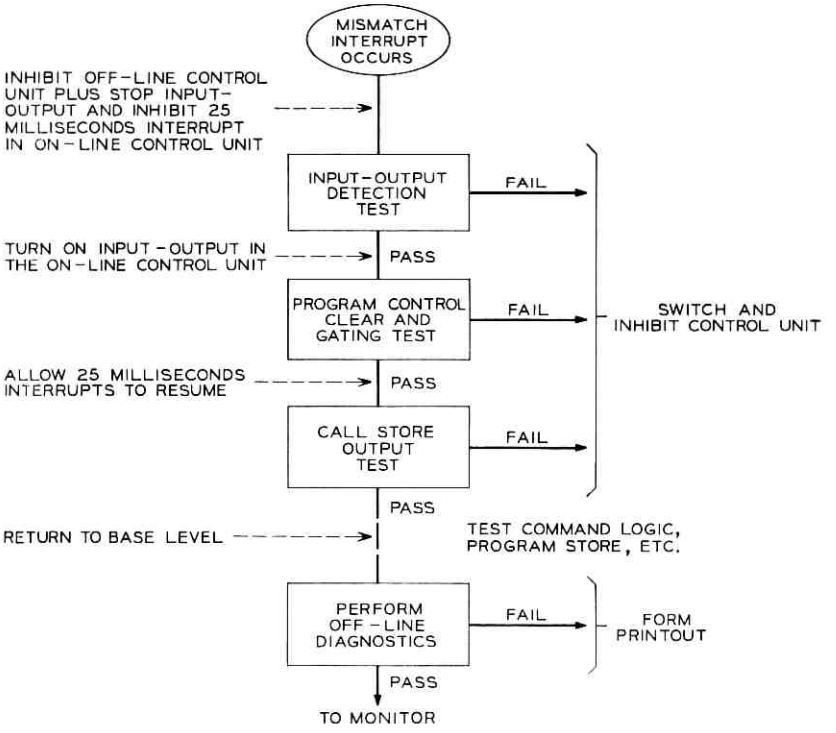



Fig. 12 — Mismatch trouble recovery.

tection is taken against transient and intermittent faults overloading the system.

## V. PERIPHERAL SYSTEM MAINTENANCE

Both duplication and engineered redundancy are used for reliability in the peripheral system.[1,3] Duplicated control circuitry (controller) is used in the peripheral system wherever a trouble would affect a significant portion of the equipment. Either controller may be accessed from either control unit. Peripheral decoders, each of which controls up to four trunk or service cricuits, are not duplicated. Each may also be accessed from either control unit. Network links, trunks, junctors, and service circuits are provided in sufficient numbers that a faulty unit can be avoided without significantly affecting service.

Troubles must be detected quickly and the faulty unit identified and removed from service. Several troubles may have to be tolerated at the same time, including those induced by the craftsman when attempting repair or making additions to the peripheral system. Maintenance programs provide a "best reasonable" mode of operation and craftsman interface. Extensive and generalized troubleshooting facilities are necessary because of the frequent equipment additions to the system and the ratio of wired-in circuitry to replaceable plug-in circuit boards.

### 5.1 *Trouble Detection and Recovery*

Many of the troubles in the peripheral system are detected by check circuitry during the normal execution of an order to the periphery. This is especially true of troubles that have a significant effect on the system where rapid detection and recovery is most important. Some troubles are detected by the call programs which check for expected results at strategic points in a call, or "become suspicious" at unlikely occurrences.[2] They may initiate further tests immediately or provide results to be accumulated for further actions when an error threshold is reached.

The remaining troubles usually do not seriously affect service and are detected by manual and automatic routine exercise, audits, and trouble reports.

#### 5.1.1 *Scanner Troubles*

The scanner organization, duplication, and interconnection are described in Refs. 1 and 3. Each of the duplicated scanner controllers

may be accessed by either control unit. For some troubles in scanning, a controller would be removed from service. For other troubles, a control unit would have to be removed from service. Since interrogate and readout windings are not duplicated, there are troubles which affect a single row of 16 ferrods, or a colmun of up to 64 ferrods for any of the four controller-control unit combinations. This prevents use of these ferrods.

Detection and recovery actions are indicated in Fig. 13 with the separation between circuit and program functions. Troubles in the selection of a scanner or row are detected by check circuits in the input-output or scanner.[1,3] The scanner may be accessed by a program order or by the autonomous input-output logic in the control unit which scans for digits and line originations.[1]

A program scan order is immediately followed by a program check for an error indication and the trouble recovery program is called as a subroutine on error. For an autonomous digit scan error, the trouble recovery program is entered by an interrupt during which autonomous scan functions are stopped. The autonomous line scan stops when a new origination, or an error, or a last row is detected. The 25 millisecond interrupt program which controls this function detects such a trouble later on rescan by a program order.

Scanner output troubles are not detected by check circuits. Some of these are detected by defensive design in programs which use the scanner. For example, supervisory scan programs suspect trouble for supervisory changes in successive rows, or more than one change in a row and call an immediate scanner output test as indicated in Fig. 13.

Some output troubles cause a control unit mismatch on a subsequent call store read or write. An output test is performed for all scanners following a mismatch. Detection of other troubles relies on routine exercise with the diagnostic test.

Individual ferrod troubles are not detected by any of these checks. These troubles show up as faulty operation of the circuits to which these ferrods are assigned.

### 5.1.2 *Scanner Trouble Recovery*

The trouble recovery programs, after verifying a trouble, first try the other controller and then the other control unit in order to identify the faulty unit. The order that failed or the test that detected the trouble is used for these attempts. A bad interrogate or readout winding is assumed and recorded if none of these tries are success-

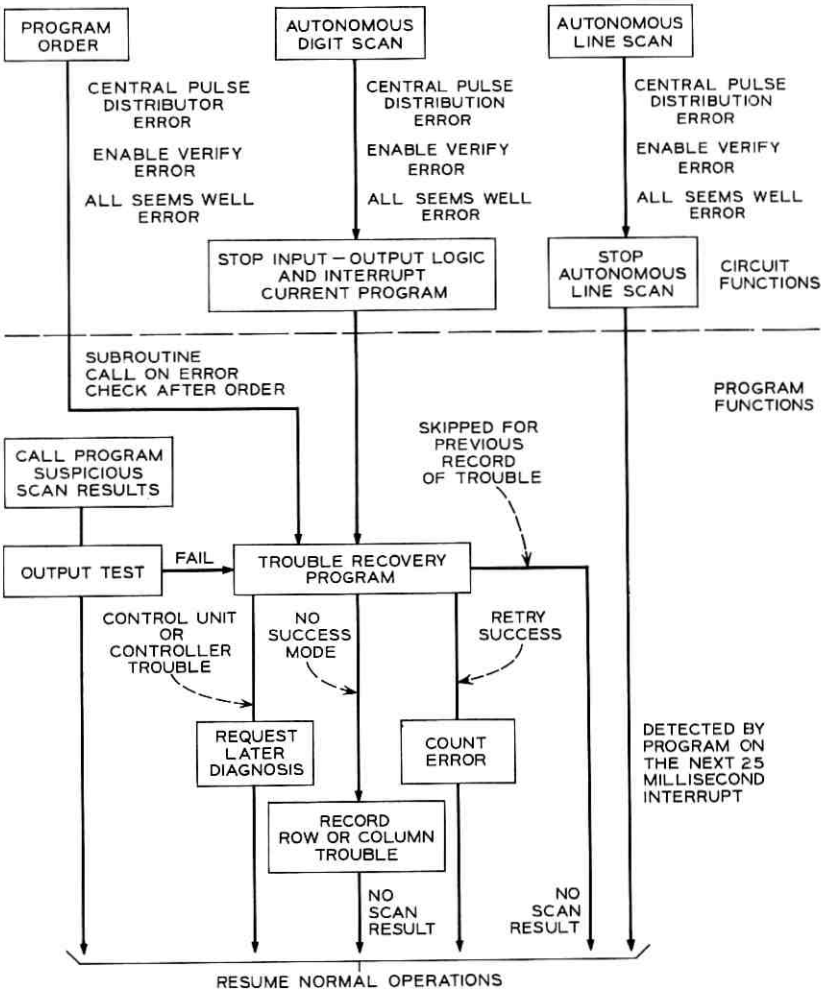



Fig. 13 — Scanner trouble actions.

ful. Once a controller or control unit is marked out of service, it will not be restored automatically or used in a later trouble recovery attempt unless the number of bad rows or readout windings becomes excessive. When the number of errors recorded becomes excessive, the out-of-service unit is restored automatically and the on-line unit removed from service. This allows the corrective action for a second

trouble to override that for the first if the second trouble has a significant effect on service.

In resuming normal operations, a later diagnosis is requested for any new trouble found, and the call operations may be cleared or skipped if a row or column cannot be scanned. For example, a row trouble detected by supervisory scanning would cause that row to be skipped, whereas a row trouble detected by digit receiving would cause the digit receiver and path to be idled and a reorder tone connected. A test of the circuits in the call to which the ferrods are assigned would be initiated.

#### 5.1.3 *Network Connection Troubles*

Orders to the network require several milliseconds to complete. In order to allow efficient use and control of the network access, orders are sent by the program to all networks in an order execution cycle of up to 10 milliseconds every 50 milliseconds.[2]

As indicated in Fig. 14, some check results are available immediately after the network order and path data is sent to the network.[1,3] Other check results are not available until the network order has had time to complete its operation. These latter check results, which are indicated by status ferrods, are checked for all network controllers just before beginning a network order execution cycle every 50 milliseconds. Any errors that have occurred on the last order execution cycle are known at this time and cause the trouble recovery program to be started.

#### 5.1.4 *Network Trouble Recovery*

Each network order attempt by the trouble recovery program will take at least 50 milliseconds and so must be interleaved with call processing. The trouble is verified by retrying the order. The peripheral order buffer from which the order was sent must be determined so that the call may be stopped and the path and order data obtained. The other calls are not affected. The order is retried first with the other controller, if in service, and then with the other control unit, if synchronized, and the controller or control unit removed from service if found bad.

Since a sequence of network orders are sent before results are checked, a control unit fault may result in several controller troubles from that sequence of orders. In this case, the other control unit is tried first to minimize delays. In other cases, controller trouble indica-

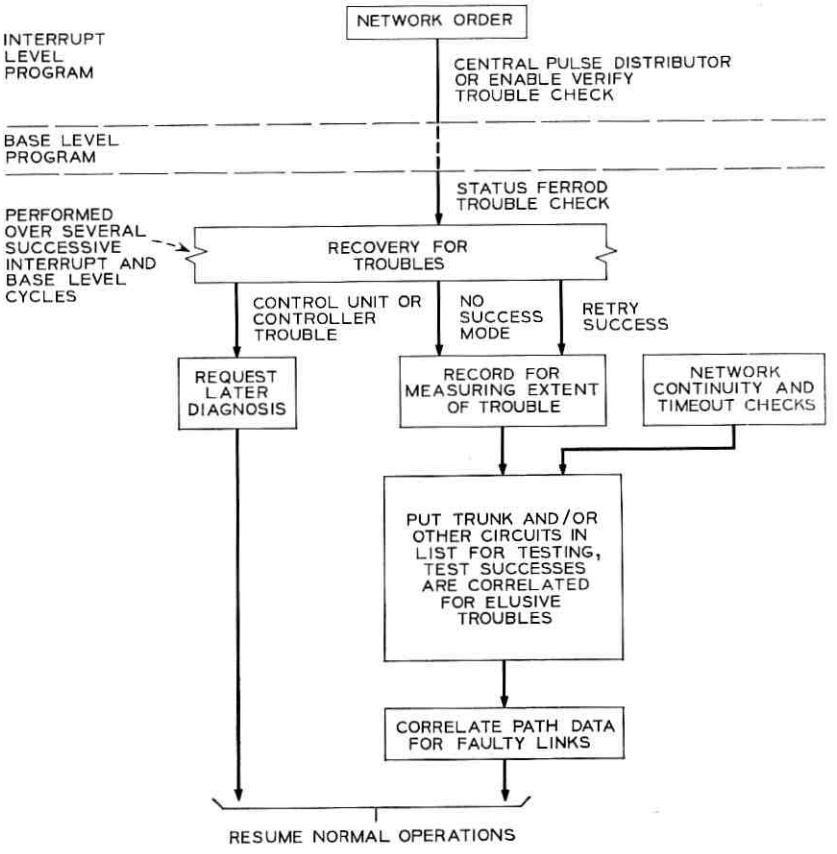



Fig. 14 — Network trouble actions.

tions are handled one at a time and a second trouble occurrence while resolving the first, stops all order execution until the first trouble is resolved. The controller status ferrods may indicate trouble such as loss of power or input noise when no order was sent. In these cases, no peripheral order buffer that accessed the controller will be found, and the entire sequence of network orders for that execution cycle is repeated to verify the trouble and try other combinations of controller and control unit.

Trouble indications may also occur because of nonduplicated network circuitry and the order will not be successful with the duplicate controller or control unit, or possibly the duplicate controller or control unit may have been left out of service by a previous trouble. Path data are accumulated in these cases to determine the extent and fre-

quency of the trouble. Faulty links are removed from service through correlation techniques. If the trouble is extensive in the frame and affects many input terminals, the other controller or control unit is automatically restored to service, as was the case with scanners. A record of the number of peripheral units of any type with extensive trouble with either control unit is maintained in this way so that the best control unit can be favored.

Lack of continuity in the network is usually detected by call processing. Path data for no continuity are correlated with that for other path troubles so that faulty links can be removed from service.

After a successful network trouble recovery attempt, the operations for that call are continued. If the network trouble recovery is unsuccessful, the call is torn down and a reorder tone is connected where possible.

#### 5.1.5 *Trunk and Service Circuit Access Troubles*

The central pulse distributor and its supplementary pulse distributor are used to change trunk and service circuit relay states. Selection and output troubles for these units are detected by check circuits when program orders are sent, and the trouble recovery actions are similar to that described for scanners. A central pulse distributor trouble may require use of the other control unit whereas a supplementary pulse distributor trouble may require use of the other controller or, for some troubles, the other control unit.[1,3] It may not be possible to find a successful configuration for some output troubles. In these cases, the output number is recorded and the circuits that are assigned to that output cannot be used. Trunk and service circuit troubles may also show up as troubles in setting up a network connection or in a time out of some operation in the call where there are several possible sources of the error. When such circuits are possible sources of trouble in a call, they are put in a list for testing. The network links and those circuits which pass their tests or which cannot be tested are correlated on successive troubles as indicated in Fig. 14 to help isolate the source of the trouble. Various operational trunk and service circuit tests are further described in Ref. 2.

### 5.2 *Diagnostic Organization and Use*

#### 5.2.1 *Common Controllers*

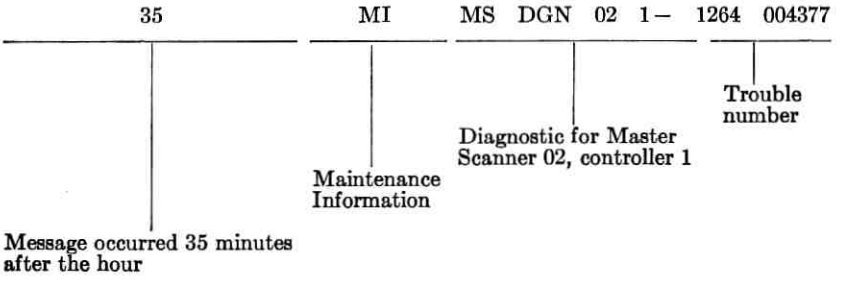
The network, scanner, and supplementary pulse distributor controllers are tested by a sequence of functional tests. For example, the functional test blocks for the scanner controller are:

(*i*) Access test with proper addresses, observing check circuit results for trouble.

(*ii*) Check circuit test using illegal addresses to see if the check circuits are capable of indicating trouble.

(*iii*) Output test using a test signal that induces outputs on all 16 readout leads.

Each of these functions is relatively independent of the others and the trouble number is formed from results of the first functional test block which fails. Several test results within a functional test block are usually combined in the test number to improve the resolution and provide error data to help the craftsman when the *Trouble Locating Manual* listing is insufficient. For example, in the scanner controller, the one-out-of-64 ferrod rows is selected by an 8-by-8 coincident core matrix.[3] The two one-out-of-eight selections are received over 16 leads of the peripheral unit address bus. In the access test block, all 64 rows of the ferrod matrix are accessed in sequence. The OR function of the proper peripheral unit address bus contents for all failing addresses provides a 16-bit error data field (6 octal characters) which is part of the trouble number in the diagnostic message shown here:




In a particular test block, failure modes, such as bus receiver or core driver faults, are easily analyzed by class when designing the program for determining the associated trouble numbers. These failure modes can then be ignored in the program design and analytical generation of *Trouble Locating Manual* entries for any functional test block that follows in the sequence.

Peripheral system access troubles may be caused by faults in the control unit access that are not detected by the control unit diagnostic program. The trouble recovery programs would normally leave the faculty control unit off line. The same functional tests that are used for testing a controller are also used for testing the access to a con-

troller from the off-line control unit. Additional tests are needed in some cases for testing input-output control circuitry that is not tested by the control unit diagnostic program.

When testing the control unit access, the orders are executed from the off-line control unit, and the first character of the trouble number is modified so that different equipment may be indicated by the *Trouble Locating Manual* listing. The listing for these trouble numbers is only accurate if the trouble is in the control unit access and not in the controller, so the request for diagnosis from the off-line control unit is rejected if there is a controller trouble from the on-line control unit. In addition, the off-line control unit is automatically diagnosed and must pass before the access is diagnosed. Diagnosis of a controller or controller access is also rejected unless the duplicate controller is in service for call processing use. For simplicity, only one diagnosis may be in progress at any one time and it will be aborted if a new trouble occurs while it is in progress.

A separate *Trouble Locating Manual* is provided for each type of peripheral unit.

#### 5.2.2 *Use of Diagnostics*

The diagnostic programs are also used for restoring equipment to service after repair, for giving the peripheral system automatic routine exercise, and for testing new equipment additions.

Controllers and their access are tested about once a day in a low traffic period, except for scanner output where service affecting troubles are not detected by check circuits while in use. These are tested more frequently (about every minute or faster, traffic permitting).

Remote execute facilities, with pushbuttons and indicator lamps located on various frames throughout the office, allow tests previously specified by teletypewriter to be stepped through or repeated by operating a pushbutton. Pass-fail results are indicated on lamps which are part of the remote execute facility.

### 5.3 *Peripheral System Growth*

New equipment added in an operating system for growth and for a new installation must be tested thoroughly before it is put into service. In an operating system, the installing and testing should have a minimal effect on service. Troubles that occur during shipment, installation errors, and any troubles that did not show up in factory testing

must be corrected at this time. The use of connectors for interframe wiring reduces the installation interval. It also reduces the wiring errors since the connector wiring can be tested at the factory.

The diagnostic test programs described are used where possible to help locate these troubles. The *Trouble Locating Manual* is not satisfactory for this purpose because several troubles and wiring errors may be present. Entries in the *Trouble Locating Manual* were formed by predicting test results for a single failure that could occur in an operational system. Specifying test results for combinations of trouble is impractical and the craftsman must trace the symptoms manually. The diagnostic program provides a means to exercise all functions of the equipment and identify the trouble symptoms. The diagnostic program may print out a failing order or functional operation, for example, instead of a trouble number. The order or functional operation may be requested, with the repeat option if desired, and the operation observed by oscilloscope or other test instruments. This method of locating faults is also useful for troubles that occur in operation in the few cases where the *Trouble Locating Manual* is insufficient.

Additional test programs and manual test procedures are needed for some of the possible troubles at installation. Network link troubles, which are identified by error correlation techniques in normal use, require a program test for any new network frames added. Translation data in the memory, which relates to equipment in the office, changes when equipment is added, and must also be verified.

Such units as the scanner and supplementary central pulse distributor tie into the common peripheral unit address bus and scan answer bus. The off-line control unit is not usable until the bus leads are connected and verified for correct wiring, polarity, and waveform in both directions from the new frame added. Interframe bus connectors help to minimize the exposure of the system to other troubles during this operation. If other troubles do occur, the system will stabilize in a "best" operating mode, and the craftsman may reinitiate this decision process, if necessary, after restoring the bus integrity.

Half of a scanner ferrod matrix (512 ferrods) may be added during growth. In this case, the output leads must be disconnected and the additional ferrods added in series with minimal effect on normal use of the other ferrods. Here again connectors are used to allow rapid change over and recovery in case of trouble in matrix addition. The repetitive test mode allows rapid indication of a trouble on pass-fail

lamps and provides a diagnostic printout so that the craftsman can minimize the time that a faulty matrix addition is connected.

The network may be increased in size by the addition of a network control and junctor switching frame or by adding line or trunk switch frames to an existing network control and junctor switching frame.[3] The network control and junctor switching frame contains new network and scanner controllers. Both the access to these from the off-line control unit and the controller themselves are tested using the diagnostic tests. Controllers access to the line-trunk switching frame is also tested with the diagnostic tests. In both cases, the network links are tested with a special network fabric test program. While testing the network control and junctor switching frame, the junctors are connected back into the same network control and junctor switching frame at the junctor grouping frame in a standard test pattern. A junctor reassignment over all network control and junctor switching frames must then be performed before the new network control and junctor switching frame can be put into service. The junctor translation tables are updated in memory to indicate both the existing assignments and the new assignments. The junctors are segmented into four parts at the junctor grouping frame and only one of the segments is removed from service at a time for junctor modification. A verification of the new junctor connections with the translation tables and a test of circuit junctors is made for each segment before those junctors are put back into service. After the reassignment is complete, any new trunks, service circuits, and lines to be added are tested and put into service. The common control frames such as the scanner or supplementary pulse distributor are, of course, tested before the circuits they monitor or control are tested.

## VI. MAINTENANCE MONITOR

The base level maintenance monitor is the primary noncall processing executive program in the office and all maintenance programs come together through it. The responsibilities of the monitor include:

(*i*) Recognizing changes in the system's state and the initiating a proper response to a state change (state detector).

(*ii*) Controlling almost all base level maintenance programs in the office (scheduler).

(*iii*) Monitoring all maintenance input messages from the craftsman.

(*iv*) Initiating periodic work that must be done on a time schedule.

(*v*) Miscellaneous functions such as timing and controlling system

alarm conditions and checking the integrity of the entire base level scan.

Figure 15 shows of the "information flow" through the main parts of the monitor.

### 6.1 *System State Detector*

State changes in the system are either caused by trouble detection mechanisms or are induced by the craftsman from the maintenance center or via the teletypewriter.

It is the state detector's responsibility to look over the last scan and detect and take action on the occurrence of: (*i*) a mismatch interrupt, (*ii*) an input-output trouble recovery action, (*iii*) a system initialization or restart, (*iv*) the failure of an automatic error detecting circuit or a periodic control unit detection test in either control unit, and (*v*) manually initiated changes such as putting the system into the manual mode of operation.

When the state detector sees that a change has taken place, it initiates an output message identifying the change, records the occurrence of a control unit switch, and lets the craftsman known what automatic action will be taken.

In addition, the state detector feeds state information to both the maintenance program scheduler and the teletypewriter maintenance input message monitor which enable these programs to control items already in progress in the system or coming into the system via the teletypewriter. For instance, when a craftsman puts the system into the manual mode, the state detector feeds the necessary information to the maintenance program scheduler to inform it that he has taken control over the off-line control unit, and no automatically initiated maintenance program from then on should interfere with him.

The state detector also guarantees a consistency of hardware control in the system. As an example, when the craftsman goes to the manual mode, the state detector blocks all interrupt signals from the maintenance center. If the craftsman wishes to generate a manual interrupt, he must then type in the necessary input request message which will store what the craftsman wishes to do when the interrupt occurs and release the interrupt "block."

### 6.2 *Maintenance Program Scheduler*

The heart of the maintenance monitor is the maintenance program scheduler. The scheduling algorithm is fairly simple and is tailor fit

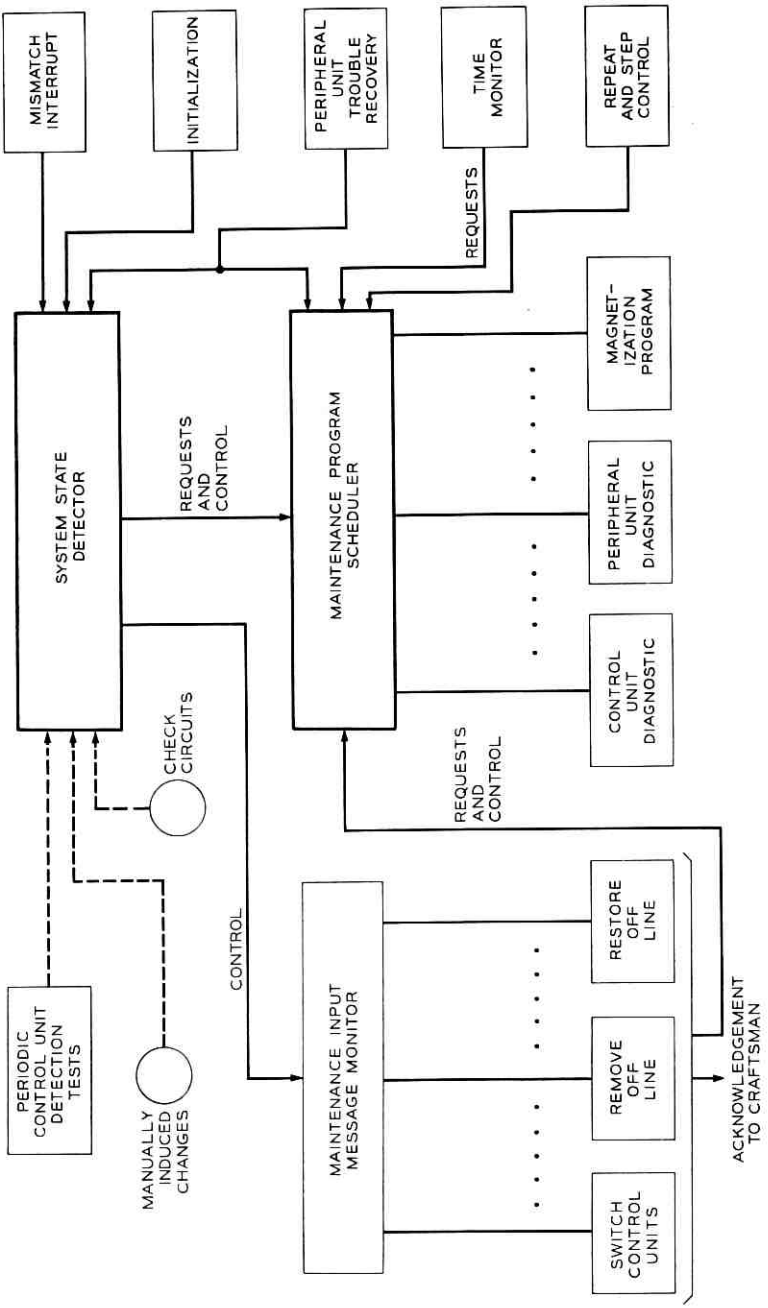



Fig. 15 — Maintenance monitor information flow.

to the type of programs it controls. These programs involve interface with the craftsman and are all multiscan (that is, they are allotted only a certain amount of real-time per scan and take many scans to complete). In general, the scheduler will allow only one such program to be in progress in the system at any time. This avoids both confusion by the craftsman and interference problems which can result from running these programs concurrently. For instance, control unit diagnostics program assumes it has absolute control over the off-line control unit during the entire time it is in progress and no other program can change off-line register or memory contents. Off-line peripheral unit diagnostics assume the off-line control unit is healthy and can produce misleading information were they to be initiated while control unit diagnostics are in progress.

The basic "one-at-a-time" algorithm of the scheduler should also help desensitize the system to potential troubles which show up in different areas depending on the time of occurrence.

The scheduler operates on a four-word memory block representing a matrix of four rows and 16 columns. Each maintenance program is assigned one or two column positions and has associated with it a "request," an "in progress," an "allow," and an "abort" bit. Figure 16 is a picture of the matrix showing the matrix positions of the programs under control of the scheduler. Any initiator of one of these programs (teletypewriter or automatic) simply sets the proper request bit in the matrix. The scheduler will then analyze this request with regard to whether the program is "allowable" in the present system state (the allow word is initialized every scan by the system state decoder) and whether a higher priority maintenance activity is now in progress in the system. The priority of the various programs is represented by their column positions in the matrix. This *a priori* priority structure can be determined by considering both the programs themselves and the request source. The various maintenance programs in the system operate in a realm of concentric circles in their importance to the system and in their assumptions (that is, off-line peripheral unit diagnostics assume that the off-line control unit is healthy). Automatic requests are triggered by changes in the system state and by automatic trouble detection signals; consequently they are more urgent than teletypewriter requests.

The scheduler will take the abort entry associated with each program if: (*i*) a higher priority request enters the system while it is in

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| INPROG | | PU EXER-CISE | CU EXER-CISE | SCW TRANS UP-DATE | | OFF PU DIAG TTY | ON PU DIAG TTY | CU DIAG TTY | | UP-DATE & SYNC | | | OFF PU DIAG AUTO | ON PU DIAG AUTO | | CU DIAG AUTO |
| REQEST | | | | | | | | | | | | | | | | |
| ALLOW | | | | | | | | | | | | | | | | |
| ABORT | | | | | | | | | | | | | | | | |

CU = CONTROL UNIT
DIAG = DIAGNOSTIC
AUTO = AUTOMATIC
PU = PERIPHERAL UNITS
SYNC = SYNCHRONIZATION
TTY = TELETYPEWRITER
SCW = SINGLE CARD WRITER
TRANS = TRANSLATION
INPROG = IN PROGRESS
REQEST = REQUEST

Fig. 16 — Maintenance program scheduler matrix.

progress, or (*ii*) a change in the system state occurs and the state detector marks it nonallowable.

Abort routines are necessary to prevent erroneous decisions by maintenance programs. These decisions can result in the improper removal of equipment from service and in misleading information on the teletypewriter. They also serve to let the craftsman know what is happening in the system. Further, in the case of a higher priority request, the time cannot be taken to let the lower priority activity finish but the request can be held until the abort is complete. The abort programs, if necessary, can distinguish between the two reasons for entering the abort by looking at the allow bit. For instance, the abort for the magnetization program (single card writer translation update) will not force the craftsman to start from the beginning if it finds a higher priority request entered.

Various minor options are available with the scheduler. By use of logic masks, subsets of maintenance programs can be allowed to run concurrently. The craftsman, via a teletypewriter overwrite, can control the state detector and either allow or not allow a program to run until he tells the system otherwise.

The scheduler also takes care of the control of "repetitive" and

"step" functions (remote execute facility). The craftsman can specify the repetitive or step option on an input request for a maintenance program. The scheduler will then either continuously repeat, or repeat on signal the program requested and provide output on the teletypewriter or in lights depending on which option was chosen. The scheduler will allow only one repetitive or step function to be in progress at any one time. The repeat or step control is very useful in the repair procedure.

The maintenance programs not under control of the scheduler include the service circuit and trunk tests, and tests associated with the ringing and tone plant and the automatic message accounting unit. Many of these tests are progress mark routines operating out of transient call records. The tests do not require the complex control of the programs under the scheduler and do not interfere with other maintenance activity in the office. As far as the maintenance monitor is concerned, the tests behave like a typical call in progress. The monitor, however, still is responsible for initiating these tests periodically and the tests must be aware that a major maintenance action has occurred.

### 6.3 *Maintenance Input Message Monitor*

The maintenance input message monitor routes the various messages from the teletypewriter to the proper subprograms and performs validity checks common to all inputs. If the craftsman specified a priority on the input or used the repetitive or step option, these items are checked for validity. In addition, the present system state computed by the state detector can be compared with the states allowable for a given input and the message rejected if the state is not correct.

Input messages to change the system state (switch control units, remove off-line equipment from service, restore off-line equipment to service, and so on) or to provide system status information, are processed wholly within the monitor. Messages that request programs under the scheduler's control merely set the proper request bit in the matrix.

### 6.4 *Time Monitor*

The time monitor initiates all periodic activity that operates on a time schedule. It must deal with time spans from seconds up to once a day.

Status information at the maintenance center, such as the stand-by lights, is updated once per second and small interval timing is provided for traffic measurements.

The time monitor serves to initiate periodic exercises on the control until and the periphery during low traffic hours. These programs seek to find failures by exercising all equipment, thus avoiding the unnecessary failing of calls. The control unit exercise checks all the special maintenance circuitry in the processors, performs store margin tests and checks the ability of the control units to switch.

### 6.5 *Miscellaneous Functions*

The alarm monitor keeps track of various alarm conditions for the office, such as fuse failures, and times local alarms if the alarms are transferred to a remote location. The fuse ferrods are checked at intervals from the time monitor.

The maintenance monitor is responsible for resetting the program timers and checking the integrity of the entire base level scan. Software checks are made to detect program skipping and the program timers protect the system against program looping.[1]

During every scan the monitor calls in the teletypewriter base level processing program and programs which deal with trouble correlation and measurements. Counts are kept of almost all troubles in the system ranging from customer receiver troubles to failure of control unit diagnostics. These service and performance measurements ("plant") should give the craftsman a good picture of the total "health" of the system.

## VII. DATA MAINTENANCE

One of the primary maintenance objectives for any electronic switching system is to insure the best possible integrity of call store information in the system during periods of trouble. Call store mutilation can result from hardware faults and intermittents, and from program bugs. During periods of nonsynchronous operation, the mismatch detection mechanism is lost to the system and the detection of some hardware troubles will be delayed until a specific periodic test finds them (that is, those that would not be caught by the other automatic error detecting circuits). In this situation, data multilation can occur between the time the trouble occurs and the time it is detected.

System response to transient and intermittent failures depends on programs recognizing a high error signal occurrence over a period of

time. Very infrequently accessed branches of the program can have bugs which will mutilate some memory despite the best effort in program debugging prior to cutover. It is interesting that the mismatch detection mechanism and synchronous operation is not much help for program bugs since for most bugs the processors will maintain synchronization. In this sense, a program bug is equivalent to two simultaneous and identical hardware failures.

### 7.1 *Preventive Techniques*

The potential effect of any of these problems on memory is highly dependent on the basic program algorithms of the system. The ease of communication among programs, the absence of linked list structures, and the per call assignment of major blocks of memory significantly aids the task of data maintenance. For instance, the progress mark approach to call processing taken by the No. 2 ESS assigns an arbitrary block of call store (transient call record) to each call which remains fixed while it is being processed ("transient"). Additional storage associated with the call is added as necessary (peripheral order buffers and originating registers). Each progress mark routine is entered with the transient call record or peripheral order buffer block address as data, and the routine works within the block with relative addressing. The scope of a progress mark routine is then a small set of data relating to a single call.

In addition, defensive programming techniques are used throughout the No. 2 ESS. Table indexes obtained from call store are range-checked or small tables are made complete to cover all possible index values with invalid indexes pointing to error routines. Programs that transfer, based on the call store, check the address for all zeroes first in case the recovery programs had zeroed this word so that they will not continue to transfer wildly. Translation information is obtained by accessing a master table index which will insure that, with bad data in call store, program instructions will not erroneously be read as data with a resultant parity error failure. The intent of such defensive programming techniques is to allow processing to continue in the face of bad data and limit the effect of the data to one or a few calls.

### 7.2 *Corrective Techniques*

Despite the algorithms employed and the defensive techniques used, programs are still required which will detect and recover from

bad data. In the No. 2 ESS, these involve (*i*) audit programs, and (*ii*) system recovery or initialization programs.

### 7.2.1 *Audits*

The No. 2 ESS call store memory contains many items of redundant information in different forms, some associated with individual calls and other information primarily equipment oriented. The memory also contains links connecting blocks associated with a call. It is the function of audit programs to ascertain whether these various items in memory are consistent.

Separate audit programs are written for the various memory blocks such as the transient call records, terminal memory records, line status bits, originating registers, peripheral order buffers, and the network map.[2] For example, the originating register audit program checks for a correct linkage from the originating register to a terminal memory record and transient call record. When an audit program finds inconsistencies, it attempts to idle the memory blocks and, if possible, the corresponding equipment. The audit programs are called in periodically from the time monitor, can be initiated from the teletypewriter, and form an important part of the system recovery strategy.

### 7.2.2 *System Recovery*

The system recovery program (or "initialization" program) is triggered by hardware (program timers[1]) upon the occurrence of a control unit switch when the system was not running in synchronization. Multiple control unit switching while in the synchronous mode will also activate it. Control unit switching can be caused by bad data and software bugs as well as hardware failures. For example, the program timers will switch control units if the program hangs up in a loop. The control unit switch itself is the proper response to a hardware failure and the recovery program's job is to:

(*i*) Insure the new on-line call store is reasonably consistent with the state of the periphery regardless of the cause of the control unit switch. When the off-line control unit is inhibited and is not under the control of a craftsman, the base level maintenance monitor activates circuitry which causes the off-line call store to automatically track the on-line call store. If a control unit switch occurs, the early "phases" of recovery can assume the new on-line call store is consistent. If, however, a switch occurs while the craftsman is in control,

the entire new off-line call store must be moved across to the new on-line call store.

(*ii*) Clear memory in increasingly larger segments in an attempt to stabilize the system if software is at fault.

A typical sequence of recovery attempts would involve:

(*i*) Isolating the program in control of the system at the time of the control unit switch, and taking appropriate action on memory blocks associated with the program (either clearing the blocks or marking them bad for later action).

(*ii*) Calling in the audit programs in an emergency mode.

(*iii*) Clearing all transient data in the call store while preserving the stable call records.

The craftsman can force in the recovery program and he alone can trigger the initialization of the stable data. Any recovery attempt will notify the base level maintenance monitor to abort any maintenance program in progress in the system.

## VIII. EXAMPLE OF SYSTEM TROUBLE

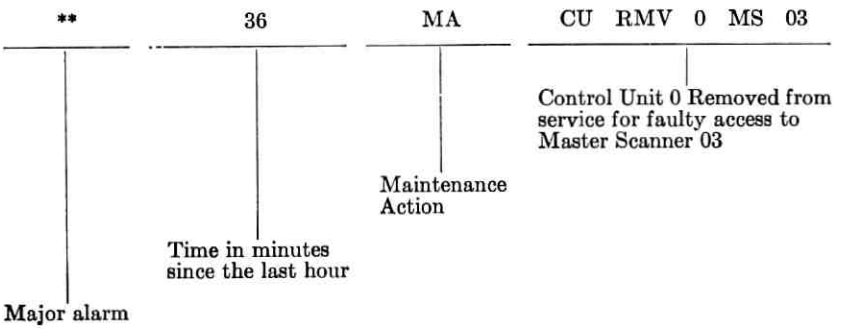
We will now go through a trouble example where a fault occurs in the input-output unit of the control unit affecting the access to a peripheral unit. We describe the sequence of actions for detection, recovery, discontinuing the troubleshooting that was in progress, diagnosis, and repair.

Assume that network controller 1 for line trunk network 2 is out of service and the craftsman is troubleshooting this controller. He has typed in a request to repetitively execute an order to the controller. Assume, at this time, that the system is in synchronism with control unit 0 on-line and a central pulse distributor enable translator gate fails (see Fig. 17). This failure is first detected by a program scan order to master scanner 3, controller 0.

### 8.1 *Trouble Detection and Recovery*

The failing scan order returns an indication which results in a program transfer to a maintenance recovery program. This program verifies the order failure and retries with duplicate controller 1 which, for the trouble specified, will also fail. The recovery program then switches control units and again retries the order which now will be successful.

The new off-line control unit 0 is marked out of service with bad access to master scanner 3 recorded. The following message is printed:




Both controllers of the scanner remain in service from the on-line control unit 1. A request is made for diagnosis of the stand-by control unit 0 and its access to master scanner 3 controller. In addition, the base level maintenance monitor is notified to abort any conflicting activities. The trouble recovery program then returns control to the call program. Only a few milliseconds have elapsed since the trouble was detected and the scan order was accomplished, so service is not affected.

When the maintenance monitor gets control at the next end of scan, it will cause the repetitive order operation that the craftsman initiated to be aborted. The craftsman will be informed of this action by the pass-fail lamps both being dark, and by the following tele-

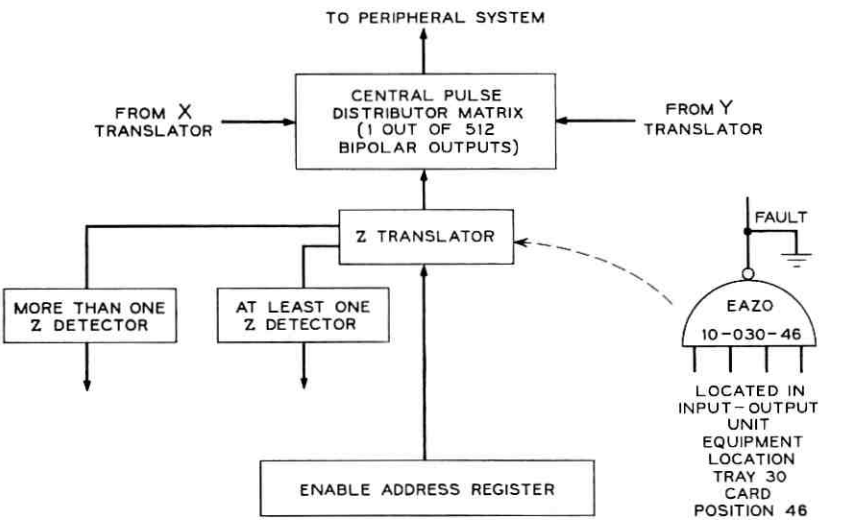
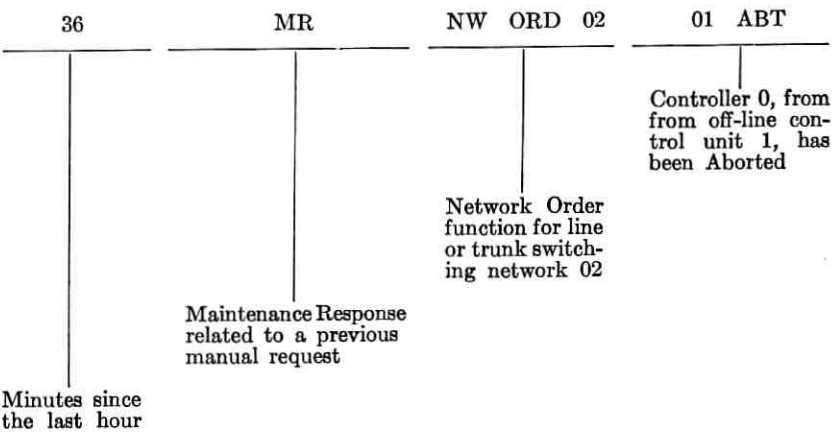



**Fig. 17 — A control unit fault.**

typewriter message:




8.2 *Diagnostics*

Once the abort is completed, the monitor will recognize the diagnostic request made by the recovery program and diagnostics will be initiated interleaved with call processing. In this case, control unit diagnostics will fail in an input-output circuitry test block.

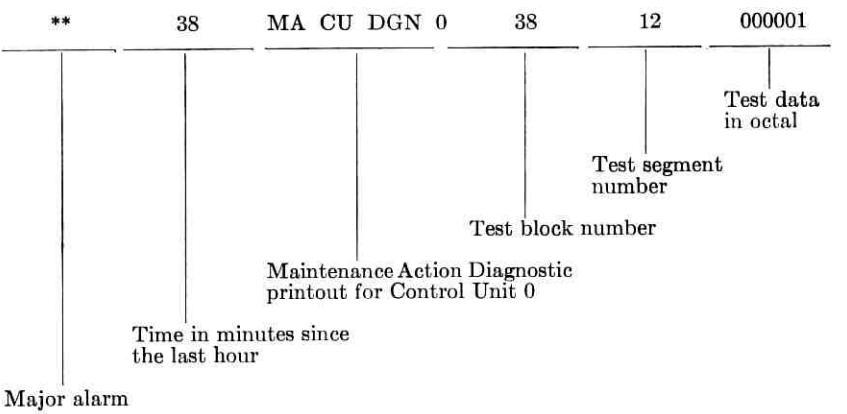
As shown in Fig. 9, input-output circuitry is tested by test blocks located near the end of the diagnostic sequence. This means input-output circuitry can be tested by programs run entirely in the off-line control unit since almost all off-line program control circuitry has been previously tested.

The on-line control unit uses external commands to first set up an off-line program starting address and then to start the off-line program control. At some later time (after sufficient time to allow test completion), the on-line control unit looks at the state of the off-line unit to determine if the off-line unit passed or failed its diagnostic test. If it failed, the state of off-line registers is used to form the diagnostic printout.

Diagnostic test block 38 tests central pulse distributor circuitry in this manner. Internal maintenance commands are executed in the off-line program control with the input-output stopped in order to test the Z central pulse distributor translator. The enable address register is first set to an address which should select a particular Z translator gate. The translator is then enabled using a maintenance instruction which generates clock pulses for one clock interval. If the Z translator does not fire properly, one of Z check circuits will produce an

error signal: the "not more than one Z detector" or the "at least one Z detector." The "at least one Z detector" will produce an error when the EAZO gate is selected because no output can be produced. In test segment 12 of block 38, the off-line program control exercises all 16 Z translator gates and accumulates a data word which has a 1 corresponding to each translator gate which resulted in an "at least one Z detector" or a "not more than one Z detector" error signal. For the EAZO gate output ground failure, the data word is 0000000000000001 in binary or 000001 in octal code.

The off-line control unit stops itself when the Z translator data word is found to be nonzero. The on-line control unit finds that the off-line control did not successfully complete the central pulse distributor test and uses the off-line state to form a diagnostic printout. The block number (38) and the segment number (12), plus the data word 000001 in octal form, are used to form the diagnostic printout:

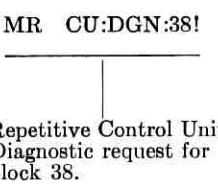



The craftsman uses this diagnostic printout to look up the translation section of the *Trouble Locating Manual* to obtain a list of replacement circuit packs. Figure 2 shows the *Trouble Locating Manual* entry corresponding to the diagnostic printout obtained for the Z translator failure. Notice that the *Trouble Locating Manual* entry for 3812 contains a short explanation of the failure area. The circuit packs are identified by the circuit pack location (I0-030-46) and the circuit pack type (A403).

### 8.3 *Repair*

After obtaining the list of replacement circuit packs, the craftsman next requests continuous execution of the failing test block by tele-

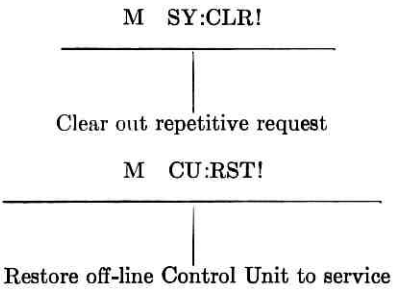
typewriter:




This request should produce a verification that the failure still exists by producing a teletypewriter printout identical to that originally obtained and should cause the fail light on the maintenance center panel to come on. This fail light is turned on by program each time the diagnostic fails. The maintenance center panel pass light is turned on if the test passes.

Circuit packs are replaced with this request running in the active control unit. Of course, off-line power must be removed while a circuit pack is being replaced. In this case, almost immediately after the first circuit pack (I0-030-46) is replaced and power is restored, the fail light should go out and the pass light should come on indicating the trouble has been fixed.

After the fault has been repaired, the craftsman can now type teletypewriter requests to remove the diagnostic test and to restore the off-line to service:




These requests should turn off the out-of-service light on the maintenance center panel and put the two control units back in synchronism, after first successfully completing another test of the standby control unit and its access to master scanner 3 controller 0.

The craftsman may now return to the original problem in controller 1 of line trunk network 2 by again requesting a repetitive network order.

REFERENCES

1. Browne, T. E., Quinn, T. M., Toy, W. N., and Yates, J. E., "Control Unit System," B.S.T.J., this issue, pp. 2619–2668.
2. Andrews, R. J., Driscoll, J. J., Herndon, J. A., Richards, P. C., and Roberts, L. R., "Service Features and Call Processing Plan," B.S.T.J., this issue, pp. 2713–2764.
3. Digrindakis, J., Freimanis, L. Hofmann, H. R., and Taylor, R. G., "Peripheral System," B.S.T.J., this issue, pp. 2669–2712.

# Apparatus and Equipment

By CHESTER W. LONNQUIST, JOSEPH C. MANGANELLO, ROBERT S. SKINNER, MICHAEL T. SKUBIAK, and DONALD J. WADSWORTH

(Manuscript received March 6, 1969)

*The No. 2 ESS achieves a standardized modular design that is easy to engineer, manufacture, install, operate, maintain, and administer. The use of much existing standard ESS hardware combined with thin film integrated circuits, plug-in growth units, and judicious interframe connectors, has contributed to the development of an economically attractive system for the small central office market.*

## I. INTRODUCTION

A small electronic switching system, even with the installation of 100 to 200 offices per year, would require production of a modest number of frames. Similarly, a new small electronic switching system cannot by itself support a large number of new apparatus codes that are normally incorporated in new system developments. Therefore, the No. 2 ESS has been designed to utilize much of the No. 1 ESS apparatus and thereby take advantage of the high volume, low-cost items associated with that system.[1,2] No. 1 ESS type peripheral equipment is also utilized: ferreed networks, ferrod scanners, and similar trunk and service circuits. Except for the smaller number of frames, a No. 2 ESS office appears strikingly similar to a No. 1 ESS office.

Utilizing existing standard apparatus provides a further dividend within the Western Electric Company. Start-up costs and development of machinery to fabricate new apparatus items are minimized, and existing standards and frame assembly techniques can be used. For these reasons, the first No. 2 ESS model at Bell Telephone Laboratories, Indian Hill, Naperville, Illinois, more nearly represents a production system than other early models for previous developments. This contributed significantly to minimizing development intervals.

Economies are realized in the system by controlling the size of

growable units so that incremental growth unit costs, which are important in a small office, will not be prohibitive.

One example (others are discussed later) is the No. 2 ESS program store, which can grow in pluggable units of 16,384 words of 22 bits per word, contrasted with the No. 1 ESS growth unit of 131,072 words of 44 bits per word.

To further reduce costs, simplified installation is emphasized. The control complex frames are interconnected with connectorized cables. This allows a complete factory test of the entire control complex thus contributing to simplification and reduction in the installation effort in a central office. Other portions of the system, such as the network and communication bus, are also connectorized.

A low cost, small, easily installed system with a highly standardized modular design has been successfully achieved with the No. 2 ESS hardware design.

## II. NEW APPARATUS AND EQUIPMENT—BASIC PACKAGES

The major new apparatus items in No. 2 ESS are: (*i*) thin film transistor-resistor logic gates; (*ii*) the network shift register; (*iii*) a peripheral decoder; (*iv*) an ac bus connector; and (*v*) a single-card memory card writer. Sections V and VI discuss these items in detail.

Because of system organization, and to meet economic considerations associated with the small office objectives, No. 2 ESS uses new codes of program store and call store memories. However, these memories are fabricated with the same parts and techniques as in No. 1 ESS to save money and to minimize development effort.

## III. FRAMES

The system uses the No. 1 ESS framework; the frames are single sided, seven feet high with modular widths of one foot, one inch. Sheet metal uprights of one and one-fourth by five inches are placed on a one foot deep base to provide an 8½ inch depth for apparatus on the front and 3½ inch depth for wiring on the rear.

### 3.1 *Frame Equipment*

No. 2 ESS uses No. 1 ESS type apparatus for most of its functions. This apparatus includes ferreed switches for network switching, ferrod sensors for scanning, wire spring relays for trunk circuits, twistor and ferrite sheet memories, and plug-in circuit packs. Newly coded ap-

paratus is described in Section V. These components are arranged on 22 frames listed in Table I in a manner which makes each frame, as nearly as practicable, a completely functional building block free of options.

The master scanner, recorded announcement, power distributing, miscellaneous power, protector, miscellaneous, and the combined distributing frames are the same as in No. 1 ESS. The remaining 15

TABLE I—FRAMES, ABBREVIATIONS, AND LENGTHS

| Frame | Abbreviation | Length | | Number Required |
|---|---|---|---|---|
| | | Feet | Inches | |
| Automatic message accounting | AMA | 2 | 2 | 0 or 1 per office |
| Combined distributing | CDF* | 6 | 6 | 1 to 8 per office |
| Central processor | CP | 4 | 4 | 2 per office, includes 8192 words of call store and 512 CPD points |
| Junctor grouping | JG | 2 | 2 | 1 to 3 per office |
| Line-trunk switching | LTS | 3 | 3 | 1 to 4 per line-trunk network |
| Miscellaneous | M* | 2 | 2 | As required |
| Maintenance center | MC | 4 | 4 | 1 per control complex may include single card writer |
| Miscellaneous power | MP* | 2 | 2 | 1 per office |
| Master scanner | MS* | 2 | 2 | 1 minimum per office |
| Miscellaneous trunk | MT | 2 | 2 | As required |
| Network control junctor switching | NCJS | 6 | 6 | 1 per line-trunk network |
| Power distributing | PD* | 2 | 2 | 2 or 4 per office |
| Protector | PROT* | 6 | 6 | 1 to 5 per office |
| Program store | PS | 4 | 4 | 2 to 8 per office |
| 6.7 V, 200 A power plant | PWR | 2 | 2 | 2 per control complex |
| Recorded announcement | RA* | 2 | 2 | 1 per office |
| Ringing and tone power plant | RT | 4 | 4 | 1 or 2 per office |
| Supplementary central pulse distributor | SCPD | 2 | 2 | 0 to 8 per office, supplements CPD in CP |
| Supplementary call store | SCS | 2 | 2 | 0 or 2 per control complex supplements call store in CP |
| Supplementary ringing & tone | SRT | 2 | 2 | As required |
| Trunk test | TT | 2 | 2 | 1 per office |
| Universal trunk and junctor | UTJ | 6 | 6 | As required |

* Identical No. 1 ESS equipment.

frames are designed to reflect the system objective of providing a low cost small electronic central office.

### 3.2 *Control Panel*

Control panels are provided on most of the No. 2 ESS equipment frames. A typical control panel contains a group of pushbuttons, indicator lamps, and test jacks. Power can be disconnected from or restored to various sections of the frame by operation of the keys. The keys are mechanically interlocked to prevent concurrent removal of power from duplicated units. The control panel also has a remote execute switch which will permit a maintenance man to start a maintenance program previously enabled at the maintenance center. Telephone jacks on the control panel permit convenient voice connections to other locations within the central office.

## IV. OFFICE ARRANGEMENTS

### 4.1 *Floor Plans*

Standard frame arrangements in an office minimize engineering and installation costs. A universal floor plan has been developed which grows naturally from the smallest to the largest installation. The pattern applied to a typical office is shown in Fig. 1. Some important features of this pattern are:

(*i*) The control complex frames, that is, maintenance frame, central processors, stores, 6-volt power frames, and trunk test frame have a fixed relationship in every office.

(*ii*) The frame lineups are so arranged that the office will grow approximately one lineup for every network added.

(*iii*) The protector and combined distributing frames grow perpendicularly to the frame lineups. These frames are aligned with the associated network frames for orderly growth together in a way that automatically minimizes cable and combined distributing frame jumper lengths.

(*iv*) The floor plan fits standard building bays of new buildings and can be readily adapted to existing buildings.

### 4.2 *Cable Rack, End Guards, and Office Lighting*

The cable rack, end guards, and office lighting equipment are the same as in No. 1 ESS. The cable racks conceal and shield interframe cabling and are supported by the frames (see Fig. 2). Because the frames are low and the aisles are relatively free from overhead racks,

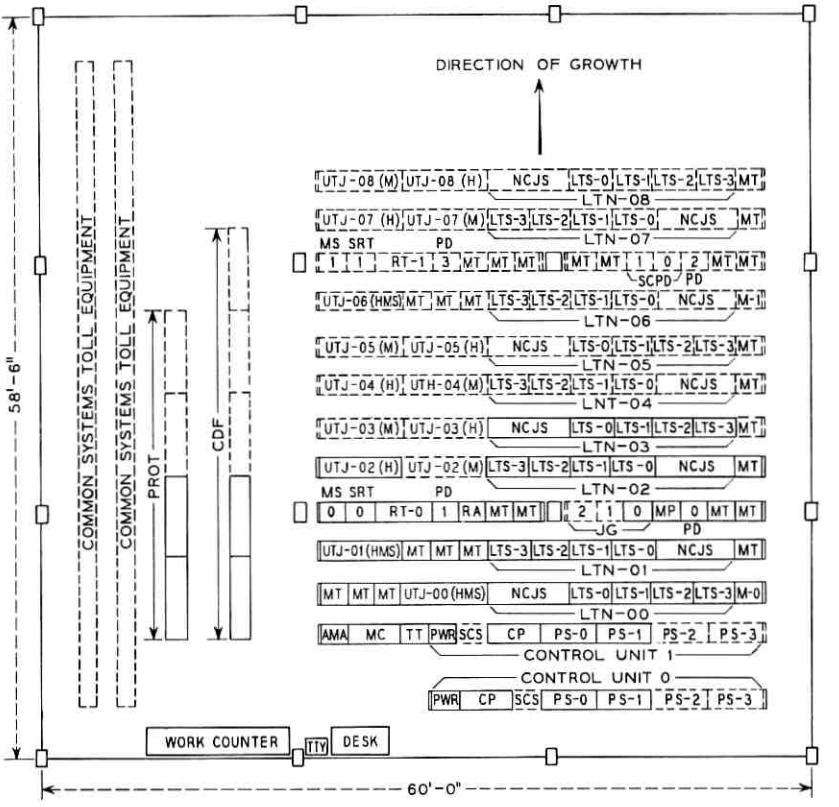



Fig. 1.—Typical No. 2 ESS office floor plan. Solid lines are four line-trunk networks with 4:1 concentration ratio for approximately 6500 lines and 850 trunks and service circuits. Dashed lines show how pattern continues with growth. See Table I for key to abbreviations.

excellent illumination is obtained with either the frame supported lighting or ceiling lighting fixtures. There are guards at ends of frame lineups, and at each exposed frame within a lineup. Aisle alarm lamps, aisle directories designating the frames in each lineup, and frame lighting control switches are in the end guards.

## V. APPARATUS

### 5.1 *Logic*

The No. 2 ESS control complex uses approximately 6,600 logic substrates (14,300 gates) and 300 binary counter substrates. Up to

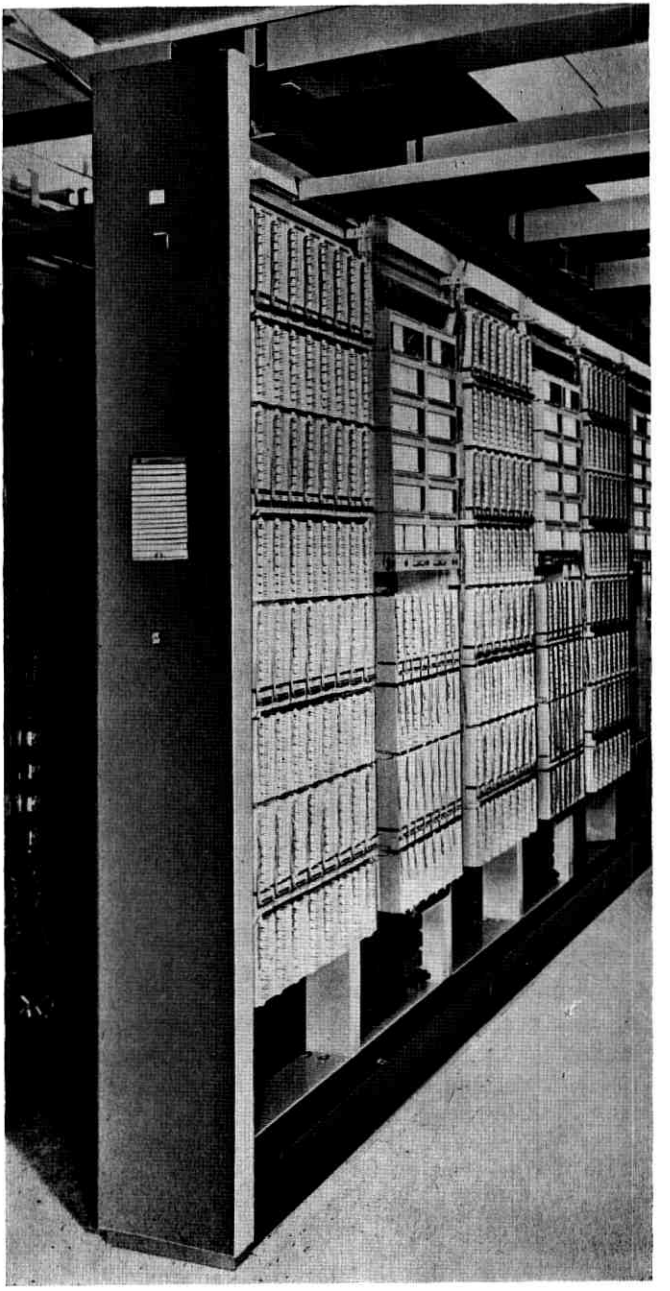


Fig. 2 — Typical equipment aisle.

five logic substrates or four binary counter substrates mount on a circuit pack.

The No. 2 ESS logic circuits are composed of high speed transistor resistor logic gates which are physically constructed from thin film resistors and beam lead transistors on an alumina substrate. Eighteen different codes make up the family of gate configurations consisting of two and four input NOR gates with fanout capabilities of three, ten, and forty-four. The number of circuits per substrate range from one to four depending upon complexity.

Three silicon npn devices are used in the gate circuits: a low current single transistor, a high current single transistor, and a double transistor. The alumina substrate is 0.025 inch thick by 0.55 inch wide by 1.825 inches long with seven terminals on each of the two long sides (Fig. 3). Tantalum nitride resistor films and titanium-gold conductor films are deposited on the underside of the alumina with the beam lead devices appliqued. The nickel and gold plated copper terminals are connected "inboard" from the edge of the substrate with sufficient space remaining for conductor paths to encircle the terminals. This design minimizes the number of crossovers to achieve the desired topology. Only three of the 18 codes require physical crossovers which are made with copper backstraps. Thermocompression bonding is used for all connections on the gate.

The binary counter circuit is compatible with the highspeed tran-

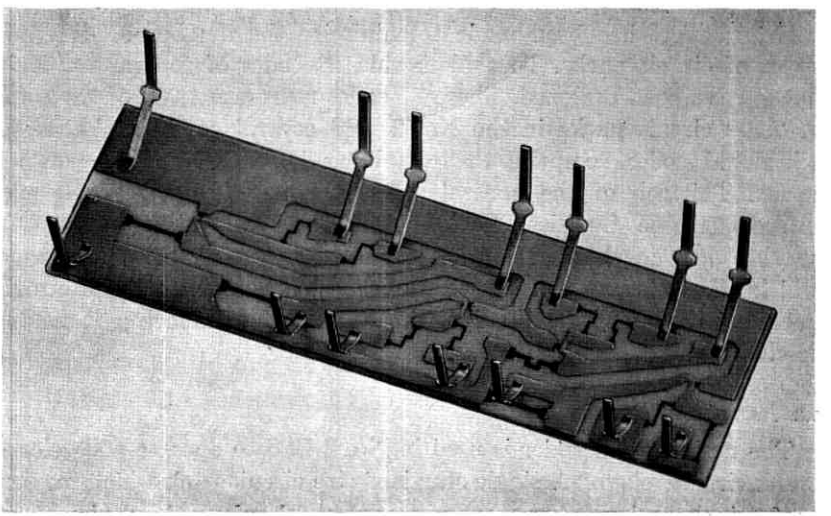


Fig. 3 — Transistor-resistor logic gate.

sistor resistor logic gate and consists of one bistable flip-flop with emitter follower outputs. A pulse steering network at the input enables the binary counter to be used as a toggle flip-flop or shift register stage. A thin film resistor network on a ceramic substrate forms the basic package. Circuit elements other than resistors are conventional leaded components with the leads inserted through holes in the ceramic and connected by soldering to the thin film circuit. Four discrete capacitors, two diodes, four transistors, and 16 thin film resistors are contained on a 0.025 inch by 1.0 inch by 2.0 inch substrate (Fig. 4).

### 5.2 *Circuit Packs*

The circuit pack is a plug-in subassembly consisting of a 3/32 inch phenolic laminate board with two-ounce copper paths formed through an etching process. These paths interconnect circuit components mounted on the board to form circuit configurations of various types. A plastic faceplate is riveted to the front edge of the circuit pack as a convenient place for code identification and to provide a slot for the tool used to remove circuit packs from the equipment. Twenty-eight terminals at the rear of the board mate with a connector mounted in the wiring field of the frame to provide an interface between the system and the circuit pack. The nominal size of the circuit pack is 3.75 inches high, 6.94 inches deep, with horizontal mounting centers in 0.20 inch multiples from 0.40 inch up through 2.00 inches (Fig. 5).

Every effort has been made to minimize the number of different codes of circuit packs. To achieve this, 21 codes of general use logic packs have been designed. These 21 codes represent 1,900 of the total of 2,700 circuit packs in the No. 2 ESS control complex. About 80 additional codes comprise the remaining logic and special purpose packs. Thirteen of the general use packs have been grouped into interchangeable families. Packs within a family differ from one another only in their output drive capacity. Thus, a circuit change which affects fanout can be accomplished by changing plug-in circuit packs without rewiring that position. Table II summarizes these interchangeable groupings.

### 5.3 *Peripheral Decoder*

The peripheral decoder is used in No. 2 ESS to operate and release trunk circuit relays.[3] A single standard size circuit pack which mounts on 0.4 inch horizontal centers will control 12 relays. A maximum of 64 of these circuit packs is required for one universal trunk frame.

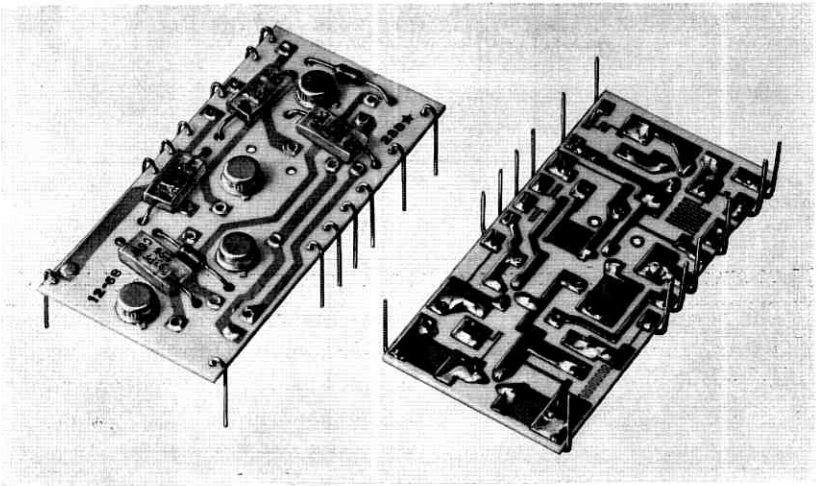


Fig. 4 — Binary counter.

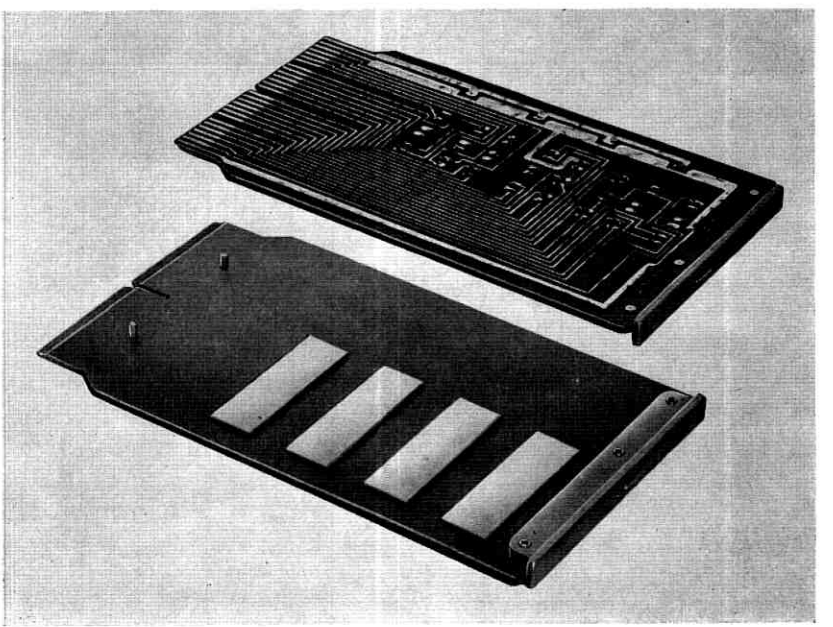


Fig. 5 — Logic circuit pack.

TABLE II—INTERCHANGEABLE CIRCUIT PACKS

| Fanout | 8 two input gates | 2 binary to 1/4 translators | 5 four input gates | 3 high-fan-in gates | 4 gates plus 4 bit register | 4 gates plus 2 bit register |
|---|---|---|---|---|---|---|
| Low | x | x | x | | | |
| Medium | x | x | x | x | x | x |
| High | | | x | x | x | x |

The peripheral decoder consists of a number of discrete components and two hybrid integrated circuits which together form a shift register and relay driver unit (Fig. 6). A number of discrete resistors, capaci-

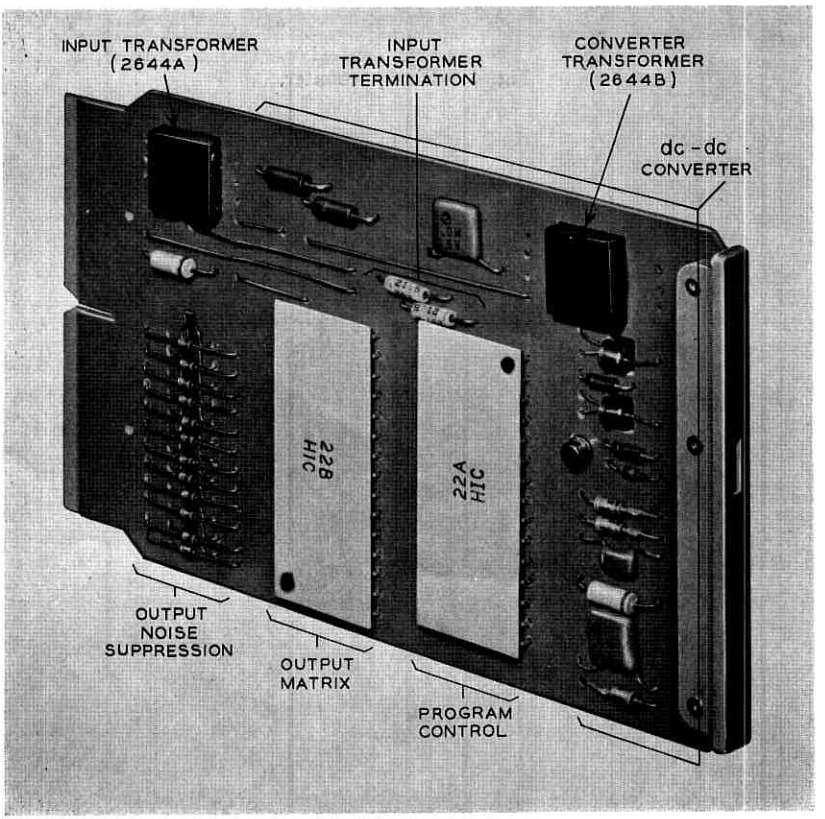



Fig. 6 — Peripheral decoder.

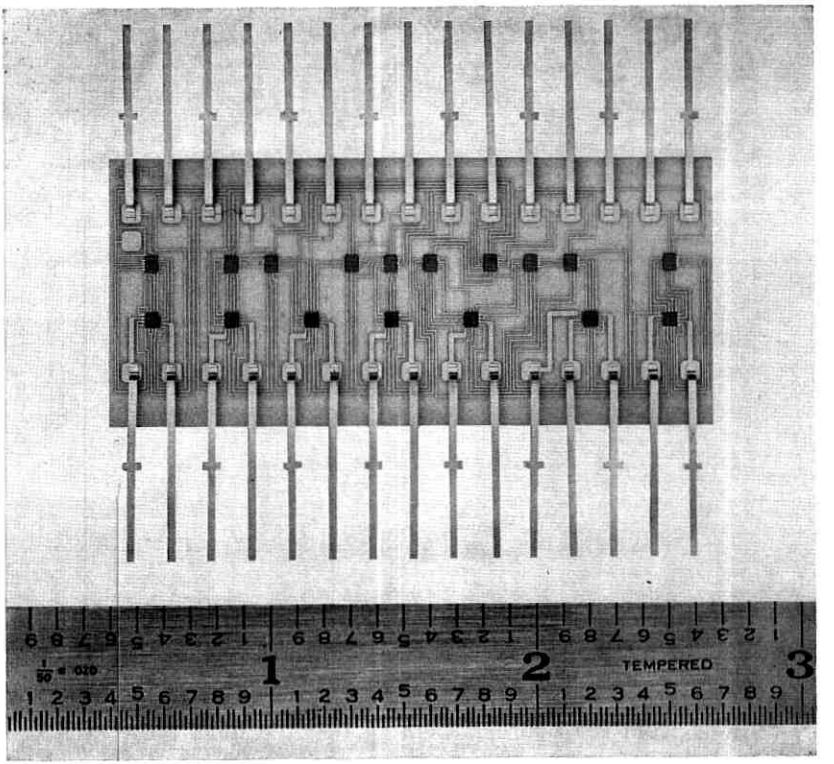



Fig. 7 — Hybrid integrated circuit.

tors, transistors, and a toroid transformer are used to convert 24 volts dc to 4 volts dc. A second transformer receives information from the central pulse distributor. This information is translated by the peripheral decoder and is used to control the trunk relays.

The substrates used for the two hybrid integrated circuits are 0.025 by 1.00 by 2.28 inches, high-alumina ceramics. Gold conductor paths five mils wide are deposited on the ceramic to provide circuit patterns. Thirty of these paths terminate at 75 mil square pads to provide bonding areas for external leads; 15 leads are thermocompression bonded on 0.150-inch centers on each long side of the substrate. The hybrid integrated circuits are mounted across the width of the board and 0.150 inch above it. This minimizes the possibility of broken ceramics from board warpage and rough handling. The separation of

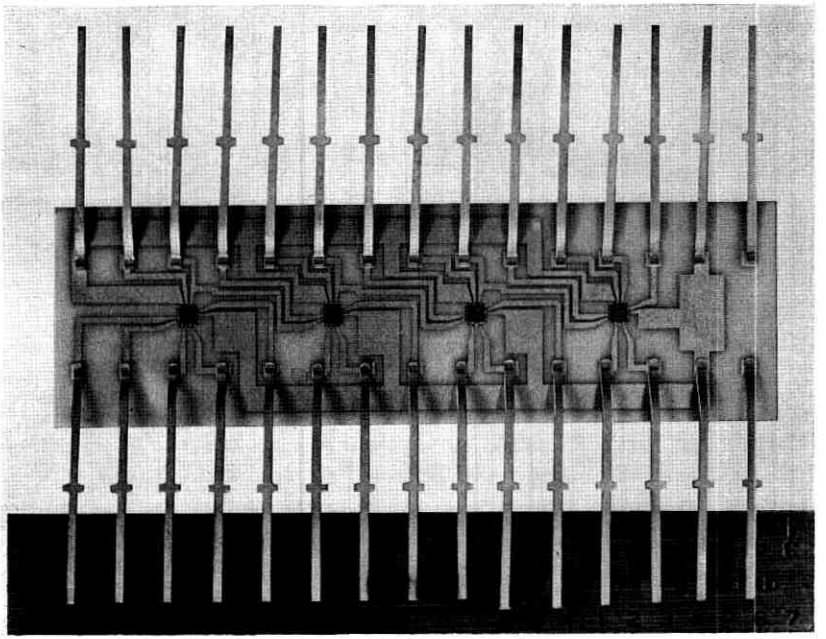


Fig. 8 — Four-bit register.

the ceramic from the board aids in cooling the units since at maximum load they dissipate as much as 0.5 watt each.

As shown in Fig. 7, seven devices on the lower edge of the ceramic form an 8-bit shift register with its associated gating circuitry. The remaining ten units are crossover chips which provide up to 25 "one-over-one" crossovers each. These chips will be replaced by air insulated crossovers in the future. All devices and crossover chips are thermocompression bonded to gold pads on the ceramic surface. A protective coating is applied to the device surface of the substrate. The complete hybrid integrated circuit is then mounted on the peripheral decoder circuit board in the same manner as other discrete components.

### 5.4 *Network Integrated Circuits*

The device shown in Fig. 8 contains four J-K flip-flops arranged to form a 4-bit register and is used on circuit packs in the network control junctor switch frame. The ceramic circuit, which is 0.025 by 0.68 by 2.28 inches, is fabricated in the same manner as similar devices used in the peripheral decoder.

### 5.5 *Bus Connector*

Transformer-coupled bus signaling is used between the central processor frame and the program store frames, and between the central processor frame and the network frames (see Section 7.2). As shown in Fig. 9, these connections plug in by means of newly coded connector assemblies.[4]

The balanced transformer-coupled bus loops through each frame and passes through a pick-off transformer for each required bit. Good noise and crosstalk characteristics are achieved by balanced grounding through center-tapped inductors in a physical package similar to the transformer. The inductor or transformer assembly mounts in the frame on a plate. The mating connector is arranged for attachment to the end of an interframe cable. The inductor or transformer assembly has a structure resembling a chest of drawers with a maximum of 12 trays assembled into a stamped metal housing. Eight male connecting terminals are provided on the front surface of each tray. Four of these connecting terminals mate with one connector while the four remaining terminals mate with the second connector. Thus, two cables plug in to each assembly.

## VI. EQUIPMENT FRAME DESIGNS

### 6.1 *Control Complex*

#### 6.1.1 *Central Processor Frame*

The central processor frame, Fig. 10, is part of the control complex which provides the primary control for operating a No. 2 ESS central

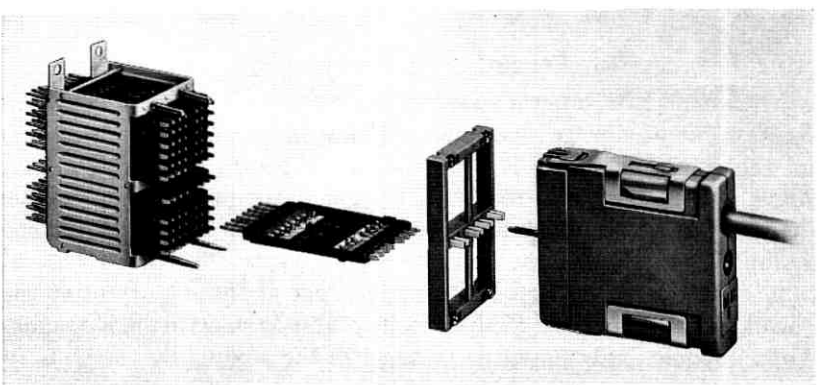


Fig. 9 — Bus connector.

office. The central processor uses instructions coded in adjacent memory frames to direct calls through the central office and to aid in detecting and analyzing improper performance of the equipment involved in this task. The processor is capable of addressing and reading a maximum of 262,144 words of semipermanent memory. This memory is located in program store frames providing up to 65,536 words of memory per frame.

Each central processor provides the logic to read and write into 32,768 words of temporary memory. This memory is provided by a complement of four call store units. The first two call store units mount in the processor frame while the remaining two optional units mount in a supplementary call store frame.

A local central pulse distributor matrix located in each processor frame provides for up to 512 enabling central pulse distributor points which may be equipped as required. One circuit pack provides a maximum of eight central pulse distributor points. Points on this matrix are also used to enable supplementary central pulse distributor frames. Each supplementary frame provides an additional 512 duplicated central pulse distributor points used for signaling purposes.

Communications with the peripheral equipment is by means of ac signals. These signals are carried over dedicated interframe address and answer buses which originate and terminate on the central processor frame. This plug-in access, provided by the bus connectors as well as the other control complex connectorization discussed in Section 7.2, allows the control complex to be thoroughly shop tested, disconnected, the frames shipped to the central office, and easily reconnected. This results in improved shop testing and a reduced installation interval.

#### 6.1.2 *Program Store Frame*

The No. 2 ESS program store frame shown in Fig. 11 provides 65,536 words of storage when fully equipped. The word size is 22 bits and the store cycle time is 6 microseconds. Four pluggable memory units allow for growth in 16,384 word steps to the maximum of 65,536. The pluggable memory unit (Fig. 12) installs from the front in one of the four dedicated frame locations and occupies 26 inches of vertical frame space. This unit can be installed either at the factory or at the central office. The heart of the memory unit is a permanent magnet twistor module. Information is retained in the module by the state of

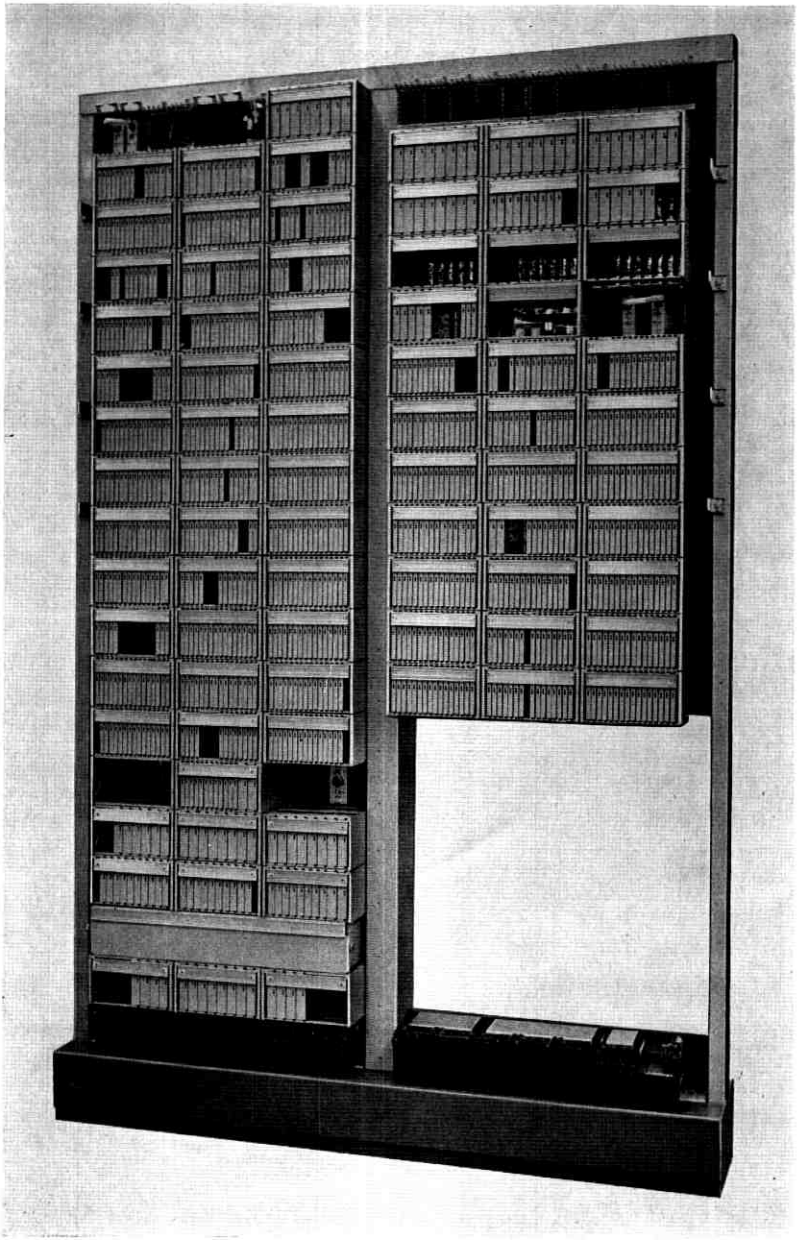


Fig. 10 — Central processor.

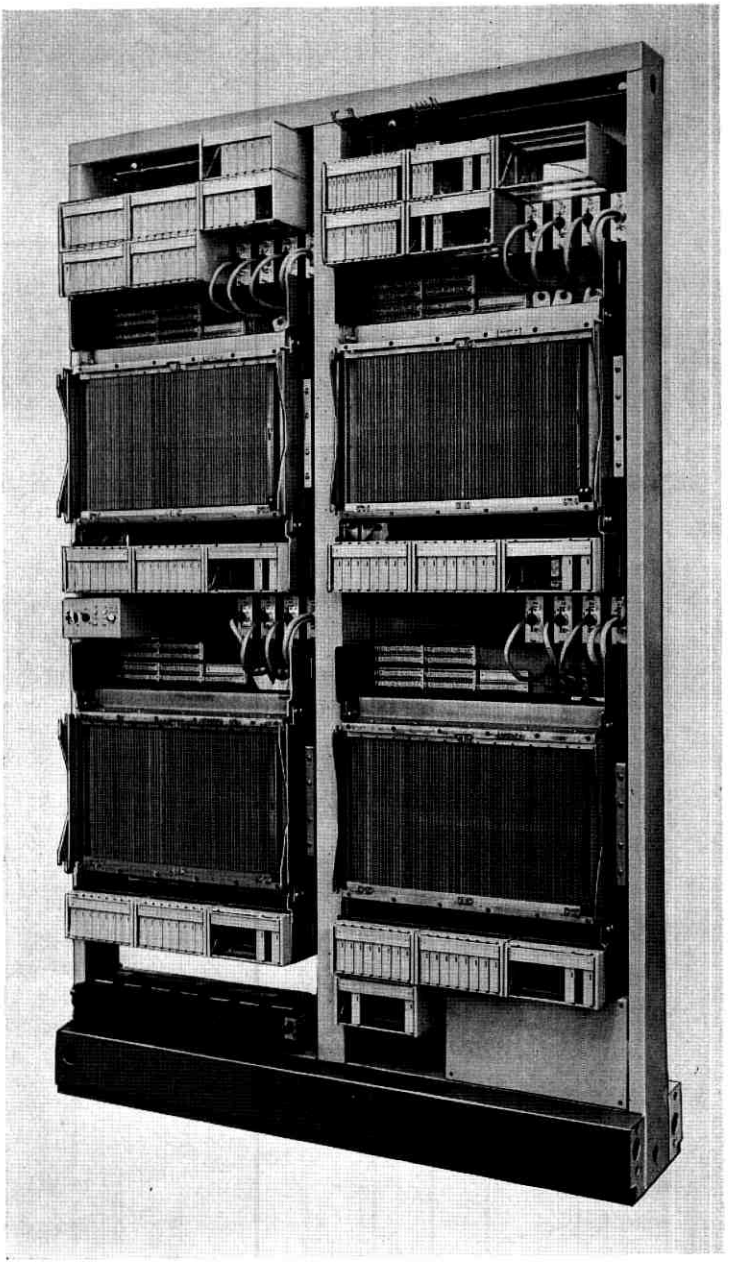


Fig. 11 — Program store.

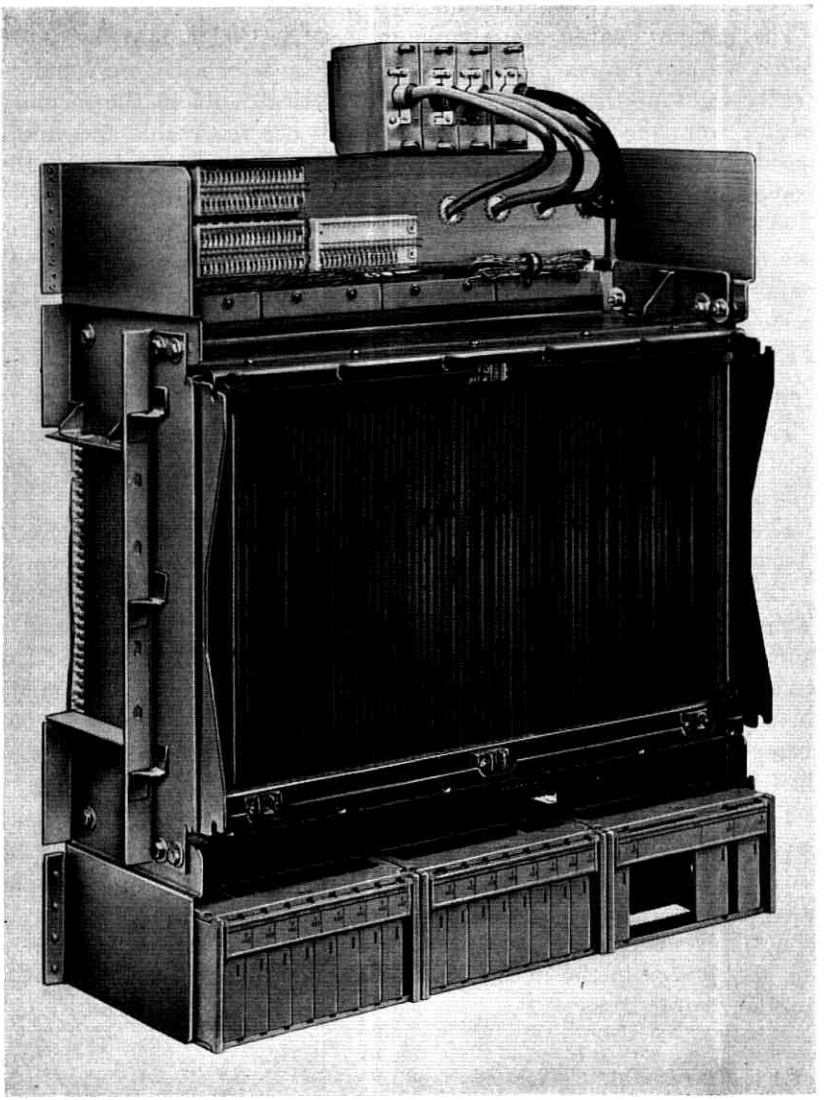


Fig. 12 — Program store unit.

magnetization of small permanent magnets. A magnetized magnet represents a binary "0" and a demagnetized magnet a binary "1".

The memory cards are aluminum sheets 6½ by 11¼ by 0.016 inches with a matrix of 64 by 44 magnets attached. One hundred twenty-eight of these cards slide into vertical slots in the memory module from the front. The memory cards are removable and information on them can be changed by removing the cards and using the memory card writer to change the magnetization of their magnets.

The common control for the memory module units is contained in two units each occupying 12 inches of vertical frame space. The remaining frame space contains miscellaneous units such as the fusing, connectors, and a test jack panel.

#### 6.1.3 *Call store*

The call store is used to hold rapidly changing information that must be available for interrogation and modification by both the program control and the input-output circuits.[5]

The call store is arranged in modular units each providing 8,192 16-bit words of storage in the fully equipped form and 4,096 words in the half equipped form. A 4,096 word call store unit consists of 74 circuit packs, a ferrite sheet plug-in module of 4,096 words, and a dummy module containing terminating resistors. To grow to 8,192 words, the dummy module is replaced by a second 4,096 word ferrite sheet module and several circuit packs are inserted in prewired connector positions. The 8,192 word call store unit as a whole connects to the frame through three patch cables.

The first 16,384 words are provided in two call store units equipped in the central processor frame. Two additional call store units may be provided in the supplementary call store frames to increase the temporary storage from 16,384 words to the maximum of 32,768 words. The supplementary call store frame is a growth unit and may or may not be required in the initial installation, depending on the office call storage requirements.

#### 6.1.4 *Maintenance Center*

The maintenance center frame (Fig. 13) provides the equipment which allows manual control of the system. Interrogations are made through a teletypewriter or the display panel controls. The answers are returned on a printout from the teletypewriter and visually on the display panel.

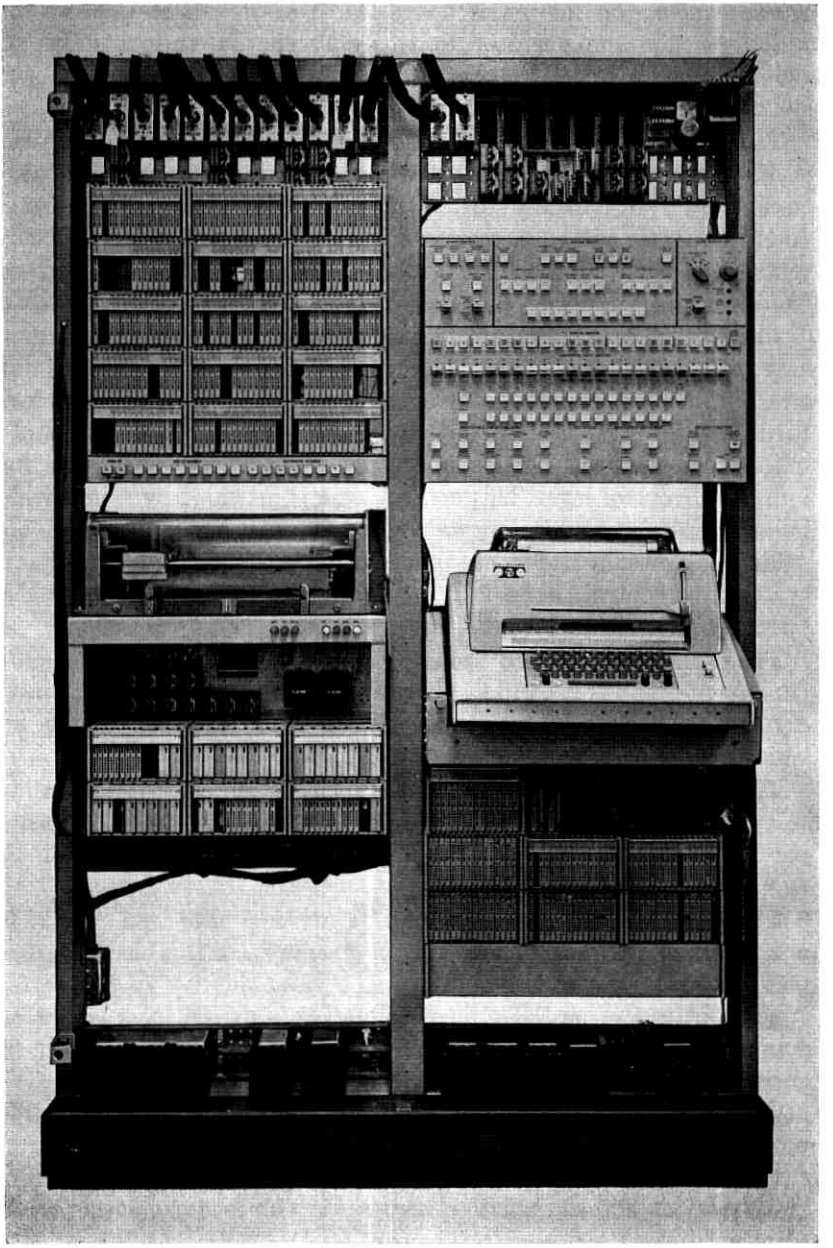


Fig. 13 — Maintenance center.

The controls on the maintenance frame are arranged in a manner which permits the operator to easily reach the various keys and switches which must be operated. Units not containing keys and switches are relegated to less convenient areas on the frame.

The maintenance center circuit provides the means for interrogation of the control complex. The maintenance center gathers information to be displayed visually to the operator. The control unit signals enter the frame via connectors and through buffer circuits their contents are shown on the display panel. The logic, which gathers the control unit information, is mounted on general purpose circuit packs.

The teletypewriter control circuit provides the logic to communicate with a maximum of eight teletypewriters directly or through data sets. The control equipment consists of a panel which contains keys to disconnect signals from the teletypewriters and data sets, and a unit which directs signals shared by all teletypewriters to their individual control logic. This control logic is provided by a unit which is arranged for two teletypewriters but equipped to control one. Additional circuit packs are added separately for the second teletypewriter control, when it is required. The control logic is able to operate either 33- or 35-type teletypewriters.

The optional single card writer (Fig. 14) is mounted in the maintenance center and is used to magnetize permanent magnet twistor memory cards. The card writer consists of two basic units: the control and logic circuit, and the mechanical card writing unit.

The control and logic unit contains circuit packs, controls, and a relay panel. Two connectors for the writing head are centrally located on the rear of the relay panel together with a connector cable for coupling to the card writing unit.

The card writing unit consists of a writing head and means for accurately moving the head over a memory card. The head drive uses two rods to precisely position the head relative to the surface which locates the card. The head mounting is driven by a lead screw coupled to a dc motor with a gear belt. A base casting provides a flat surface for the card and a rigid mounting for the head drive. The casting is fastened to the subframe at three points to minimize strains or distortions.

Once a magnet card is placed in the writer, the magnetizing information, after a teletypewriter request, is obtained from the system via a teletypewriter request. The card writing unit, driven at a speed of 10.3 inches a second, is used to write 11 bits simultaneously. Since

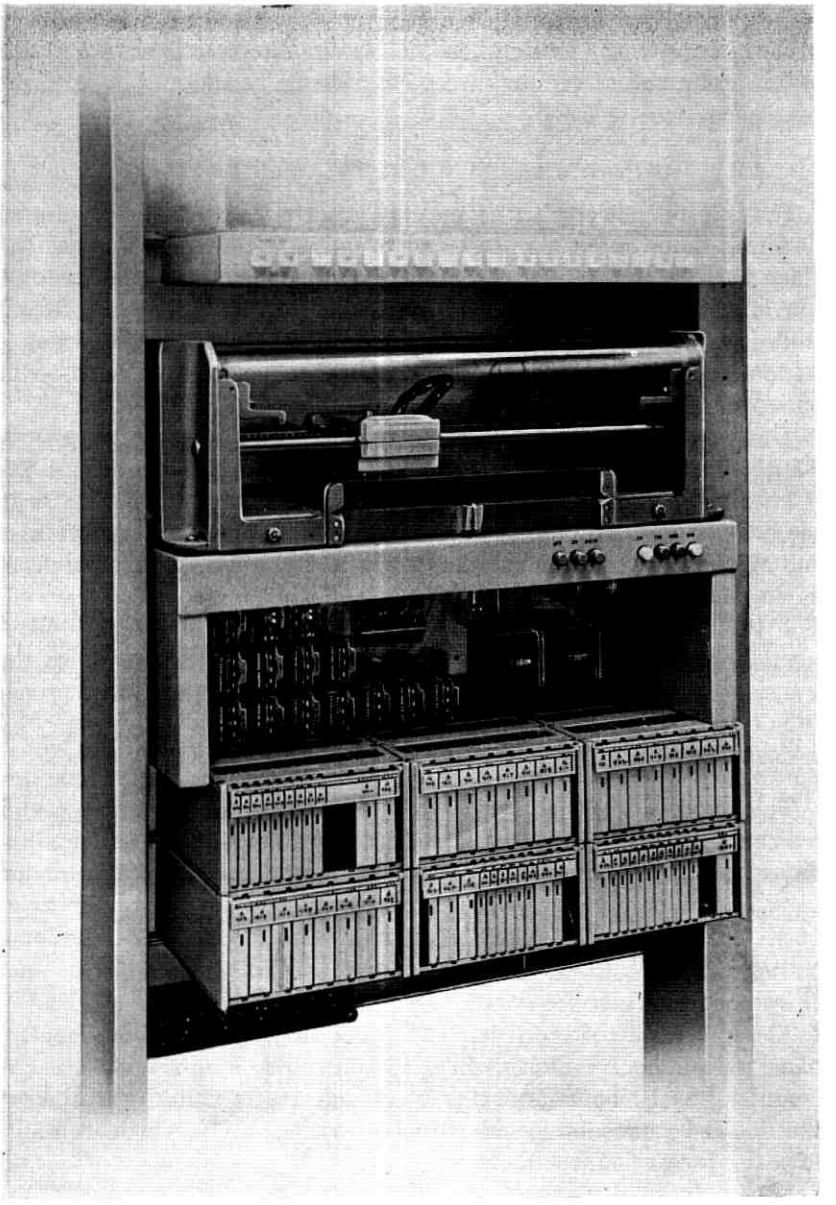


Fig. 14 — Single card writer.

the system can furnish 22 bits every 25 milliseconds, the information to be magnetized is stored in a register and on demand is gated in 11-bit groups to the writing head. The same circuitry is used for each 11-bit group and after four successive passes all 44 bits of every magnet row are written.

#### 6.1.5 6.7 *Volt Power Plant*

The 6.7 volt power plant is designed to provide +6.7 ±0.2 volts up to a maximum of 200 amperes and to operate from −48 volt central office battery. Each plant consists of two major parts: the dc-to-dc converter and plant control circuitry. The converter uses silicon controlled rectifiers to invert the −48 volt dc to high frequency ac which is stepped down, rectified, filtered, and appears as regulated 6.7-volt power on output buses in the plant.

The control circuitry, consisting of contactors, wire spring relays, distribution fuses, and timing circuits, connects and removes the +6.7, +24, and −48 volt loads located in the various equipment frames in the control unit. The power is removed in descending order of voltage and restored in ascending order to protect the semiconductors from damage caused by higher voltages being on while lower voltages are off. The total 6.7-volt load is split into approximately 25 ampere time-sequence steps for removal and restoration in order to minimize the transient effects and ground noise which could interfere with proper circuit operation.

#### 6.1.6 *Trunk Test Frame*

The trunk test frame is located between the maintenance center frame and the 6.7 volt power frame. This equipment permits maintenance personnel to make a variety of operational and transmission tests on trunks and service circuits. Leakage and continuity checks of both trunks and lines can be made although complete functional testing of customer lines is directed from a No. 3 local test cabinet elsewhere in the office.

The upper half of the trunk test frame, Fig. 15, contains a control panel which houses alarm, display, control, and test apparatus. A modular panel-mounted telephone set and access trunk control keys are located immediately above the writing shelf which contains facilities for storing trouble record cards.

Above the control panel provision is made for the voltmeter test panel and five optional test units: (*i*) transmission measuring set, (*ii*) noise measuring set, (*iii*) impulse counter, (*iv*) peak-to-average

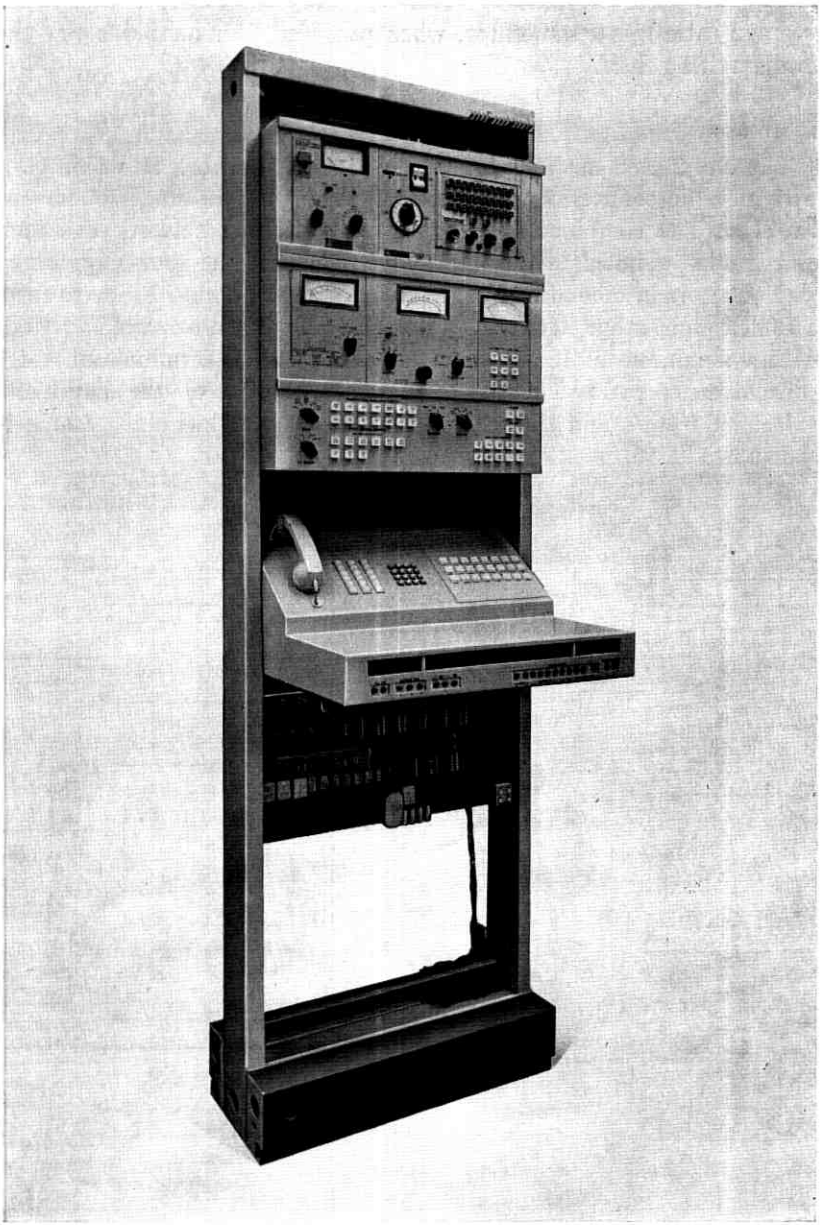


Fig. 15 — Trunk test frame.

ratio meter receiver, and ($v$) voice frequency oscillator. The peak-to-average ratio meter generator, when provided, is mounted below the frame writing shelf.

### 6.2 *Network*

The No. 2 ESS network is a space division network in which two-wire metallic connections are switched through eight stages of ferreed switches. The combined line and trunk switching network has lines and trunks assigned to terminals at one side of the four-stage array with junctors interconnecting the switches on the opposite side to form a folded eight-stage network. No. 2 ESS offices are arranged to work with a maximum of 15 networks providing 30,720 terminals at a 4:1 concentration ratio. The interconnection pattern of the line-trunk switching frames and the network control-junctor switching frame is shown in Fig. 16.

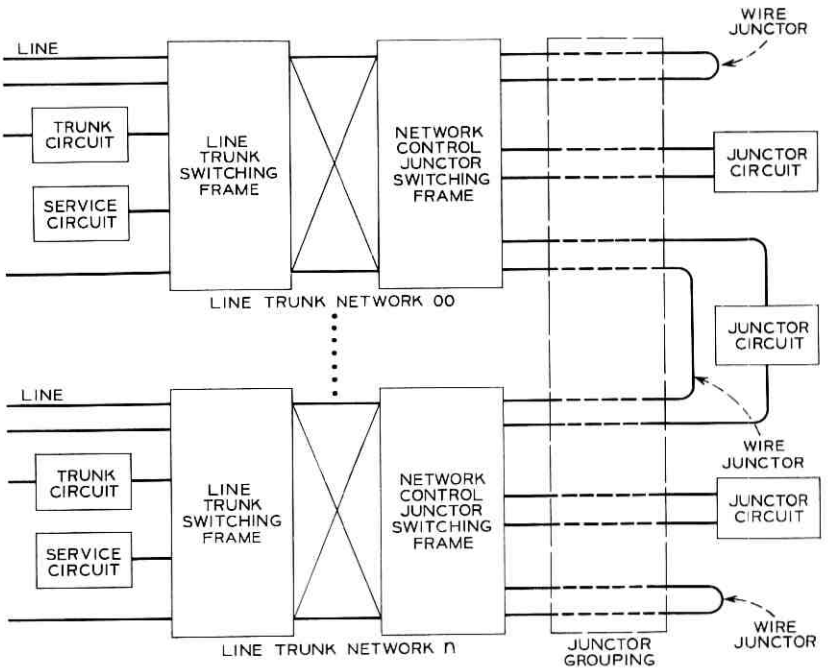



Fig. 16 — Network block diagram.

6.2.1 *Line-Trunk Switching Frame*

The basic line-trunk switching frame, Fig. 17, provides two 256 terminal concentrator groups with two stages of switching. Two of these frames, together with a network control junctor switching frame, provide a four-stage 1,024 terminal network having a 2:1 concentration ratio. Each concentrator contains its own scanner module, two stages of ferreed switches plus the cutoff switches, switch control relays, and connectors for B links and network control paths. The first stage switches have 4 by 4 crosspoint arrays and the second stage switches have 8 by 4 arrays to provide the 2:1 concentration.

Line-trunk switching frames 2 and 3 of a network are equipped with B link umbilical cords which permit these frames to be multipled to the B link connectors of frames 0 and 1. Thus, a network is provided having 2,048 terminals with a 4:1 concentration ratio.

Connectors in each concentrator group receive mating plugs for scanner control and readout, network control, ferreed switch pulsing paths, and diagnostic circuits. Scanner readout is via two series paths, each through one to four concentrator groups in a network.

The minimum growth increment for a line-trunk switching network is a frame containing 512 terminals. Testing and installation intervals are minimized through the umbilical cord and connector system incorporated in this design.

6.2.2 *Network Control Junctor Switching Frame*

Duplicated network and scanner controllers contained in the network control junctor switching frame, Fig. 18, serve from one to four line-trunk switching frames. One network controller normally serves the switches on any one of the eight concentrator groups and any one of the eight grids. At the same time, the other controller may be carrying out an order in one of the other equipped concentrator groups and in one of the seven remaining grids. Under no condition is it permissible to address a controller in such a manner that it will try to carry out an order in a concentrator group or grid being used by the other controller. However, each controller is capable of taking over complete control of the network if a failure occurs in its mate.

The scanner controllers detect service requests via the 512 ferrod sensors located on each line-trunk switching fame. One scanner readout cable plugs into a connector on concentrator group 0 and the other into concentrator group 4. Patch cords are used to extend these

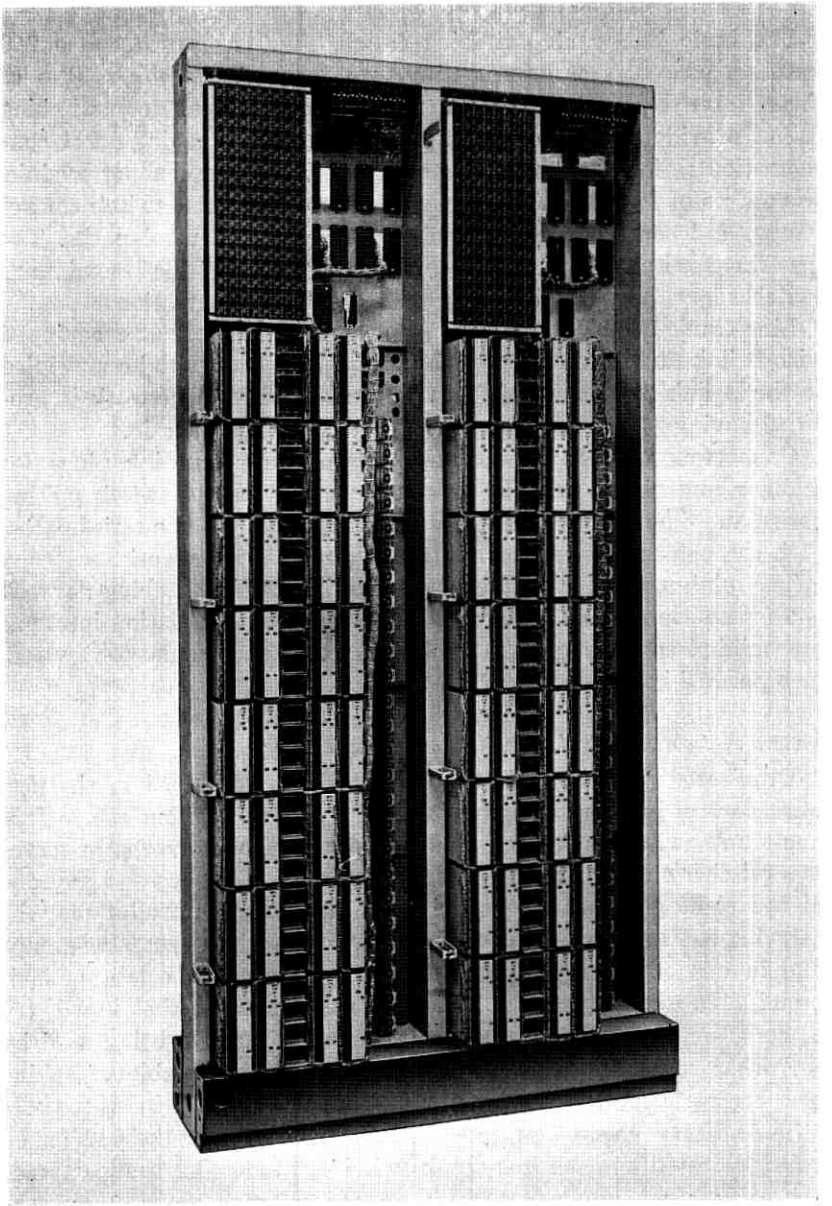


Fig. 17 — Line-trunk switching frame.

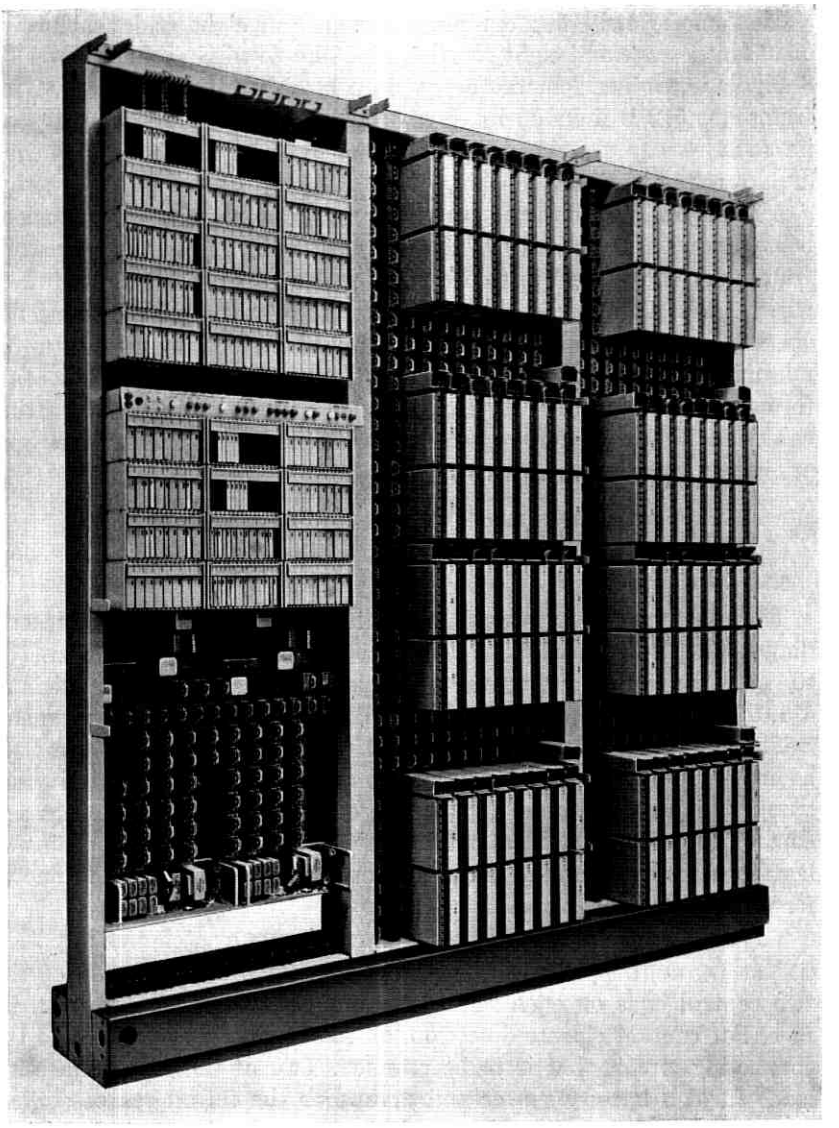


Fig. 18 — Network control junctor switching frame.

series loops for ferrod sensor state detection from concentrator groups 0 and 4 to groups 1, 2, 3, and 5, 6, 7, respectively.

The junctor switching equipment consists of eight grids, each with eight third-stage and eight fourth-stage 8 by 8 switches and eight 1 by 8 bipolar switches for test access into established connections. This frame has 512 B links on its third-stage switches and 512 junctors on its fourth-stage switches. Multipling of B links has been organized in patterns to minimize blocking.

All network access to the peripheral bus system is by connectorized bus transformers located in the network control bay. 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 has to be exercised in locating apparatus to minimize lead length and in separating leads into several local cable forms to minimize interference. The use of intranetwork connecting cables facilitates factory testing of a complete network complex and minimizes installation intervals.

### 6.3 *Peripheral Equipment*

#### 6.3.1 *Master Scanner*

The master scanner is used to monitor various administrative and diagnostic points throughout the system. This scanner, like those on the network frames and universal trunk and junctor frame, consists of a 1,024 point ferrod sensor matrix and duplicated control equipment.

The master scanner, shown in Fig. 19, provides a 1,024 point matrix as contrasted with the 512 point optional matrix unit in the universal trunk and junctor frame, Fig. 22. In order to provide the necessary control point duplication at minimum expense, each office contains at least one master scanner frame and one universal trunk and junctor frame equipped with a master scanner.

To provide for detecting either contact closure or 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 face. For contact closure detection, all four contacts are cabled to the connecting circuit that is to be scanned. Two of the four connecting leads furnish battery power and ground to the ferrod sensor control windings so that a contact closure over the second pair may be observed. For potential change detection, these coils are strapped together, series aiding, and are connected to the circuit under surveillance with a single pair of wires.

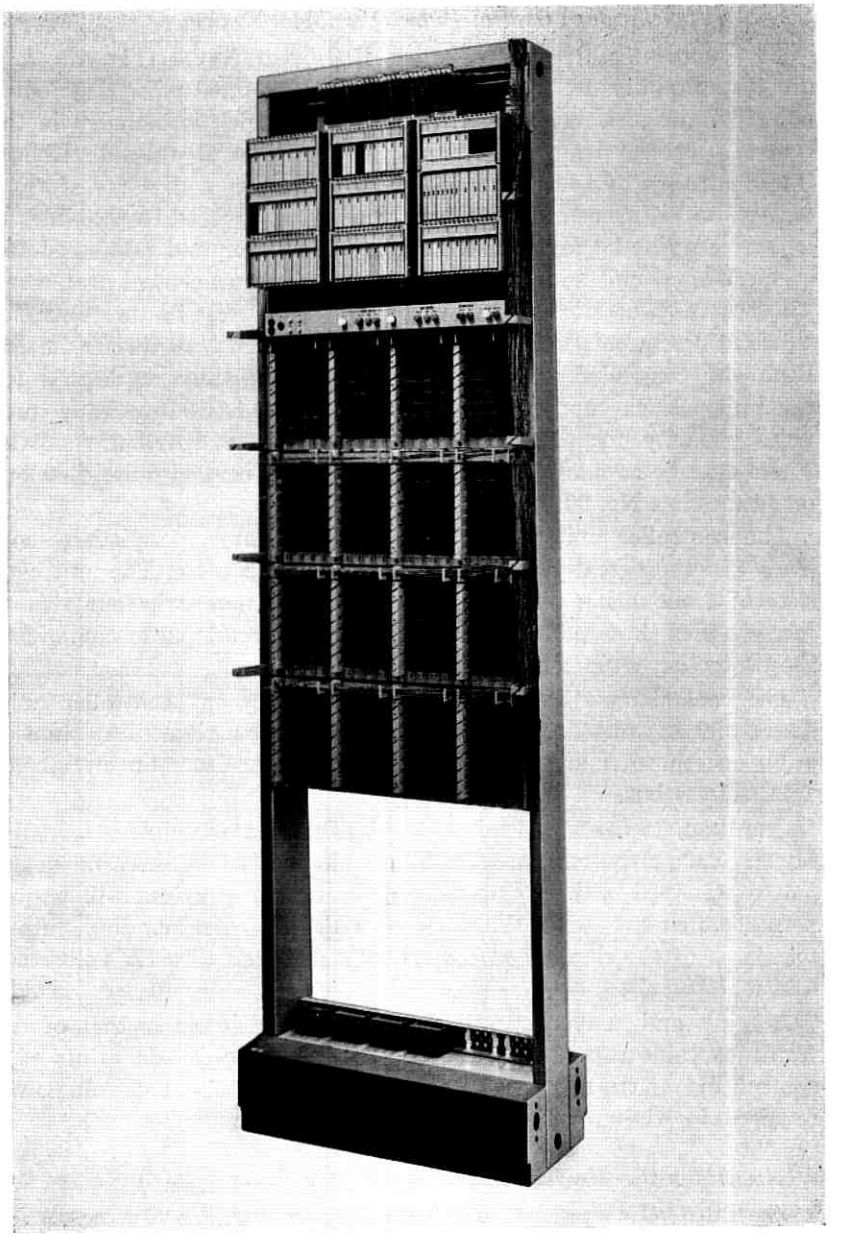


Fig. 19 — Master scanner frame.

6.3.2 *Supplementary Central Pulse Distributor Frame*

The supplementary central pulse distributor frame (Fig. 20) is a growth unit intended to supplement the 512 central pulse distributor points of the local central pulse distributor which is included in the central processor frame. The supplementary central pulse distributor frame consists of two major parts, a controller and a transformer matrix, and is used primarily for signaling as opposed to enabling. It is enabled via the local central pulse distributor and addressed via the peripheral unit address bus by either of the two central processors. The central processors signal the central pulse distributor controller which in turn accesses the central pulse distributor transformer matrix. Both the controller and the matrix are completely duplicated for reliability. Each supplementary central pulse distributor frame provides capacity for 512 duplicated central pulse distributor points. There may be as many as eight supplementary central pulse distributor frames in a No. 2 ESS office.

The controller unit is at the top of the frame, permitting short leads and thus greatly reduces their exposure to noise. The unit consists of a mounting plate equipped with terminal strips and transformers plus three plates equipped with the circuit packs comprising the controller equipment.

The control panel is mounted directly below the controller unit. The panel keys control frame power for the supplementary central pulse distributor and permit isolating the frame from the peripheral unit address bus.

One supplementary central pulse distributor matrix unit is mounted directly below the control unit. This unit consists of three mounting plates equipped with circuit packs. The matrix provides 512 output points which are wired in parallel with the corresponding output terminals of the duplicated matrix unit mounted directly below the first matrix unit.

A relay and converter unit mounted in the lower section of the frame contains two + 24-volt to + 6-volt dc-to-dc converters plus control and alarm relays. The converters furnish the + 6-volt power required for the circuit packs mounted in the controller unit.

6.3.3 *Automatic Message Accounting Frame*

The automatic message accounting frame provides the means for recording customer billing data on a nine-track magnetic tape. The tape is written as data are collected by the system. This tape is sub-

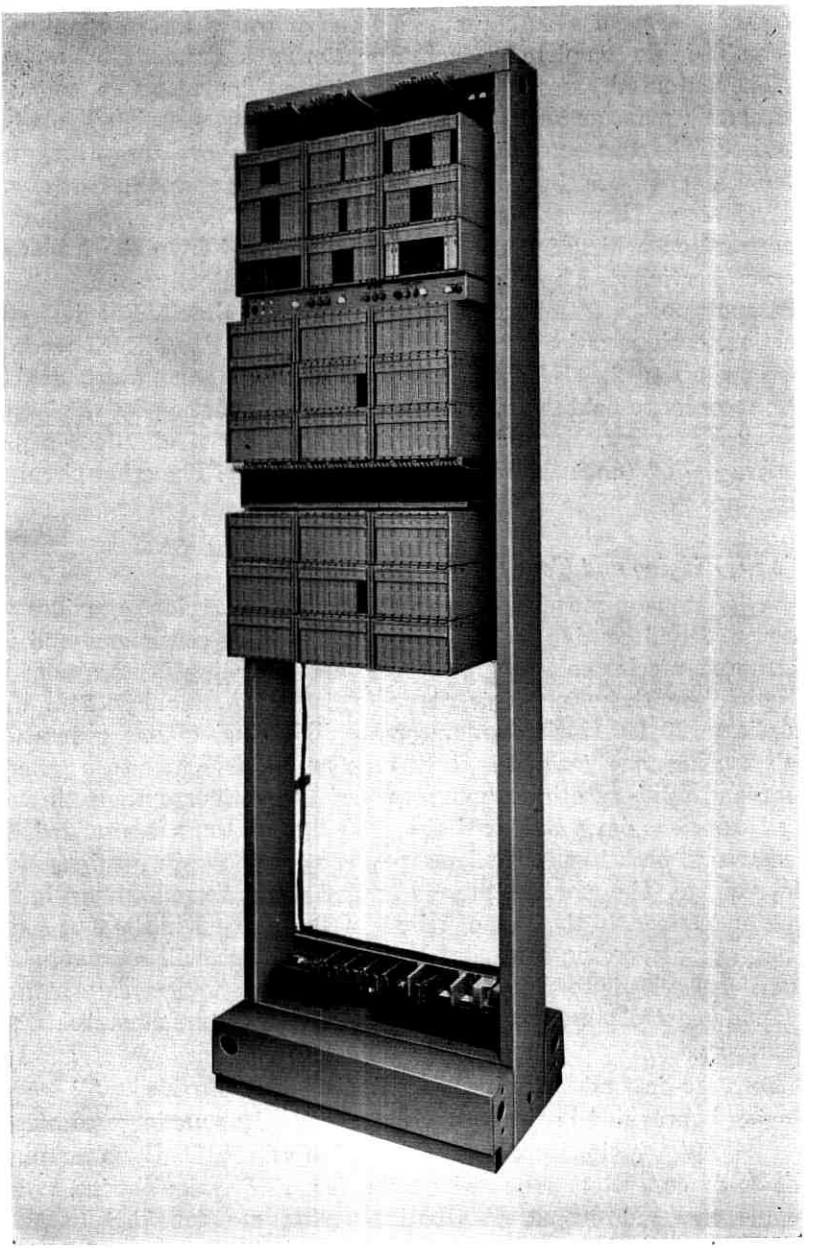


Fig. 20 — Supplementary central pulse distributor frame.

sequently sent to a data processing center where the information is assembled into complete billing information.

The automatic message accounting frame includes two complete automatic message accounting circuits and their associated tape recorders. The two recorders are mounted one above the other, near the center of the frame with the control logic and other circuit packs above them at the top of the frame. The tape transport is mounted on a hinged plate and can be removed from the frame for major servicing. Three connector cables electrically connect each transport to the frame-mounted portion of the circuit. For dust protection, each recorder has a hinged transparent door. Each recorder accepts up to 2,400 feet of half inch magnetic tape which enables one recorder to hold several days of billing information for the largest No. 2 ESS office. Normally, one circuit handles all data for a prolonged period (until the tape reel is full of data). The other circuit is for standby.

#### 6.3.4 *Ringing and Tone Frame*

A ½ ampere capacity ringing and tone power plant provides interrupted and continuous 20 Hz ringing current, continuous and interrupted precise call progress tones, and signaling interruptions as required by the office. This plant (called 841A, see Fig. 21), uses duplicate 20 Hz ringing generators, audible ringing tone generators, call waiting tone generators, high tone generators, busy tone generators, and solid-state interrupters to feed the various outputs through load transfer relays. Balanced distribution of all tones is provided. All generators, both regular and reserve, are continuously monitored for low voltage. The monitor outputs are fed to the ferrod sensors in the master scanner. In the case of failure of any element in the 0 or 1 side of the plant feeding the office loads, the system will automatically transfer the loads to the other side and provide the necessary alarms. Manual controls are provided to override automatic control when necessary.

Both ac-dc and superimposed ringing are provided. All coded ringing is provided in the connecting circuits. In superimposed offices, the + 48V tripping supply is obtained from the 610D power plant (dc-to-dc converter) mounted on the frame. Terminal strips, power connectors, and output distribution fusing are provided in these plants for connection to associated office equipment.

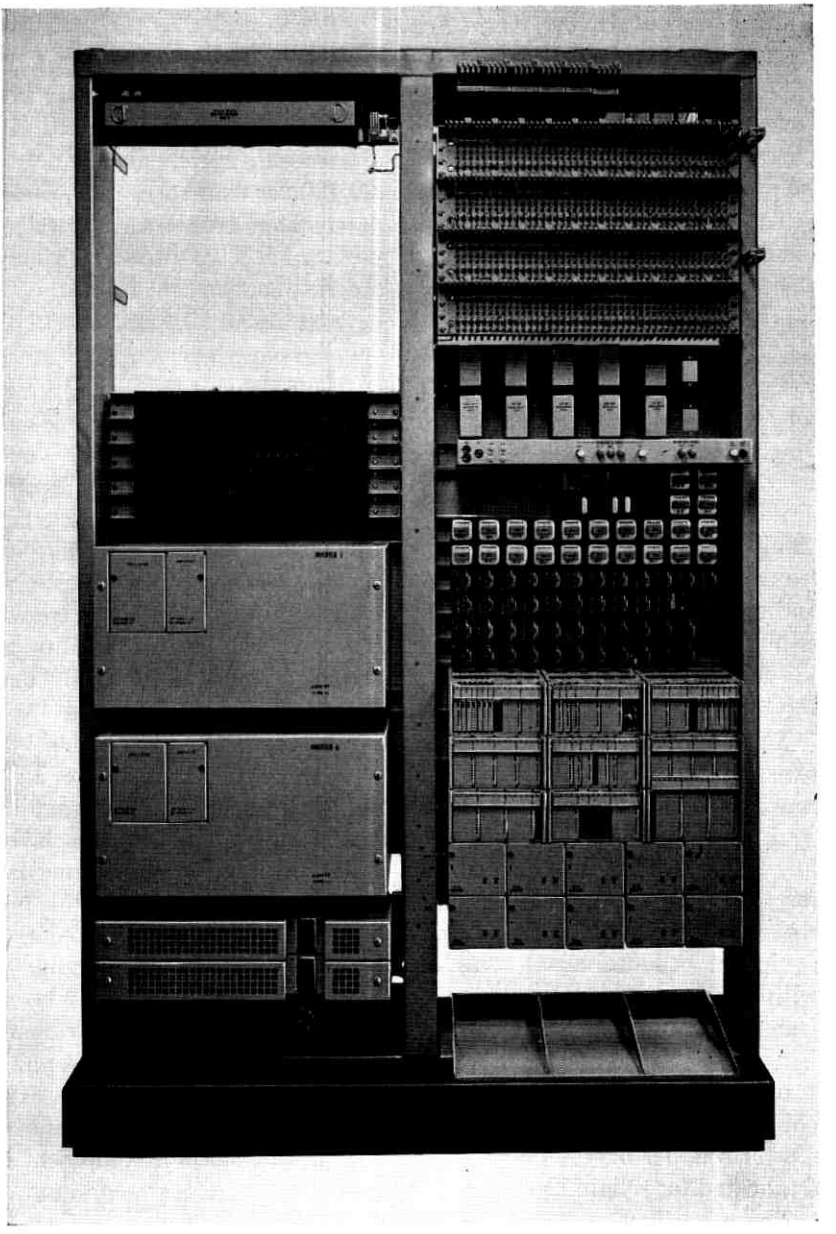


Fig. 21 — Ringing and tone frame.

6.3.5 *Recorded Announcement Frame*

The recorded announcement frame provides for a maximum of six announcements on a small magnetic drum recorder. 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.

6.3.6 *Trunk Frames*

There are two types of trunk frames, the universal trunk and junctor frame, and the miscellaneous trunk frame. The universal trunk and junctor frame accommodates a maximum of 256 trunk circuits and their associated trunk control scanners and peripheral decoders. It also provides space for an optional 512 point scanner for miscellaneous trunk circuits. The miscellaneous trunk frames contain a variety of trunk and service circuits which do not fit the universal trunk size and control point requirements. The scanner function for these trunks is performed by a master scanner. The peripheral decoder points on the miscellaneous trunk frame are assigned as needed for trunk and service units located on each specific frame.

The universal trunk and junctor frame (Fig. 22), as its name implies, is universally wired so that any universal trunk or junctor unit consisting of four circuits arranged on a 2-inch mounting plate may be equipped in each of the 64 unit positions in bays 0 and 2 of the three-bay frame. These bays also contain the scanner ferrod sensors needed for trunk circuit supervision. The center bay contains the trunk peripheral decoder and scanner control equipment. To save on control equipment, the universal trunk and junctor frames may be equipped with a 1024 point scanner control. If so equipped, it is called a home frame, or if not equipped with a scanner control, it is called a mate frame. The home frame operates either one-half of a 1,024-point scanner matrix on each of the home and mate frames or a 512-point scanner matrix on the home frame and the optional 512-point master scanner on the home frame. The scanner control equipment is duplicated for reliability.

The miscellaneous trunk frames, Fig. 23, contain such a variety of trunk and service circuits that it is uneconomical to provide them with universal scanners to satisfy all conditions. Instead the scanning

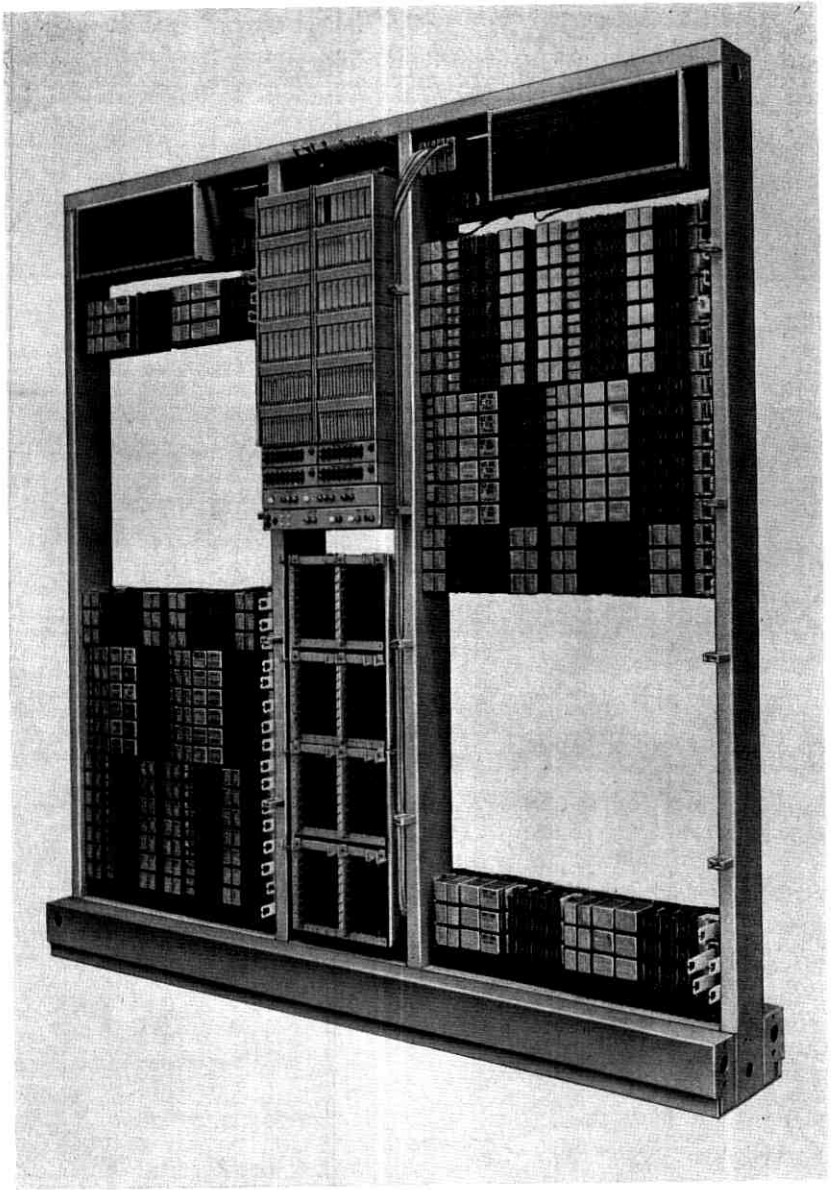


Fig. 22 — Universal trunk and junctor frame.

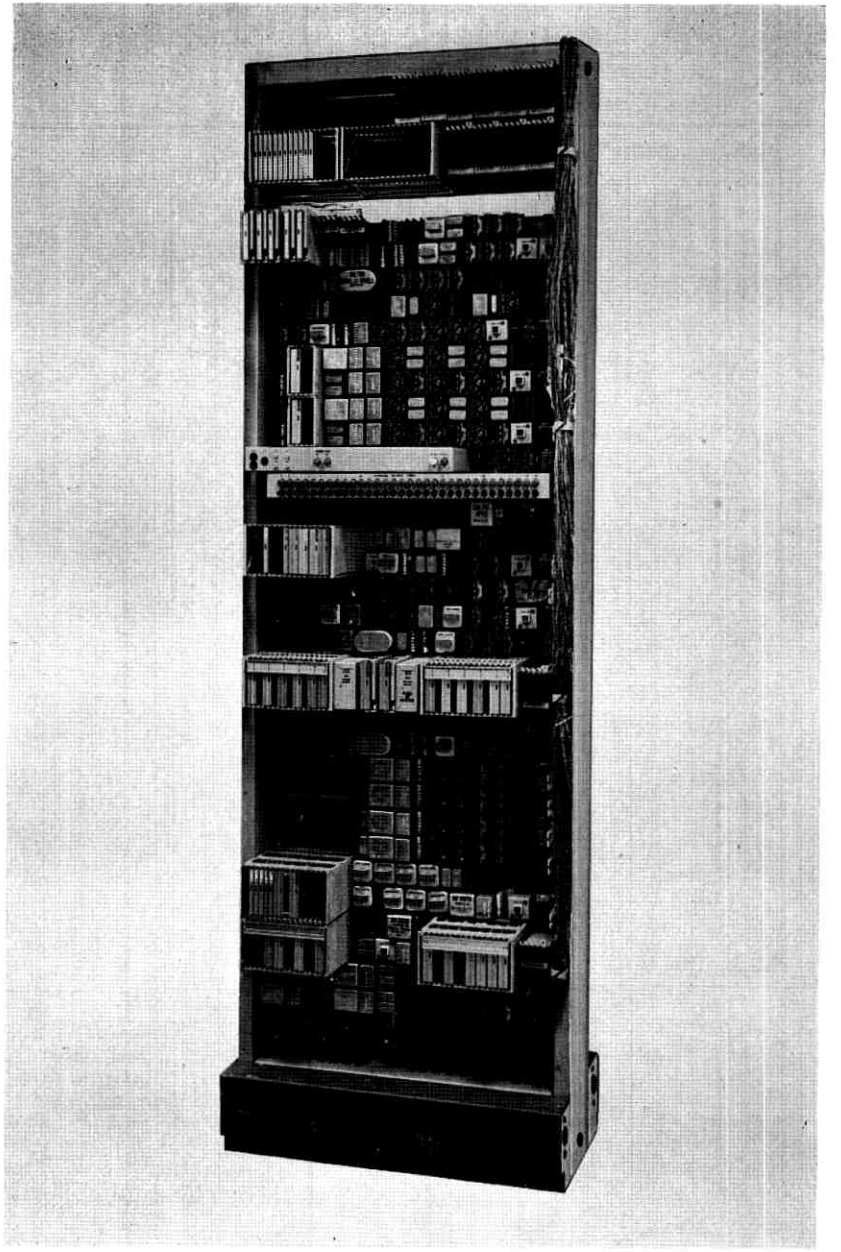


Fig. 23 — Miscellaneous trunk frame.

function for these trunks is performed by the scanner either on a universal trunk and junctor (home) frame or on a separate 1,024-point master scanner frame.

Connectorized bus transformers are provided at the top of the universal trunk and junctor frame for access to the peripheral bus.

6.3.6.1 *Universal Trunk and Junctor Units.* Since many trunk and junctor functions are performed by the common control equipment such as the central processor scanners and peripheral decoders, the size and complexity of these trunk and junctor circuits are greatly reduced. Most high-runner incoming and outgoing trunks are simple circuits containing two or three relays. This simplicity and uniformity of control point requirements permitted the development of a family of single four-circuit units on two-inch mounting plates, as Fig. 24 shows. All units comply with an assigned terminal pattern to insure compatibility with the universal frame wiring. In a typical office, all junctor and 50 percent of all trunk units will be of this type.

6.3.6.2 *Miscellaneous Trunk and Service Units.* Those trunks, transmitters, receivers, 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. A typical unit of this type is the multifrequency receiver shown in Fig. 25. These units are located on the miscellaneous trunk frames and are cabled to their associated master scanner control points.

6.3.7 *Miscellaneous Frame and Miscellaneous Power Frame*

The miscellaneous frame is designed to accommodate a variety of units which require neither peripheral decoder nor scanner control. These units include emergency manual lines, a multiplicity of common systems units, the power for testing battery supply, and so

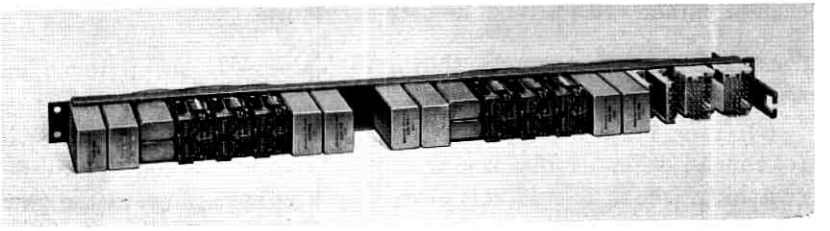


Fig. 24 — Universal trunk unit.

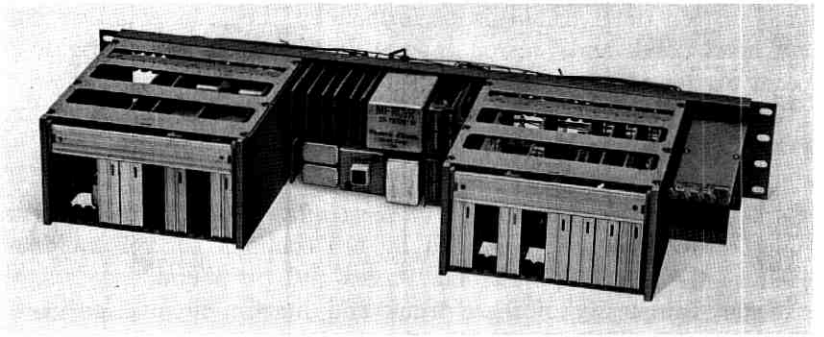


Fig. 25 — Multifrequency receiver.

forth. 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 a miscellaneous frame equipped with an ac distribution panel for 120-volt single phase loads requiring protected or essential 60-Hertz supply, the + 130-volt and − 130-volt fuse panels, and the floor alarm units.

6.3.8 *Power*

6.3.8.1 *Power Distributing Frame.* The power distributing frame is the battery load distributing point of the system. 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 microfarad capacitor banks near the bottom of the frame 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 for frame loads) from 5-, 15-, or 30-ampere cartridge fuses, each in parallel with a 1-1/3 ampere alarm fuse.

6.3.8.2 *Power Plants.* As shown in Table III, the power plants associated with No. 2 ESS include:

(*i*) Two 111A battery plants with 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.

(*ii*) Two 6.7 volt power plants supply the control complex logic. These are located in the switchroom and are described in Section 6.1.5.

TABLE III—POWER SUPPLIES

| Power Supply | Type of Plant or Unit | Capacity (Rated) Amperes | Code |
|---|---|---|---|
| **In Power Room** | | | |
| −48 volt dc (−43.75 to −52.5 V) +24 volts dc (+21.75 to +26.25 V) | storage batteries (without emergency cell or counter cell switching) rectifier charged | 10–800 | 111A |
| +130 volt dc<br>−130 volt dc | dc-to-dc solid-state conversion from −48 volts for coin control | $\frac{3}{4}$<br>2 and 5 | 610B<br>651A |
| Reserve ac supply | dc motor-driven alternator for 120–208 volt single-phase power | (1½ kW) | 504B |
| **In Switchroom** | | | |
| Ringing and tones | solid-state generator with a precise tone plant | 0.5 | 841A |
| 6 volt power plant | dc-to-dc converter with sequence controller | 200 | J86859A |
| PBX talking battery filter on miscellaneous frame | coil and capacitor panels | 15, 25, and 60 | |
| 120 volt ac for<br>Frames: maintenance center<br>RA<br>RT<br>Miscellaneous for ac TTY data sets, test battery supply unit | commercial power with or without reliable supply distributed from MP frames | | |
| Appliance outlets frame lighting | distributed from ceiling-supported busway | | |

(*iii*) The ringing and tone supplies located in the switchroom. These are described in Section 6.3.4.

(*iv*) The + 130 and − 130 volt dc-to-dc converters (610B power plants) which convert the − 48 volts to the potentials needed for coin control. Power from these plants is delivered to fuse panels on a miscellaneous frame and then distributed to all frames in the office which require it.

(*v*) A small emergency 504B alternating current plant (an alternator driven by a dc motor) providing "protected" power. A distribution panel on a miscellaneous frame provides a centralized point in the switchroom for all frames in the office which require ac power even when commercial power fails.

(*vi*) An engine alternator to substitute for commercial ac power to charge batteries and supply essential ac loads after a commercial power failure has persisted for a time.

### 6.4 *Distributing Frames*

The No. 2 ESS combines all of the functions of the main distributing frame, intermediate distributing frame, and trunk distributing frame in a combined distributing frame. As in No. 1 ESS, junctor grouping and protector frames are also provided.

#### 6.4.1 *Combined Distributing Frame*

The combined distributing frame provides for connection of 6,080 central office pairs (Fig. 26). Reliable quick jumper connections to connecting blocks are made with a special tool and 22 gauge W-type distributing frame wire.

The combined distributing frame faces the protector frame across a four-foot aisle and the two frames grow perpendicular to all other frame lineups. The combined distributing frame provides a means for interconnecting:

(*i*) lines terminated on the protector frame and network terminals,

(*ii*) trunks terminated on the protector frame and trunk circuits,

(*iii*) trunks terminated on signaling and transmission equipment and trunk circuits,

(*iv*) trunk circuits, service circuits, and network terminals, and

(*v*) lines or trunks terminated on the protector frame and miscellaneous circuits.

Service observing jack panels, which provide access for service

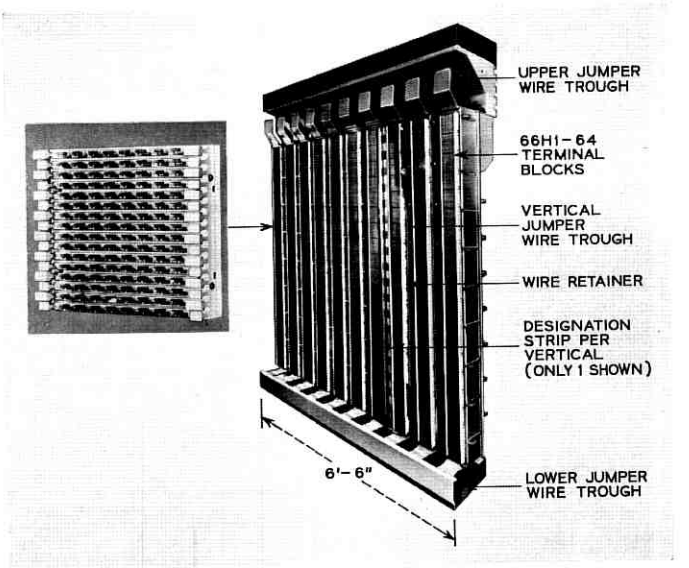



Fig. 26 — Combined distributing frame.

observing circuits for every line within the office, are mounted on the rear of the combined distributing frame as needed.

6.4.2 *Protector Frame*

The No. 2 ESS uses the No. 1 ESS protector frame (see Fig. 27), which has protectors for 6,000 outside plant pairs. These protector units which serve the tip and ring conductors guard against lightning and other foreign high potentials.

6.4.3 *Junctor Grouping Frame*

The junctor grouping frame, which is used for the distribution of network junctors, provides up to ten vertical files for network or junctor circuit appearances (see Fig. 28). Each vertical file contains 32 plugs and 32 jacks which may be used to interconnect either the junctors from one network (64 eight-pair junctor subgroups) or 32 junctor circuit subgroups.

The junctor grouping frame can grow in two-file increments as central office requirements dictate. One growth module consists of two 37-inch high vertical mounting plates containing two files of 16 plugs and 16 jacks each. Files may be added when necessary and any file may be used for either network junctor or junctor circuit

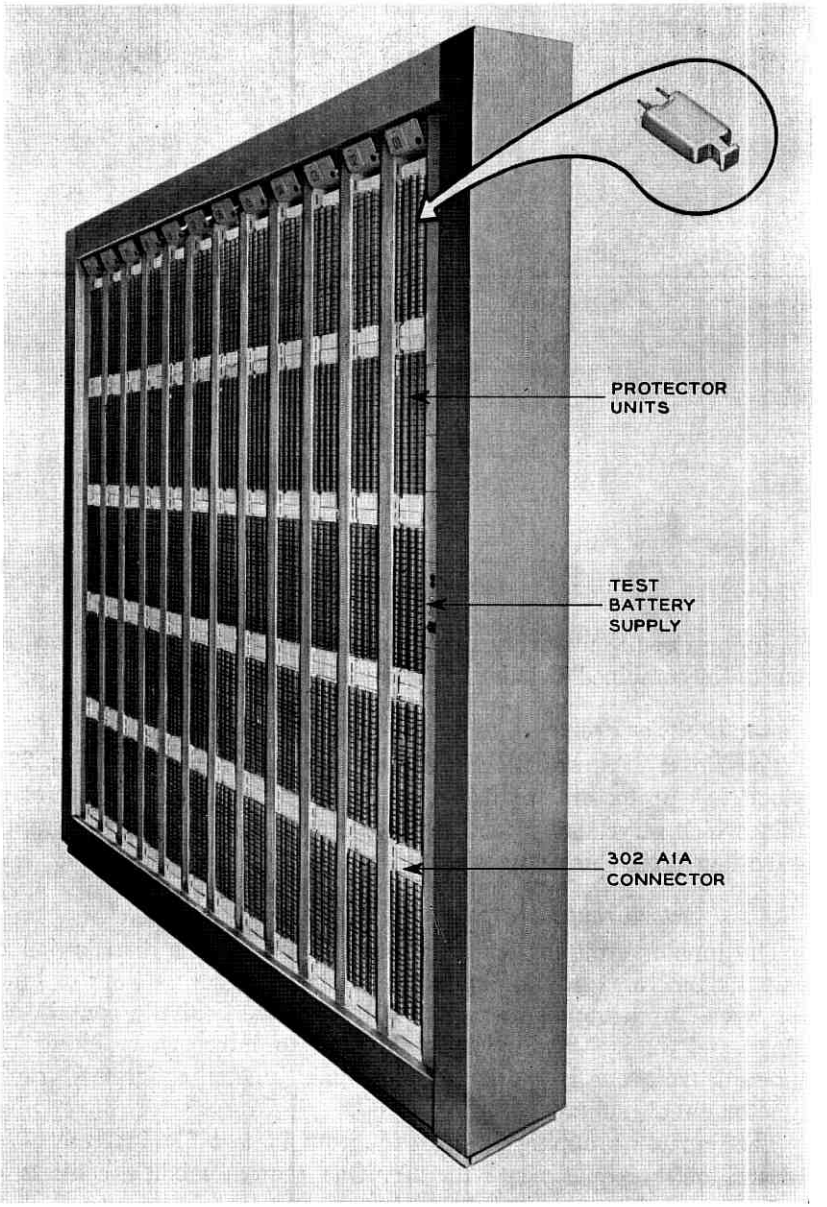



Fig. 27 — Protector frame.

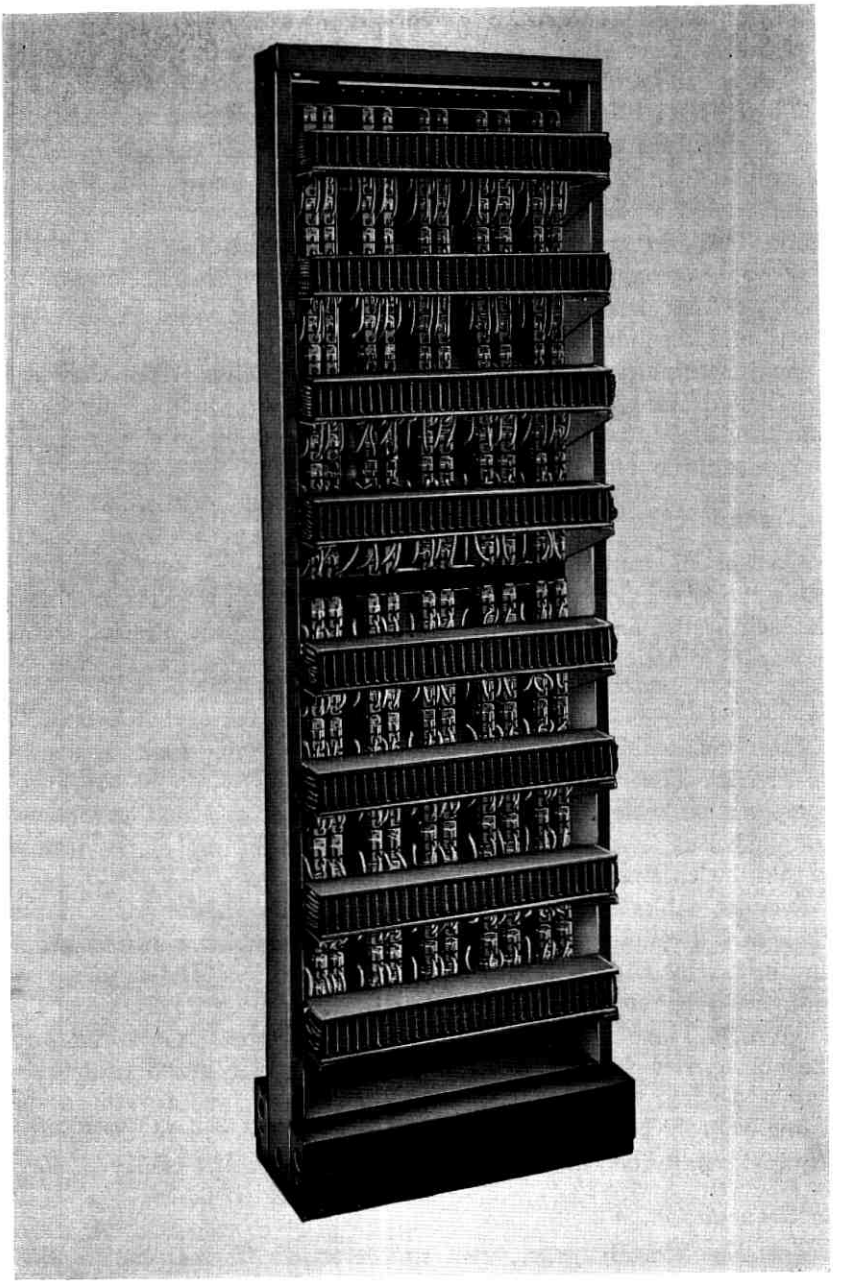


Fig. 28 — Junctor grouping frame.

appearances. Files chosen for junctor circuits are totally filled before a new junctor circuit file is assigned. A typical 4,000 line office with 4:1 concentration requires four files on one junctor grouping frame.

There are eight cable trays on the front of the frame, one for each shelf, which are used to store plug patch cords. Every file has four plugs and four jacks per shelf. Plugs and jacks on network files are wired such that each receives one junctor from every grid of a network. Two slips are wired between the line trunk networks and the junctor grouping frame by an installer on alternating shelf pairs of the file.[3] This permits all plugs to be wired in identical fashion and distributes the junctors to minimize blocking.

As junctor circuits are added to a junctor circuit file, they are distributed evenly over all eight shelves until the file is totally filled. Junctor distribution is achieved by patching the plugs of one shelf into the jacks of the same shelf in a prescribed manner.

## VII. INTERCONNECTING METHODS

Control of transient noise requires the use of compartmented cable racks, segregated cables and wiring paths on the frames, filters in each frame on all dc power supply feeders, and special frame grounding practices.

### 7.1 *Cabling Practices*

Interframe cables are run in compartmented cable racks as shown in Fig. 29. The relatively sensitive bus leads are run in the lower compartment which is shielded when the front and rear covers are installed. Bus transformers are mounted near the top of the frames so that the length of exposed leads will be minimized.

Scanner cables are run in a shielded channel at the front of the cable rack (equipment aisle side) where they can drop down to frame terminal strips with relatively short noise exposures. The tip and ring leads and relay control leads are placed in the center top section of the cable rack. These cables run down the front of the frame uprights to the appropriate frame terminals. The power distribution cables are run in the rear top section of the cable rack. These cables feed the frame from the rear. Separate cross aisle racks are used to provide the required separation between these different classes of leads.

### 7.2 *Connectorization*

The No. 2 ESS system uses connectorized frames that can be functionally tested in the factory. The connections between the main-

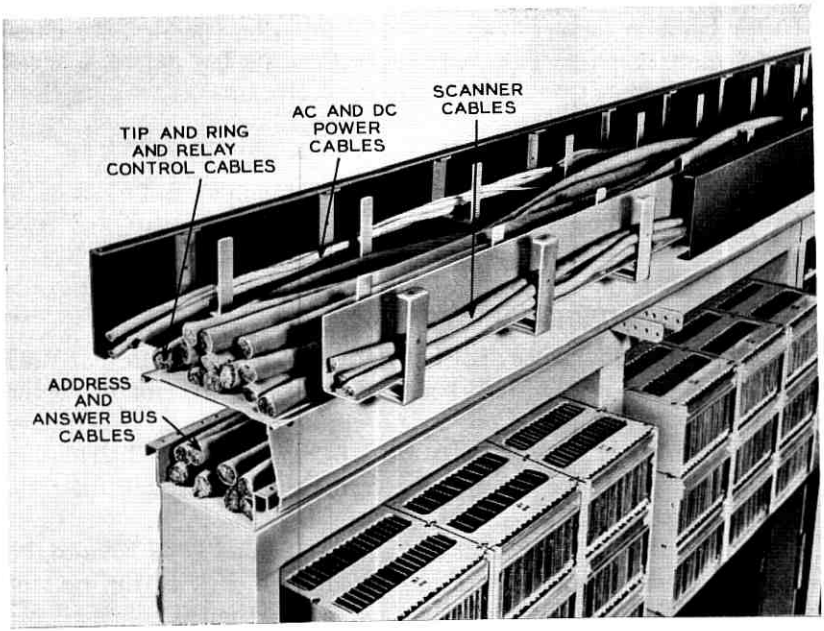



Fig. 29 — Cable rack.

tenance center, central processor, supplementary call store, and program store frames are provided by means of cables equipped with connectors. Similarly, all the leads between each network control junctor switch frame and its four associated line-trunk switch frames have connectors. The buses between the central processor, master scanner, universal trunk and junctor, and network frames also have connectors.

All of these connections represent an important but small percentage of the total number of interframe connections that are provided in a typical office. It is not practical to equip all leads with connectors because of the wiring variability from office to office.

The use of connectorized cables in central office equipment in place of installer-connected wiring has some obvious advantages as well as disadvantages. Some advantages are:

(*i*) installation intervals are shortened and installation costs reduced,

(*ii*) office growth procedures are simplified,

(*iii*) factory frame testing is facilitated, and

(*iv*) subsystem factory tests reduce installation troubles, the number of installation test sets and delays in replacing defective office equipment.

Some disadvantages of equipping cables with connectors are:

(*i*) additional equipment mounting space is required, and
(*ii*) equipment costs are increased by added apparatus and connections.

### 7.3 *Direct Current Distribution*

The No. 2 ESS dc power distribution system is similar to that used in the No. 1 ESS except that + 6.7 volt power is distributed to the control complex logic. This power is derived from the 6.7 volt power frame which converts − 48 volts to + 6.7 volts. Power distribution feeders are run between these power frames and the frame filters in the control complex.

The power distributing frames distribute + 24 and − 48 volts to the individual frame filters. Filters are designed to restrict the rate of change of current in the power feeders to less than 0.1 ampere per microsecond to insure less than one volt of noise at the power distributing frame bus.

Except for the control complex, all even-numbered frames are fed from an even-numbered power distributing frame and all odd-numbered frames from an odd-numbered power distributing frame. In the control complex, all frames in control unit 0 are fed from even-numbered power distributing frames and 6.7 volt power frames and those in control unit 1 from odd-numbered frames. Equipment frames having duplicated control equipment are supplied by two sets of feeders, one set from each power distributing frame.

In order to minimize noise caused by either stray circulating ground currents or transient noise potentials within the building, the No. 2 ESS equipment is grounded at a single point.

### 7.4 *Alternating Current Power Distribution*

Only single-phase 120-volt ac power is required in the No. 2 ESS switchroom and the only units requiring protected ac are: (*i*) the recorded announcement machine, (*ii*) maintenance teletypewriters, (*iii*) ac data sets, and (*iv*) key telephone equipment, when provided.

All loads are normally supported by commercial ac power. In the event of a power failure, the emergency power plant sup-

plies the required protected ac within five seconds and the remainder of the load, the essential ac, is furnished by the standby engine alternator. The protected and essential ac is distributed from an ac distribution panel on the miscellaneous power frame.

## VIII. SUMMARY

The following equipment design developments and philosophies have been instrumental in achieving the low-cost objectives for No. 2 ESS: (*i*) Use of currently manufactured, high volume production, low-cost apparatus items, (*ii*) plug-in growth units such as the call store, program store, and central pulse distributor, (*iii*) thin film integrated circuits (*iv*) judicious use of connectors on cables between major frames, (*v*) peripheral decoder, and (*vi*) single card writer. The design has provided for extensive factory testing, reduced installation intervals, and ease of growth. Compact designs provide additional savings in floor space.

## REFERENCES

1. Higgins, W. H. C., "A Survey of Bell System Progress in Electronic Switching," B.S.T.J., *44*, No. 6 (July–August, 1965), pp. 937–997.
2. 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., *43*, No. 5, Part 2 (September, 1964), pp. 2355–2439.
3. Digrindakis, J., Freimanis, L. Hofmann, H. R., Taylor, R. G., "Peripheral System," B.S.T.J., this issue, pp. 2669–2712.
4. "New Connectors Plug in ESS Frames for Fast Installation and Easy Growth," Bell Laboratories Record, *46*, No. 9 (October 1968), pp. 311–313.
5. Andrews, R. J., Driscoll, J. J., Herndon, J. A., Richards, P. C., and Roberts, L. R., "No. 2 ESS Service Features and Call Processing Plan," B.S.T.J., this issue, pp. 2713–2764.

# Service Programs

By MARSHALL E. BARTON, NEIL M. HALLER, and GUY W. RICKER

(Manuscript received March 6, 1969)

*The programming of a large switching machine requires that many service functions be performed on a general purpose computer. This article describes the service program package that supports the No. 2 Electronic Switching System. This package includes an advanced macro assembler to convert symbolic source programs into binary object programs, a loader to combine independent assemblies, and a simulator to provide checkout facilities.*

## I. INTRODUCTION

The No. 2 Electronic Switching System is a stored program control telephone switching system. The minimum program consists of about 75,000 instructions arranged in 22-bit words and written in more than 50 separate sections. It is clearly impractical to write such a program without extensive computer aids.

### 1.1 *Functions*

Major steps provided by the No. 2 ESS service programs are shown in Fig. 1. The assembler separately converts each symbolic source program from the language written by the programmer to the binary language of the No. 2 ESS. The loader then combines these separately assembled sections and resolves references between them. The output of the loader may be used by the magnet program to prepare the punched cards used in magnetizing the actual program store, or by the simulator. The simulator, using tables prepared by the assembler, furnishes more powerful program checkout facilities than would be practical on a switching machine and provides these facilities before the laboratory model is available.

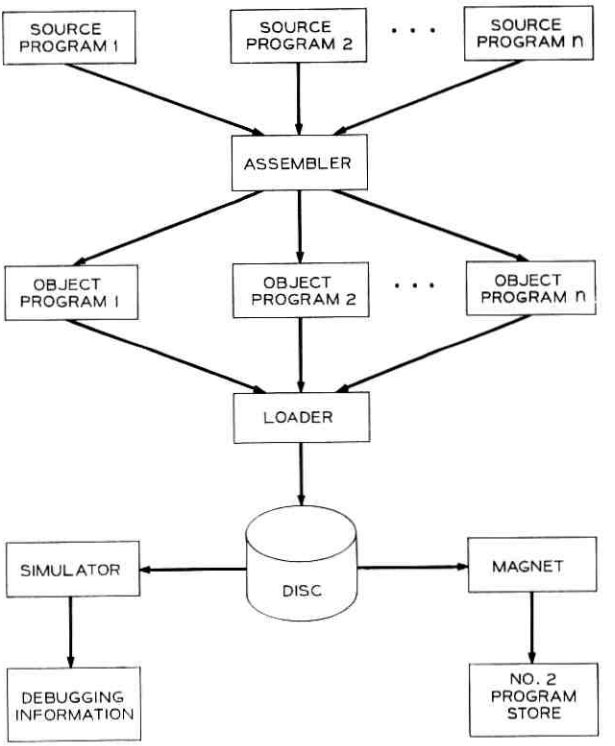



Fig. 1 — Major steps in program processing.

### 1.2 *Organization*

The No. 2 ESS service programs run on an IBM 360 computer system with at least 262,144 bytes of storage. They operate as a subsystem under Operating System 360. This gives the users a uniform interface to all service programs and isolates them from the complexities of System 360 job control language.

## II. ASSEMBLER

The primary purpose of an assembler is to convert a source program written in symbolic language to an object program which is in machine (binary) language. The switching assembler program (SWAP) was designed to do this for No. 2 ESS and several other stored program systems. Each machine instruction is represented by a symbolic code which the assembler translates to the appropriate bit pattern in the object program.

A programmer may assign a symbolic representation for any location in the object machine memory and then refer to that location symbolically in an instruction. By this method of reference, the program can be changed without regard for location changes caused by the insertion or deletion of words.

Symbolic addressing also allows a data constant to be defined with a symbolic name and then used in many places. If the value of the constant is changed, all the places it is used need not be hunted and changed. The method of defining data as well as many other functions is done through assembler control instructions called pseudo-operations. A pseudo-operation does not usually result in object code, but tells the assembler how to translate symbolic information it will encounter in the program.

## 2.1 *Input Syntax*

### 2.1.1 *Fields*

The general form of a SWAP input line is anchored free field. This means that, beyond column one, there are no restrictions as to where on the line the information appears. The four major divisions of an input line are called fields: the location field, if present, must start in column one; the operation field follows the location field; the variable field follows the operation field; and the comment field is last.

Fields must be separated by one or more blanks or a single comma, except for the comment field which must start with a sharp sign (#). (If a sharp sign appears in column one, the entire line is treated as commentary.) The variable field, sometimes called the address field, may be terminated in three ways: (*i*) any number of blanks followed by a sharp, (*ii*) the physical end of the line, or (*iii*) the logical end of line which is indicated by a semicolon. When a line is truncated by a semicolon, the character immediately following the semicolon is considered to be column one of the next line.

### 2.1.2 *Continuations*

A continuation may be indicated by the at sign (@) used in either of two places. If an at sign is the last nonblank character on a line, then the next line is a continuation. If an at sign appears in column one of a line, then the line with the at sign is a continuation of the previous line. In both cases, the lines are joined at the at sign which is discarded by the assembler.

### 2.2 *Basic Pseudo-Operations*

#### 2.2.1 EQU, SET, *and* TEXT *Pseudo-Operations*

The most basic pseudo-operations are those that define a symbolic representation for a quantity: the EQU and SET pseudo-operations assign a numeric value to a symbol; the TEXT pseudo-operation defines a symbolic name for a string of characters. While symbols defined with SET and TEXT may be redefined, those defined by EQU may not.

Either of the following lines would assign the value 10 to the symbol SYMB:

```
SYMB   EQU   10
SYMB   SET   10.
```

To define or redefine the symbol VOWELS with the string of five vowels as its value, the following statement would be used:

```
VOWELS TEXT 'AEIOU'
```

#### 2.2.2 JUMP *and* DO *Pseudo-Operations*

The pseudo-operations that allow the normal sequence of processing to be modified provide the real power of an assembler. In SWAP, the pseudo-operations that provide that control are JUMP and DO. JUMP forces the assembler to continue sequential processing with the indicated line, ignoring any intervening lines. As an example, consider the following sequence of lines:

```
        .
        .
        .
JUMP   .LINE
A   EQU   2
.LINE;B EQU   3
        .
        .
        .
```

The symbol A will not be defined because that line would be skipped under control of the JUMP. The symbol .LINE is called a sequence symbol and is treated not as a normal location field but only as the destination of a JUMP. The first character of a sequence symbol must be a period. The line that is "jumped" to may be either before or after the JUMP statement.

The JUMP is taken conditionally when an expression is used as the following example shows:

```
INC   SET   0
.AA;INC   SET   INC+1          # INCREASE COUNTER
   JUMP   .XX,INC>10           # IS IT OVER LIMIT
    .
    .
    .
   JUMP   .AA                  # GO AROUND AGAIN
.XX
```

The JUMP to .XX will occur only if the value of the symbol INC is greater than ten.

The DO pseudo-operation is used to control an assembly time loop and may be written in one of three forms:

(*i*) DO .LOC,VAR=INIT,TEXP,INC
(*ii*) DO .LOC,VAR=INIT,LIMIT,INC
(*iii*) DO .LOC,VAR=(LIST)

Types *i* and *ii* assign the value of INIT to the variable symbol VAR and then assemble all the lines up to and including the line with .LOC in its location field. The value of INC (if INC is omitted, 1 is assumed) is added to the value of VAR. For type *i*, the truth value expression TEXP is evaluated; if it is true, the loop is repeated. Type *ii* compares the value of VAR with the value of LIMIT; the loop is repeated if INC is positive and the value of VAR is less than or equal to the value of LIMIT. If INC is negative, the loop is repeated only when the value of VAR is greater than or equal to the value of LIMIT. Type *iii* assigns to VAR the value of the first item in LIST. Succeeding values are used for each successive time around the loop until LIST is exhausted.

The following is an example of the use of DO

```
        DO      .AA,INC=1,3
        GT        INC
.AA     TCS       TBL+INC
```

The assembler will produce the same output from these three lines as from the following sequence:

```
GT    1
TCS   TBL+1
```

```
GT    2
TCS   TBL+2
GT    3
TCS   TBL+3
```

### 2.3 *Symbols and Attributes*

Symbolic names are limited to fewer than 250 alpha-numeric characters at least one of which must be alphabetic. The alphabetic characters are considered to include the 26 upper and 26 lower case letters as well as the underscore and percent sign. A symbol may be used to identify a program store location, call store address, or a program parameter. The value of a symbol may be a 32-bit integer, a character string, or another symbol. In addition, every symbol may have up to 250 attributes which are 24-bit integers. The x attribute of the symbol A is represented by X(A). The following line, for example, sets the ALPHA attribute of the symbol SA to ten greater than the BETA attribute of symbol SB.

```
ALPHA(SA)  SET  BETA(SB)+10
```

#### 2.3.1 *Symbol Types*

Each symbol has an associated type character; a program store location symbol, for example, has type L. The type of each operand in an arithmetic or logical expression is used to determine the correct method of evaluating the expression. It is also used to check for illegal combinations of operands and to appropriately mark the error.

#### 2.3.2 *Available Type Characters*

Some of the available type characters and their meanings are:

A—absolute symbol
C—call store location symbol
D—program store data location symbol
J—truth valued symbol
L—program store location symbol
N—integer
T—text symbol
X—external symbol

### 2.4 *Arithmetic and Logical Expressions*

Arithmetic or logical expressions consist of a string of operands separated by operators or parentheses. An operand may be an integer, symbol, function call, attribute reference, indirect symbol, or character

string. A character string in an expression is represented by enclosing the string in either single or double quotation marks and is converted to a 32-bit binary integer when used in any operation except a comparison. An indirect symbol is a symbol defined by the text pseudo-operation where the character string definition is a valid arithmetic or logical expression.

The following is a list of the available arithmetic, logical, comparison, and special operators, listed in the order of hierarchal preference; the first to be evaluated are at the top of the list. The order of evaluation may be controlled by the use of parentheses.

*Special Operators*

| | |
|---|---|
| " or " | Indicates beginning or end of character string. |
| ? | Result is true when preceded by an operand. |

*Arithmetic Operators*

| | |
|---|---|
| ** or ↑ | Exponentiation |
| * | Multiplication |
| / | Division |
| unary − | Negation |
| unary + | No operation |
| + | Addition |
| − | Subtraction |

*Comparison Operators*

These operators are all of the same hierarchal value and yield a result of either true or false.

| | |
|---|---|
| = | Equals |
| > | Greater than |
| < | Less than |
| ¬= or ≠ | Not equal |
| =>, >=, or ≧ | Greater than or equal |
| =<, <=, or ≦ | Less than or equal |

The hierarchy of comparison operators is slightly different when they are used in a double relation; for example, A<B<C will have a result of true only when A is less than B and B is less than C.

*Logical Operators*

| | |
|---|---|
| & | Logical intersection. |
| ¬ | The intersection of the left operand with the complement of the right operand. |

| | |
|---|---|
| unary ¬ | Complement |
| \| | Logical union |
| ! | Logical exclusive OR |

2.4.1 *Predefined Arithmetic and Logical Functions*

Several built-in or predefined functions are available to aid in evaluating some of the more common or complicated expressions. The following is a partial list of the available predefined functions:

| | |
|---|---|
| D(EXP) | Results in the value of the expression EXP with any integers that occur interpreted as decimal. |
| E(EXP) | The result is 2 raised to the EXP power. |
| MAX($EXP_1$ , ··· , $EXP_n$) | Results in the maximum of $EXP_1$ through $EXP_n$ . |
| PAR(EXP) | Returns the even parity of the value of the expression EXP. |
| STYP(EXP,C) | Returns the value of EXP, but the type of the result is the character C. |

2.4.2 *Programmer-Defined functions*

To allow the programmer to define any number of new functions, the DFN pseudo-operation is provided. The general form of a function definition is written:

$$\text{DFN } F(P_1 , P_2 , \cdots , P_n) = A_1 : B_1 , A_2 : B_2 , \cdots , A_n : B_n$$

where F is the function name, the $P_k$ are dummy parameter names, and the $A_k$ and $B_k$ are any valid expressions that may contain the $P_k$ and other variables.

To evaluate the function, the $B_k$ are evaluated left to right. The result is the value of the $A_k$ corresponding to the first $B_k$ that has a nonzero or true value. If $B_n$ is not present, it is assumed to be true; also if all the $B_k$ are false, the value returned is zero. The parameter expressions are evaluated, and these values are used whenever a dummy parameter is encountered in the defining expression.

Two features are provided to allow an arbitrary number of arguments in the call of a function. The first is the ability to ask if an argument was implicitly omitted from the call (explicitly omitted arguments are treated as zero). This feature is invoked by a question mark immediately following the dummy parameter. If the argument was present, the result of the parameter-question mark is the value true; otherwise, the value is false. For example, the definition:

$$\text{DFN INC(X, Y)} = \text{X} + \text{Y} : \text{Y?}, \text{X} + 1$$

would yield the value 7 when called by INC(3,4) but the value of INC(3) is 4.

The other feature is the ability to loop over a part of the defining expression, using successive argument values wherever the last dummy parameter appears in the range of the loop. This feature is invoked by the appearance of an ellipsis (. . .) in the defining expression. The range of the loop is from the operator immediately preceding the ellipsis backward to the first occurrence of the same operator at the same level of parentheses. As an example, consider the following statement:

```
DFN SUM(A,B,X,Y)=A+X**(Y+1)+. . .+B/2
```

The range of the loop is from the + between the A and the X to the + following the right parentheses. The call SUM(2,18,3,1,2,3) would yield the same result as the following expression:

$$2+3^{**}(1+1)+3^{**}(2+1)+3^{**}(3+1)+18/2.$$

The loop may also extend over the expression between two commas as the next example shows. A function to do the exclusive OR of an indefinite number of arguments could be defined by:

```
DFN XOR(A,B,C)=A¬B | B¬A:¬C?,XOR(XOR(A,B),C, . . .)
```

### 2.5 *Macro Definitions*

The real power of an assembler lies in the flexibility it provides the programmer. The *macro* facilities incorporated in SWAP have more than the necessary features to make it one of the most powerful assemblers available. A macro instruction is an abbreviated form for a sequence of predefined instructions, pseudo-operations, or comments. Whenever a macro is called, the predefined sequence is generated in place of the macro call. The sequence of statements generated by a macro may be varied by the use of any of the several conditional assembly facilities provided.

The general form of a macro definition is:

```
MACRO
prototype statement
macro text lines
MEND
```

The prototype statement contains the name of the macro definition as well as the dummy parameter names which are used in the definition.

The macro text lines, a series of statements which make up the definition of the macro, will be reproduced whenever the macro is called.

The following is an example of a simple macro that may be used to move data from one call store location to another:

```
      MACRO
LOC MOVE FROM,TO
LOC RED FROM
      WRI TO
      MEND
```

The subsequent macro call:

```
SAVE MOVE NEWDATA,OLDDATA
```

would generate the following instructions:

```
SAVE RED NEWDATA
     WRI OLDDATA
```

### 2.5.1 *Macro Arguments and Operators*

The general forms of a macro argument are OP(ST) or PAR. OP is called the operation part of the argument and includes all characters up to the first left parenthesis. ST is called the strip and (ST), the body of the argument. PAR is an argument that is not of the op-strip form. Several macro operators are available to allow the programmer to obtain these parts as well as other pertinent information about an argument. A macro operator is indicated by its name character followed by a period and the dummy parameter name of the operand. For example, the operation part of a parameter named ARG is obtained by the use of O.ARG, and the strip is represented by S.ARG. Whenever the op part of a argument is requested and the argument is not of the op-strip form, a null value is returned; the strip of a non op-strip argument is the entire argument.

### 2.5.2 *Macro Subarguments*

Many times the strip of a macro argument consists of a sublist of parameters. Any subparameter may be accessed by subscribing the parameter name with the number of the desired subargument. Additional levels of subarguments are obtained with the use of multiple indexes. As an example, let parameter ARG assume the value P(Q,R(S,T)), then: ARG(0) represents P; ARG(1),Q; ARG(2),R(S,T); and ARG(2,2) would be replaced by T.

The macro operators may be used on the results of each other as well as on subparameters; for example, S.ARG(2) would refer to S,T.

The subargument indexes may be symbolic expressions that contain other macro parameters as the following example of a macro with a DO loop demonstrates. A macro to copy data from one call store location to any number of other call store locations could be written:

```
        MACRO
        COPY    PARM
        RED     O.PARM
        DO      .LOOP,K=1,N.PARM
.LOOP   WRI     PARM(K)
        MEND
```

The number macro operator, N, is replaced by the number of subarguments in its operand so that the DO will loop as many times as there are subarguments in PARM. The macro call:

```
COPY    DATA(SAVE,HOLD)
```

will then generate the following instructions:

```
RED     DATA
WRI     SAVE
WRI     HOLD
```

### 2.5.3 *Macro Functions*

To provide more flexibility with the use of macros, several system parameters and macro functions have been made available. A macro function call is replaced by the string of characters that is its result. The arguments of a macro function may consist of macro parameters, other macro function calls, literal character strings, or symbolic variables. An example would be the DEC macro function, which has a single argument that is a valid arithmetic or logical expression. The result is the decimal number equal to the value of the expression; the call DEC(7+8) would be replaced by 15.

The three major macro functions are:

(*i*) IS(*expression,string*) is replaced by *string* if the value of *expression* is nonzero; otherwise, the result is the null string.

(*ii*) IFNOT(*string*) is replaced by *string* if the *expression* in the previous IS had a value of zero; otherwise, the result is null.

(*iii*) STR($exp_1$,$exp_2$,*string*) is replaced by $exp_2$ characters starting with the $exp_1$ character of *string*.

A more sophisticated example of the use of macro functions is this version of the COPY macro:

```
        MACRO
        COPY    PARM
        RED     O.PARM
        DO      .LOOP,K=1,N.PARM
.LOOP   IS('PARM(K)'='HOLD',HGR)IFNOT(WRI PARM(K))
        MEND
```

Using the above definition, the call:

```
        COPY    DATA(SAVE,HOLD,LOC2)
```

would expand to:

```
        RED     DATA
        WRI     SAVE
        HGR
        WRI     LOC2
```

2.5.4 *Keyword Arguments*

It is often convenient to be able to override the positional relationship between the dummy parameters on the macro prototype line and the arguments on a macro call. This may be done when the macro is called by writing the parameter name followed by an equal sign and the argument string. An argument of this form is called a keyword argument. An example would be the following calls of the MOVE macro.

```
        MOVE    FROM=NEWDATA,TO=OLDDATA
```

or

```
        MOVE    TO=OLDDATA,FROM=NEWDATA
```

Both calls will expand to the same instructions as the expansion of the MOVE macro without keyword arguments.

2.5.5 *Default Arguments*

Another convenience is the ability to have a standard, or default, value for a parameter. The default value would be used whenever the argument was omitted from the call. The default value must be assigned on the macro prototype line by an equal sign and the default value after the dummy parameter name. Another version of the MOVE macro is an example of assigning default values.

```
MACRO
MOVE    FROM=TEMP,TO=TEMP+14
RED     FROM
WRI     TO
MEND
```

The call:

```
MOVE    TO=OLDDATA
```

would then expand to:

```
RED     TEMP
WRI     OLDDATA
```

### 2.6 *Automatic Instruction Insertion*

No. 2 ESS instructions are put in the format of one or two per word. The half-word instructions of the No. 2 ESS occasionally cause a no-operation (NOP) instruction to be required. When there are an odd number of half-word instructions between full-word instructions or when the destination of a transfer would otherwise be in the middle of a word, a NOP is inserted by the assembler.

The half-word transfer commands specify a five bit address. The destination of such a transfer is thus limited to the same block of 32 words as the transfer. An instruction (FIL) is available to extend the addressing range to 1024 words by leaving five bits in a buffer. The assembler will give an error message each time a short transfer is used that: (*i*) requires but does not have an associated FIL, (*ii*) has a FIL that was not needed, or (*iii*) is insufficient even when a FIL is used.

When the programmer adds or deletes an instruction, a short transfer may require a FIL where it was not previously needed. SWAP inserts the appropriate FIL instruction wherever it is needed and attempts to place it where a NOP was required in the right half of a word. It is extremely difficult to have only the minimum number of FIL's and, therefore, some extra FIL commands will be inserted in the program. The automatic FIL insertion feature may be turned off if the programmer so desires.

### 2.7 *Text Manipulating Facilities*

Some of the more exotic features provided by the switching assembler program are the character string pseudo-operations and the dollar functions, so called because the function names all start with a dollar sign.

### 2.7.1 HUNT *and* SCAN *Pseudo-Operations*

The HUNT pseudo-operation allows the programmer to scan a string of characters for any break character in a second string. It will then define two TEXT symbols consisting of the portions of the string before and after the break character.

The SCAN pseudo-operation provides the extensive pattern matching facilities of SNOBOL3 along with success or failure transfer of control.[1] These features, too diverse to be discussed here, are covered in the references.

### 2.7.2 *Dollar Functions*

Dollar functions are very similar to macro functions in that the result of a dollar function call is a string of characters that replace the call. The dollar functions may be used on input lines as well as in macros. For example, $(TSYM) would be replaced by the character string which is the value of the text symbol TSYM. A very useful feature of the dollor functions is in the ability to call a one-line macro anywhere on a line by preceding the macro name with a dollar sign and following it with the argument list in parenthesis. For example, the macro:

```
        MACRO
        CHECK   A,B

IS(A<B,DEC(B-A) MORE)IFNOT(DEC(B-A)OVER)

        MEND
```

could be called by:

```
X       SET     5
        LGR     X       #       $CHECK(X,8)
```

but the line would appear in the assembly listing as:

```
        LGR     X       #       3 MORE
```

## 2.8 *The Assembly Listing*

Since the input line format for SWAP is free field, the assembly listing of the source lines may appear quite unreadable. Therefore, the normal procedure is to have the assembler align all the fields when a line is printed. For example, a programmer may punch his cards:

```
        TRA             LOC # GO BACK
REST            LGR 7; GRXAA # LOAD AA
```

but the assembly listing would show the lines thus:

```
        TRA     LOC     # GO BACK
REST    LGR     7
        GRXAA           # LOAD AA
```

The position of the fields as well as the position of the line is a programmer option. Some of the other options that are available to control the format of the listing are: double spacing, titling at the top and bottom of each page, and several classes of lines that may be printed at the programmer's discretion.

### 2.9 *Inputs*

SWAP may receive its original input from a card, disk, or tape data set. The SOURCE pseudo-operation allows the programmer to change the input source at any point within a program. Another source of input is the EDITOR program, which provides extensive facilities for making changes or corrections in a program.

#### 2.9.1 *The* EDITOR *Program*

Any SWAP input line that contains a colon in column one is assumed to be an EDITOR control card and, therefore, invokes the EDITOR program. The EDITOR is then responsible for retrieving a source data set, making the indicated changes, and passing each line back to SWAP to be assembled. The EDITOR provides facilities for inserting, replacing, or removing lines as well as modifying a part of a line and moving or copying a group of lines to another position within the data set. Since the normal output of the EDITOR goes directly to the assembler, the original data set is not changed unless the programmer explicitly requests that the changes be permanently incorporated in a new copy of the data set.

#### 2.9.2 *Libraries*

SWAP also has facilities to save symbol, instruction, or macro definitions in the form of libraries which may be loaded later in another assembly. When, for example several programs make use of a common set of macros, it is desirable to obtain them from the same source. The SOURCE pseudo-operation could be used for this, but it would require that each symbol, instruction, or macro be completely processed by the assembler. As this is relatively slow and inefficient, a method of producing a library which contains the processed definitions is provided. Later, if a program requires it, those definitions may be loaded and used, bypassing the costly definition process.

### 2.10 *Outputs*

The output of the assembler normally consists of the assembly listing and the object program module (see below) but other outputs may be obtained upon request. The libraries are one example. Special outputs can be produced from stylized comments included in each No. 2 ESS program. One type of comment provides information so that a flowchart of the program can be generated by machine. Another type of comment is used to produce a manual explaining all the teletypewriter messages that the program might issue.

Also provided as an optional output of the assembler are the results of the macro expansions and dollar substitutions. The programmer controls the format of each line as well as the deletion of undesired lines. This allows the assembler macro facilities to be used to produce an input data set for any of the No. 2 service programs or IBM 360 support programs.

## III. LOADER

The LOADER program accepts output from SWAP assemblies. The assembled programs are combined and placed on disk storage as a paged image of the program store. All interprogram linkage is performed; that is, all external symbolic references are resolved. The program store image is in the form required by the simulator or twistor card preparation program.

### 3.1 *Object Program Module*

The output of a SWAP assembly is called an object program module. All object program module's of a project are normally contained in a single partitioned data set. They may exist only on a direct access device, although they may be saved on magnetic tape. There may be a number of object program module's for any program, reflecting various stages of development. The assembler creates a PRIVATE, or working copy. A utility program creates a PUBLIC copy from a relatively debugged PRIVATE copy (and pushes down the other PUBLIC copies). The LOADER normally loads the most recent PUBLIC copies but may load others as described later under the LOAD verb.

### 3.2 *Outputs of* LOADER

#### 3.2.1 *Program Store Image*

The primary output of the LOADER is a paged image of the program store. It is made up of a 2048-byte record for each two planes (512

ESS words) loaded, plus a directory to indicate which plane is on which record.

### 3.2.2 *Printed Output of* LOADER

The LOADER produces four printouts. These are shown in the Appendix.

All control cards processed by the LOADER are listed. Diagnostics are self-explanatory; any error flags are explained at the end of the control card listings.

Unless suppressed, a loading map is generated. This map includes, for each program loaded, the version (PUBLIC or PRIVATE), time and date of assembly, first and last locations in the program, the number of a tape (if any) on which the assembly listing is stored, and a remark. The remark indicates whether the program was implicitly or explicitly loaded (see Section 3.3). If a program could not be found in the object program module data set, it is marked undefined. Printed below the loading map is a statement of whether any areas of the program store were loaded more than once. If overwrites were present, they are listed.

If a cross-reference table is requested, it lists all external references that were resolved by the loader. This may be a very large list. If the listing is not requested, only those references for which a diagnostic is generated are listed. The cross-reference table may be sorted by symbol name.

## 3.3 *Implicit Loading*

Unless EXPLICIT loading is specified, IMPLICIT loading is assumed. This means that any program that is referred to by one that has been previously loaded is also loaded. It is, therefore, possible to load all programs by explicitly mentioning only one. Programs not wanted may be excluded from the loading.

## 3.4 LOADER *Features*

The function of the LOADER is best described by describing some of the primary verbs.

### 3.4.1 LOAD *Statement*

The LOAD statement describes which programs, and what versions, are to be loaded. An unqualified program name indicates the latest PUBLIC version. Program names may be qualified by PRIVATE or a date. When the PRIVATE qualification is used, the PRIVATE version is loaded. When a date is specified, the latest version assembled not later than that date is loaded.

*Example:* LOAD BLMM IOMAINT(PRIVATE) IOTEST(6/12/69)

#### 3.4.2 EXCLUDE *Statement*

The EXCLUDE statement lists programs to be excluded from implicit loading.

*Example:* EXCLUDE CSUB PCMAINT

#### 3.4.3 SET *Statement*

The SET statement defines or changes the value of a PUBLIC symbol; that is, a symbol to which an external reference is made. In Example 1, the symbol RETURN1 in the program BLMM is given the value 10436 octal. Example 2 equates the symbol SUB23 in the program IOSUB to the value of SUBA2 in CSUB.

*Example 1:* SET BLMM.RETURN1 = 10436

*Example 2:* SET IOSUB.SUB23 = CSUB.SUBA2

## IV. SIMULATOR

The No. 2 ESS simulator, SMILE,* provides a powerful program checkout, or debugging, facility. It is also a vehicle for investigating the effects of proposed system changes.

### 4.1 *Simulation*

A typical user assembles his program, loads it along with related programs, and simulates it to find errors (see Fig. 1). He requests printouts of pertinent data at points where he expects specific results. If the results are not right, he examines all of the outputs. If he fails to find the trouble, he will simulate again, producing more output around the problem area. In this way, he can close in on the error. A map of the path taken by the program at each branch and a printout of the contents of the registers are tools for finding out what went wrong.

Proposed system changes can be evaluated using the simulator. For example, the effect on system capacity of changes in command timing can be studied by incorporating the changes into the simulator and observing the effects by simulating the call processing programs.

---

* SMILE is an acronym for switching machine interpreter for lazy engineers!

### 4.2 *Objectives of* SMILE

The main objectives in the design of SMILE were completeness of simulation, ease of use, and speed. It is thus possible to simulate a complete call using few control statements in a reasonable amount of IBM 360 time.

#### 4.2.1 *Coverage*

SMILE simulates most processor commands, some No. 2 ESS input-output, but no internal wired logic. All commands used by the non-maintenance programmer are simulated. To perform No. 2 ESS input-output, special control statements have been developed. These enable one to place digits into originating registers at specified times. Other more general control statements allow locations in call store to be changed based on the reaching of specified program store locations or the passing of a specified amount of No. 2 ESS time. Special No. 2 ESS output messages are produced on certain external commands. All of these features are discussed in more detail in the following paragraphs.

#### 4.2.2 *Ease of Use*

In order to make SMILE easy to use, default values on everything possible have been set to the most common value. For example, call store words are initially zero, and all scanners are initially ones. Each input to SMILE is written in the natural language for that item. Time, for example, may be written in cycles, microseconds, or milliseconds. Program store addresses and call store addresses are assembled and specified symbolically.

#### 4.2.3 *Speed*

To be useful in a practical environment, a simulator must be fast. The simulation of a typical ESS instruction takes less than 30 microseconds on an IBM 360 model 65. This basic simulation ratio of better than 10 to 1 (elapsed time to simulated ESS time) is achieved by trading space for speed in the more common routines.

Another contributor to SMILE's speed is the preprocessing of control statements that will be performed during simulation. Control statements are converted to interpretive code which is executed each time the function is performed. The interpretive code produced must run fast, with little consideration given to how long it takes to produce the code. The interpretive code produced must not use a lot of core

because this would greatly limit the number of control statements possible.

Generation of outputs consumes a large portion of running time, so SMILE limits unnecessary printing. Unless told otherwise, SMILE only prints a final dump of the nonzero registers and call store, and a few error and special messages. Error messages are automatically turned off after ten of a kind have been printed. To produce more output, the user must explicitly ask for it.

### 4.3 *Control Language*

#### 4.3.1 *Initializing Statements*

The control language that has been developed is natural and easy to use. To initialize a register, one merely writes an "=" statement as follows: (*i*) name of register, (*ii*) =, and (*iii*) value to be stored in the register. For example, GR = 1243. Since No. 2 ESS programmers are accustomed to octal code, data constants are interpreted as octal numbers. Counts and time are interpreted as decimal numbers. To initialize a word in call store, one writes an "=" statement as follows: (*i*) symbolic name of area in call store or octal address, (*ii*) =, and (*iii*) value to be stored. For example, CSTBL = 421 or 2310 = 1337.

Of the two groups of scanners in the No. 2 ESS, one is primarily used for lines and the other for trunks. "TRUNK(3,4)" refers to row number 4 in trunk scanner number 3. "LINE(2,5,3)" refers to bit number 3 of row number 5 of line scanner number 2. These functions may be used on the left side of "=" statements to initialize the scanners. For example, TRUNK(3,7,2) = 1 or LINE(5,4) = 176777.

#### 4.3.2 *Plants in Program Store*

Simulation can be interrupted when specified program store locations are reached. At a location, a set of features can be planted. These would be performed each time that location is reached. Initialization statements as well as other features can be "planted" at a location.

#### 4.3.3 *Time and Automatic Interrupts*

As each command is simulated, the timer is incremented by the amount of time the command takes in the No. 2 ESS. Concurrently, a check is made to see if the user has requested an interrupt at this point. There are three ways a user can request such an interrupt:

(*i*) The TIME control statement says to perform the "range" (control

statements after the TIME and before the next TIME or plant) when the timer reaches the time specified by the first operand. If there is a second operand, the range will be performed again when that amount of time has elapsed, and each time thereafter. If a third operand exists, it tells when to stop processing this TIME.

(*ii*) The AFTER control statement is similar to the TIME control statement except that it is planted at a location. Its first and second operands correspond to the second and third operands of the TIME control statement. It enables a programmer to interrupt after a certain amount of time has passed after reaching a certain location.

(*iii*) The INOUT control statement generates other control statements which create an INOUT interrupt at the time indicated by the first operand. For ease of use, 25 milliseconds is assumed if the first operand is missing.

#### 4.3.4 *Digit Insertion into Originating Register*

A digit control statement places digits into an originating register in call store, one for each time the control statement is encountered. The verb DIGDP, DIGMF, or DIGTT indicates how the second operand list is to be interpreted—dial pulse, multifrequency, or *Touch-Tone*®-dialing, respectively. The first operand indicates which originating register receives the digit. The second operand is a list consisting of symbolic codes indicating which digits are to be deposited.

If, as is usually the case, the digit control statement is used in the range of a TIME, the first digit is deposited at the time indicated by the first operand of the TIME, and successive digits are deposited at increments of time indicated by the second operand of the TIME. If the digit control statement is planted in a range at a program store location, the digits are deposited one at a time each time that program store word is simulated.

#### 4.3.5 *Symbolic Input*

SMILE has been designed to allow symbolic reference to program store locations. A program may thus be changed without modification of the control statements. The SWAP assembler produces a symbol table with equivalences which the simulator uses. When symbols from several programs are referred to, it is necessary to indicate to which program each symbol belongs. The prefix notation is one way to do this. INI.LOOPMORE is used to refer to the location LOOPMORE in program INI. Prefix notation is used when a small number of symbols are needed from a given program.

The SYMBOLS PRNAME control statement informs SMILE that all nonprefixed symbols between here and the next SYMBOLS control statement are to be found in the symbol table from the program PRNAME.

#### 4.3.6 *Conditions*

The ability to conditionally perform various control statements dependent on the comparison of variables is an essential part of the simulator control language. The IF and UNLESS verbs enable one to perform these comparisons in a natural manner.

"IF GR = 2" means to perform the next control statement if the contents of the general register (GR) is 2. Four relations are permitted: = (equal), < (less than), > (greater than), and ¬ (not—used as a prefix). Combinations are also permitted: <=, >=, ¬=, ¬>, ¬<=, and so on. The not sign (¬) negates the entire relation regardless of where it occurs.

"UNLESS LR=4" does the same thing as "IF LR ¬= 4." A condition applies to the single control statement that follows it, unless that statement is a BEGIN. Then the condition applies to the "block" of control statements starting with the BEGIN and ending with a paired END. Blocks may be nested up to a maximum depth of ten.

The expressions used on both sides of relations and "=" statements may include registers, constants, call and program store symbols, scanner functions, and the special dummy registers X0 through X20 combined with the following operators:

@ indirection (constant refers to call store)
** integer exponentiation
* multiplication
/ integer division (truncated)
+ addition
– subtraction
& logical AND
| logical OR
() parentheses may be used to alter the above hierarchy (operations on top are performed first).

For an example, consider: "IF GR = @(CSTBL+X0)*2**5&3740." The contents of dummy register X0 (which was previously set by something like X0 = 12 or X0 = X0+1) is first added to the address CSTBL. This sum is used to index into call store. The word thus obtained is saved while 2 is raised to the fifth power. Next, the saved word is multiplied by the power and the result is masked by 3740.

For another example, consider: "IF AA = LR|GR&AA −CA* LW**@CSEXP." The operations will be performed in the exact reverse order of the way they are written since each successive operator is higher in the above list than its predecessor.

#### 4.3.7 *Normal Sequence Breakers*

In the process of using conditions, it is often desirable to jump over a set of control statements. This can be done by using the JUMP verb followed by a sequence symbol and by placing that sequence symbol just before the control statements with which processing should continue. This is a branch among control statements. Sequence symbols are by definition symbols which start with a period. For example, in the following input stream: ".CASE1" and ".CASE2" are sequence symbols:

```
IF GR = 1236 JUMP   .CASE1
IF LR = 42   JUMP   .CASE2
JUMP  .OUT
.CASE1  GR = 12   SNAPTR  JUMP  .OUT
.CASE2  LR = 1232 SNAPTR  JUMP  .OUT
```

"JUMP .OUT" naturally means to jump out of this range and go back to simulation at the point it was interrupted.

It is often desirable to skip over a section of program or go to some other place. This can be done by using the GOTO verb followed by a program store address (symbolic or constant). For example, GOTO LOOPMORE+2, or GOTO LOOPMORE−1. This is a branch from performing control statements to the simulation of a particular program store word.

The verb which specifies the end of control statement processing and the start of simulation is START followed by the program store address where simulation is to begin. No address means simulation starts at the origin of the first program loaded.

### 4.4 *Outputs*

#### 4.4.1 *Transfer Trace*

Transfer tracing is the process of following the flow of a program by printing out a line every time an instruction is executed out of sequence. Each line of trace output includes the type of branch and ten data words. The different types of branches are:

| | |
|---|---|
| ADV | advance command (when progress mark found) |
| GOTO | GOTO control statement |

| | |
|---|---|
| PA= | = control statement |
| PIBn | program interrupt begin command |
| PIEn | program interrupt end command |
| TRACE | all transfer commands (when transfer is taken). |

There are two control statements used with tracing:

```
TRACE N
FORMAT LIST.
```

The TRACE N control statement starts tracing and continues until N lines have been printed. It may be placed in the initial input stream, at a location, or in the range of a TIME. In the FORMAT control statement, LIST indicates which registers, call and program store words, scanner rows, and special registers X0–X20 are to be printed. Furthermore, the FORMAT statement is dynamic; it can be used not only in the initial input stream but also as a plant or in the range of a TIME. If a FORMAT statement is not encountered first, a default FORMAT is automatically generated whose LIST is "GR, CA, @CA, AA, KM, LR, LM, LW, RF, TB."

#### 4.4.2 DSNAP *and* SNAPTR

A handy debugging device is the DSNAP statement which prints out all registers and call store words that have changed since the last DSNAP. The first DSNAP prints out all nonzero words. This type of output is slow since it has to compare the current contents of all of call store with the previous contents and also save the current as the new previous and print several lines showing changes. This kind of tool, although slow, can find errors that would otherwise go undetected. A wild write into call store shows up very quickly.

Whenever it is encountered, the SNAPTR feature causes the printing of a TRACE line. It and all the other features can be planted at a program store location or caused to happen at a given time.

The DATATR feature is a special feature which causes the printing of a TRACE line after each occurrence of the DATA command. The DATA command is used to retrieve data, such as translations, from the program store. A DATA trace effectively records the progress of a call by monitoring the translators. DATATR is usually placed in the initial input stream but can also be planted or put in the range of a TIME.

#### 4.4.3 *Special Messages*

Most of the external commands are accompanied by special printouts. These printouts give pertinent information so the programmer can

see if the program produces the necessary outputs to perform the desired switching. Several other special messages are associated with the hold-get counter and transfers.

PRINT control statements control the printing of the above messages. The default case is to print each message ten times and then suppress the message. The PRINT control statement can turn a message OFF, or ON, or ON for a certain number of times. This statement is also dynamic—the ON-OFF-COUNT status of any message can be changed during simulation.

#### 4.4.4 *Final Dump*

At the completion of each simulation, an automatic dump of all nonzero registers and nonzero call store locations is given. The dump takes the form of a DSNAP assuming the previous DSNAP was all zeros.

#### 4.4.5 *Symbolic References to Program Store*

With each request for output, the following header prints on the left side of each line:

(*i*) The current No. 2 ESS time.

(*ii*) The current contents of the program address register. (The location of the plant. On transfers, the from location.)

(*iii*) The symbolic equivalent of the program address which includes the symbolic name of the program, the nearest previous symbol, and the increment from the symbol.

(*iv*) The name of the feature.

On features requiring multiple line output (like DSNAP), the header information appears only on the first line. The remaining fields are a function of the feature requested. For example, while tracing, there are ten data fields. For special messages, it may be a sentence with a variable inserted. Examples are shown in the appendix.

#### 4.4.6 *Teletypewriter Output*

During simulation, a user's program may request the printing of No. 2 ESS teletypewriter messages. This is done by having the No. 2 ESS teletypewriter program produce printing and control characters, one at a time, during the 25 millisecond interrupt. This stream of characters is saved on a scratch data set. When the simulation of the user's program is complete, control may be passed to a special set of control statements which will cause the teletypewriter program, via a fast simulation, to generate all characters in any remaining messages.

When all simulation is complete, the teletypewriter post processor sorts by teletypewriter number and prints all the saved characters. This last step is omitted if the user never starts any teletypewriter messages.

### 4.5 *Internal Structure*

SMILE consists of three basic divisions: input control statement interpreter, command simulator, and teletypewriter message processor.

During the first phase, all control statements are scrutinized. Flags are planted where indicated functions are to be accomplished, the interpretive code required to evaluate expressions is generated, and a queue of all time-based interrupts is formed.

When the last control statement, START, is encountered, phase two, the actual simulation, commences. Each command from the program is simulated by a subroutine selected by using its operation code as an index into a transfer table. Input and output functions are performed when flags requesting them are encountered. The interrupt list is checked each time a command subroutine increases the timer. Ranges are executed whenever they are reached. Expressions are dynamically evaluated each time they are encountered and conditions are performed based on these results. Whenever a "STOP" or an unrecoverable error occurs, the simulation phase stops, and post processing takes place. Any teletypewriter messages generated during simulation are finally printed.

## V. SUMMARY

The service programs described in this article comprise about 50,000 words, about two-thirds of which are the assembler. These programs are now in use for the No. 2 ESS project. The assembler was designed to be common to several projects at Bell Laboratories, and the authors wish to acknowledge the contribution of Messrs. R. E. Archer, A. J. Emrick, and E. Walton to the design and implementation of this program.

Typical execution times are one minute for assemblies (up to two minutes for 250 page assemblies), one minute to load all programs, and one to three minutes for simulations. The use of the powerful facilities of SWAP varies greatly among programmers. Some make extensive use of the macro features; others, almost no use. This appears to be caused by the difficulty of learning these features well enough to use them effectively. Once a programmer overcomes this obstacle, he uses the tools often.

Most system programs have been extensively simulated; as a result,

they were relatively debugged before being tested in the laboratory model. For example, only five program bugs were found using the laboratory model before the first call was completed. The simulator is expected to continue to be a valuable tool for the life of No. 2 ESS program development.

## REFERENCES

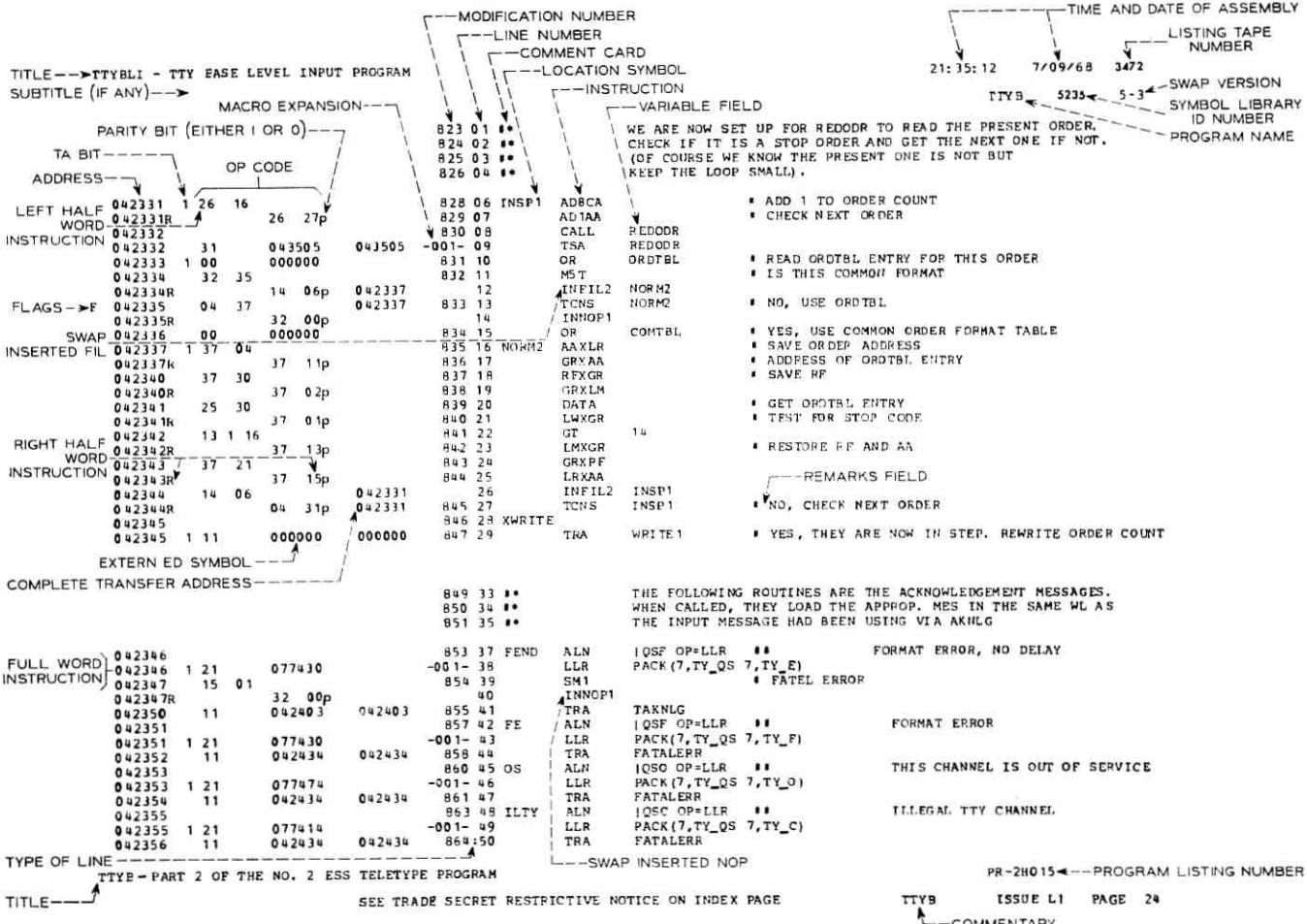
1. Farber, D. J., Griswold, R. E., and Polonsky, I. P., "SNOBOL, A String Manipulation Language," J. of the Association for Computing Machinery, *11*, No. 1 (January 1964), pp. 21–30, and "The SNOBOL3 Programming Language," B.S.T.J., *45*, No. 6 (July 1966), pp. 895–944.

## APPENDIX

### *Computer Samples*

The tables on the following pages are photographic reproductions of computer printouts.

TABLE I—SAMPLE ASSEMBLY LISTING




```
                                                  823 01 **
                                                  824 02 **
                                                  825 03 **
                                                  826 04 **
                                                  WE ARE NOW SET UP FOR REDOOR TO READ THE PRESENT ORDER.
                                                  CHECK IF IT IS A STOP ORDER AND GET THE NEXT ONE IF NOT.
                                                  (OF COURSE WE KNOW THE PRESENT ONE IS NOT BUT
                                                  KEEP THE LOOP SMALL).
042331  1 26  16                                 828 06 INSP1  ADBCA                   * ADD 1 TO ORDER COUNT
042331R                 26  27p                  829 07        AD1AA                   * CHECK NEXT ORDER
042332                                           830 08        CALL    REDOOR
042332          31      043505      043505       -001- 09      TSA     REDOOR
042333  1 00            000000                   831 10        OR      ORDTBL          * READ ORDTBL ENTRY FOR THIS ORDER
042334          32  35                           832 11        MST                     * IS THIS COMMON FORMAT
042334R                 14  06p     042337       -001- 12      INFIL2  NORM2
042335          04  37              042337       833 13        TCNS    NORM2           * NO, USE ORDTBL
042335R                 32  00p                  -001- 14      INNOP1
042336          00      000000                   834 15        OR      COMTBL          * YES, USE COMMON ORDER FORMAT TABLE
042337  1 37    04                               835 16 NORM2  AAXLR                   * SAVE ORDER ADDRESS
042337R                 37  11p                  836 17        GRXAA                   * ADDRESS OF ORDTBL ENTRY
042340          37  30                           837 18        RFXGR                   * SAVE RF
042340R                 37  02p                  838 19        GRXLM
042341          25  30                           839 20        DATA                    * GET ORDTBL ENTRY
042341R                 37  01p                  840 21        LWXGR                   * TEST FOR STOP CODE
042342  13  1  16                                841 22        GT      14
042342R                 37  13p                  842 23        LMXGR                   * RESTORE RF AND AA
042343  7   37  21                               843 24        GRXRF
042343R                 37  15p                  844 25        LRXAA                   REMARKS FIELD
042344          14  06      042331      042331   -001- 26      INFIL2  INSP1
042344R                 04  31p     042331       845 27        TCNS    INSP1           * NO, CHECK NEXT ORDER
042345                                           846 28 XWRITE
042345  1 11            000000      000000       847 29        TRA     WRITE1          * YES, THEY ARE NOW IN STEP. REWRITE ORDER COUNT

                                                  849 33 **     THE FOLLOWING ROUTINES ARE THE ACKNOWLEDGEMENT MESSAGES.
                                                  850 34 **     WHEN CALLED, THEY LOAD THE APPROP. MES IN THE SAME WL AS
                                                  851 35 **     THE INPUT MESSAGE HAD BEEN USING VIA AKNLG

042346                                            853 37 FEND   ALN     |QSF OP=LLR   **          FORMAT ERROR, NO DELAY
042346  1 21            077430                   -001- 38      LLR     PACK(7,TY_QS 7,TY_E)
042347          15  01                           854 39        SM1                     * FATEL ERROR
042347R                 32  00p                  -001- 40      INNOP1
042350          11      042403      042403       855 41        TRA     TAKNLG
042351                                            857 42 FE     ALN     |QSF OP=LLR   **          FORMAT ERROR
042351  1 21            077430                   -001- 43      LLR     PACK(7,TY_QS 7,TY_F)
042352          11      042434      042434       858 44        TRA     FATALERR
042353                                            860 45 OS     ALN     |QSO OP=LLR   **          THIS CHANNEL IS OUT OF SERVICE
042353  1 21            077474                   -001- 46      LLR     PACK(7,TY_QS 7,TY_O)
042354          11      042434      042434       861 47        TRA     FATALERR
042355                                            863 48 ILTY   ALN     |QSC OP=LLR   **          ILLEGAL TTY CHANNEL
042355  1 21            077414                   -001- 49      LLR     PACK(7,TY_QS 7,TY_C)
042356          11      042434      042434       864 50        TRA     FATALERR
```

## TABLE II—LOADER CONTROL CARDS

```
            PRINT     XREF
            LOAD      IO
            LOAD      TESTPR1(PRIVATE)
            LOAD      TESTPR2  TESTPR5
L           EXCLUDE   TESTPR4,TESTPR5   # PRGMS NOT YET DEFINED.
            #
            #               UGLY NOT AVAILABLE, USE TESTPR1
               ALIAS    TESTPR1 UGLY
               EXCLUDE UGLY
U           MISTAKE
            SET       TESTPR2.SYM12 = 5236
            SET       PR13.ALPHA = TESTPR2.BETA
            SET       CPDIG.IOSTOP = IOMAINT.IOPAUSE
            SET       CPDIG.NSTORES = 3
            ALIAS     IO,IOMAINT
            ENDL
```

```
L FLAG  MEANS  PROGRAM TO BE EXCLUDED WAS EXPLICITLY LOADED ABOVE.
U FLAG  MEANS  INVALID VERB  -  IGNORED.
```

## TABLE III—LOADING MAP

| VERSION | PROGRAM | ORIGIN | LAST | ASSEMBLED | | REMARKS | TAPE |
|---|---|---|---|---|---|---|---|
| PRIVATE | IO | 17600 | 17664 | 11:33:33 | 2/26/69 | EXPLICIT | F072 |
| PRIVATE | TESTPR1 | 03002 | 05012 | 17:35:00 | 1/20/69 | EXPLICIT | A105 |
| PUBLIC 0 | TESTPR2 | 10000 | 15002 | 10:47:44 | 1/15/69 | EXPLICIT | C850 |
| PUBLIC 0 | TESTPR5 | | | | | UNDEFINED | |
| PUBLIC 0 | CPDIG | | | | | UNDEFINED | |
| PUBLIC 0 | TESTPR3 | 20130 | 20145 | 9:14:36 | 12/06/68 | IMPLICIT | 1326 |

OVERWRITES WERE ENCOUNTERED.  SEE FOLLOWING PAGE.

## TABLE IV—CROSS REFERENCE TABLE

| FLAGS | LOCATION IN PROGRAM | REFERS TO SYMBOL IN PROGRAM | WITH VALUE | FOUND IN PROGRAM | WITH VALUE |
|---|---|---|---|---|---|
| | 017614 IO | IOSTOP CPDIG | 017630 | | |
| | 017600 IO | NSTORES CPDIG | 000003 | | |
| | 017630 IO | NSTORES CPDIG | 000003 | | |
| | 003002 TESTPR1 | ALPHA TESTPR2 | 010000 | | |
| | 003003 TESTPR1 | ALPHA TESTPR2 | 010000 | | |
| | 001002 TESTPR1 | ALPHA TESTPR2 | 010000 | | |
| | 003004 TESTPR1 | BETA TESTPR2 | 010001 | | |
| | 001003 TESTPR1 | BETA TESTPR2 | 010001 | | |
| | 003010 TESTPR1 | DELTA TESTPR2 | 015000 | | |
| | 003011 TESTPR1 | DELTA TESTPR2 | 015000 | | |
| | 003012 TESTPR1 | DELTA TESTPR2 | 015000 | | |
| | 003013 TESTPR1 | DELTA TESTPR2 | 015000 | | |
| | 003014 TESTPR1 | DELTA TESTPR2 | 015000 | | |
| | 003015 TESTPR1 | DELTA TESTPR2 | 015000 | | |
| | 003016 TESTPR1 | DELTA TESTPR2 | 015000 | | |
| | 003017 TESTPR1 | DELTA TESTPR2 | 015000 | | |
| | 003020 TESTPR1 | DELTA TESTPR2 | 015000 | | |
| | 003021 TESTPR1 | DELTA TESTPR2 | 015000 | | |
| | 003022 TESTPR1 | DELTA TESTPR2 | 015000 | | |
| | 003023 TESTPR1 | DELTA TESTPR2 | 015000 | | |
| | 003024 TESTPR1 | DELTA TESTPR2 | 015000 | | |
| | 003025 TESTPR1 | DELTA TESTPR2 | 015000 | | |
| | 003026 TESTPR1 | DELTA TESTPR2 | 015000 | | |
| | 003027 TESTPR1 | DELTA TESTPR2 | 015000 | | |
| | 003030 TESTPR1 | DELTA TESTPR2 | 015000 | | |
| | 003031 TESTPR1 | DELTA TESTPR2 | 015000 | | |
| | 003032 TESTPR1 | DELTA TESTPR2 | 015000 | | |
| | 003033 TESTPR1 | DELTA TESTPR2 | 015000 | | |
| | 003003 TESTPR1 | GAMMA TESTPR2 | 010005 | | |
| | 003007 TESTPR1 | GAMMA TESTPR2 | 010005 | | |
| | 005065 TESTPR1 | ZEBRA TESTPR3 | 020130 | | |
| P | 010002 TESTPR2 | ABLE TESTPR2 | | TESTPR1 | 005065 |
| P | 010005 TESTPR2 | PS1 TESTPR2 | | TESTPR1 | 005062 |
| P | 010000 TESTPR2 | PS2 TESTPR2 | | TESTPR1 | 005064 |
| P | 015000 TESTPR2 | PS2 TESTPR2 | | TESTPR1 | 005064 |
| P | 020136 TESTPR3 | ALPHA UGLY | | TESTPR2 | 010000 |
| P | 020135 TESTPR3 | GAMMA UGLY | | TESTPR2 | 010005 |
| | 020137 TESTPR3 | PS1 UGLY | 005062 | | |
| | 020140 TESTPR3 | PS2 UGLY | 005064 | | |

END OF SECO PROCESSING. ELAPSED TIME = :10

## TABLE V—SIMULATOR INPUT

```
* SIMULATOR CONTROL STATEMENTS:
                FORMAT GR,CA,@CA,AA,LINE(5,4),X0,PSYM,CSYM,LM,KM
                SYMBOLS MYPRGM
                TRACE 400
                2000 = ORID
                INOUT 3MS
LOOP            SNAPTR
AUXPRGM.MID     SNAPTR
RET             SNAPTR
CHECK           DSNAP  GR=1243
                IF CF=1 BEGIN CF=0 IF CA<120 GOTO NEXT END
                     ELSE BEGIN CF=1 IF CA<200 GOTO AGAIN END
                IF LR¬=200 CSTBL=421
                2310=1337
TEST            X0=X0+1
                IF X0=1 BEGIN TRUNK(3,7,2)=1 CA=@100|3 GOTO TEST1 END
                IF X0=2 BEGIN LINE(5,4)=176777 GOTO TEST2 END
                IF X0>=3 STOP
MIDWAY          AFTER 6US GOTO INTPRGM.INT3
TIME 14MS,10MS DIGDP ORID,6,8,2,2,2,7,9
TIME 50MS       STOP
START           MYPRGM.BGN
```

TABLE VI—SIMULATOR OUTPUT

```
TIME    SYMBOL ICC      CCT LOC  VERE   DATA                                                    PAGE     1

                                 FORMAT  GR      CA      aCA     AA      RM      LR      IM      LW      RF      TE

0000000 OPSVC.HPTT1     101656   START

0000006       PPTT1+1   101657   TRACE   101034  004000  102443  000000  000000  000000  000000  000000  000000  000000

0000078       CRDPTR+6  101342   DATA    167001  004006  010014  101037  000000  000000  000077  013670  000006  000000

0000078       CRDPTR+7  101343   TRACE   167001  004006  010014  101037  000000  000000  000077  013670  000006  000000
0000120       BPTT1+6   101664   TRACE   032002  004001  032002  101037  000000  156013  000077  013670  000003  000400
0000159 IO25M.PTC7S+4   136267   TRACE   032002  004001  032002  156113  000000  156013  000077  013670  000003  000400
0000207       ALTENT+6  136226   TRACE   000000  004001  032002  156115  000000  106400  000006  000002  000002  000400
0000237       TRCUEL+3  136244   TRACE   000000  004001  032002  156115  000000  106400  000006  000002  000000  000400
0000261       USED+2    136262   TRACE   100000  004001  032002  156115  000000  106400  000006  000002  000000  000400

0000306       PDC7X+4   136235   MXPDC:          EA      106400          LR      156115          KR      000010          IC      000000          GR      156115

0000390       PDC7X+5   136236   TRACE   156115  004001  032002  156115  000400  115670  177777  106400  000000  000000
0000402       AB254     136240   TRACE   156115  004001  032002  156115  000000  115670  177777  106400  000000  000000

0000432   101670        101670   DSNAP-
                                                 AA      156115          CA      004007          CSA     004007          EA      106400
                                                 GR      101701          BG      000037          IF      000016          IM      177777
                                                 IR      115670          IW      106400          TC      000220
                                         00000   000000  000240  000000  000000  000000  000000  000000  000000
                                         00360   101665  000000  000000  000000  032002  000000  000000  000000
                                         04000   102443  032002  003261  000000  000000  167001  010014  101701
                                         13200   000000  000000  000000  000000  100000  000000  000000  000000

0000432                  101670   GOTO    101701  004007  101701  156115  000000  115670  177777  106400  000000  000000

0000438 OPSVC.PPTT2+1    101702   TRACE   000000  004007  101701  156115  000000  115670  177777  106400  000000  000000
0000450       ISCAN+1    101076   TRACE   000000  004007  101701  000000  000000  115670  177777  106400  000000  000000
0000489 CSUE.CONET1+2    025615   TRACE   000000  004007  101701  000000  000000  000000  177777  106400  000000  000400
0000522       ZERO+2     025627   TRACE   000000  004007  101701  000000  000000  000000  177777  106400  000000  000400
0000543       GCCDZE+2   025643   TRACE   070000  004007  101701  000000  000000  000000  177777  106400  000000  000400

0000555       CK+1       025617   XTSC:           EA      070000          IR      177777          IC      000000

0038586 RCSUE.           100010   A PROGRAM STORE LOCATION THAT WAS UNLOADED OR CONTAINED DATA WAS ENOCUNTERED.  STOP WAS GENERATED:

0038586                  100010   STOP.  THE LAST SIMULATED TRANSFER WAS FROM 101772.

0038586                  100010   FINAL CONTENTS OF REGISTERS AND CALL STORE:
                                                 AA      003261          CA      004005          CF      000001          CSA     004005
                                                 EA      070153          GR      047200          HG      000037          IF      000006
                                                 LM      177403          LR      177402          RF      000007          SA      177777
                                                 TC      052525
                                         00000   000000  000240  000000  000000  000000  000000  000000  000000
                                         00350   101077  000000  000000  000000  000000  000000  000000  000000
                                         00360   101764  000000  000000  000000  032002  000000  000000  000000
                                         00640   000000  070000  000000  000000  000000  000000  000000  000000
                                         04000   102443  032002  003261  003000  000000  047200  060014  101762
                                         13200   000000  000000  000000  000000  100000  000000  000000  000000

 END OF SECC PROCESSING.  ELAPSED TIME =   :37
```

# Contributors to This Issue

Robert J. Andrews, B.S.E.E., 1952, University of Washington; Bell Telephone Laboratories, 1952—. Mr. Andrews was involved with the design of switching network circuits using gas tubes and pnpn devices. For a time he supervised studies and planning of the application of No. 1 ESS for service in the Autovon and toll networks. He is now Head of the Electronic Switching Systems Engineering Department, responsible for evaluating the economics and market potential of new design proposals and establishing service and operational requirements for local central offices. Member, IEEE.

Marshall E. Barton, A.B. (Mathematics), 1962, M.S. (Mathematics), 1964, Miami University; Bell Telephone Laboratories 1964—. Mr. Barton has worked on assemblers and other service programs supporting electronic switching development. Member, Pi Mu Epsilon, Mathematics Association of America.

Hugh J. Beuscher, B.S.E.E., 1959, University of Wisconsin; M.S.E.E, 1961, New York University; Bell Telephone Laboratories, 1959—. Mr. Beuscher was first engaged in the design of memory systems for Nike-Zeus. After this he performed circuit and logic design for No. 101 ESS. Next he worked on No. 2 ESS maintenance planning and programming. He also taught a course in electronic switching as part of a Bell Laboratories sponsored operating engineers training program. He is now working on system and logic design for No. 2 ESS. Member, IEEE, Tau Beta Pi, Eta Kappa Nu.

Thomas E. Browne, B.E.E., 1959, Manhattan College; M.S.E.E., 1961, New York University; Bell Telephone Laboratories, 1959—. Mr. Browne has worked on circuit design and device characterization in the development of the No. 101 ESS. In 1965 he was appointed Supervisor of the 2A Switch Unit Group, and in 1967 he became supervisor of a group developing magnetic and semiconductor integrated circuit memories. Member, IEEE, Eta Kappa Nu.

John Digrindakis, B.E.E., 1956, Brooklyn Polytechnic Institute; M.E.E., 1961, New York University; Bell Telephone Laboratories, 1955—. Mr. Digrindakis has worked on the development of trunk and service circuits for the crossbar tandem system, the No. 1 ESS, and more recently for the No. 2 ESS.

JOHN J. DRISCOLL, B.S.E.E., 1964, Clarkson College of Technology; M.S.E.E., 1966, Stevens Institute of Technology; Bell Telephone Laboratories, 1964—. He completed the graduate study program in 1967. During this period, he worked on logic design and trunk maintenance for the No. 2 ESS Autovon project. For the last two years, he has been engaged in service circuit design and maintenance programming for trunks and service circuits. Member, Eta Kappa Nu, Tau Beta Pi.

GEORGE E. FESSLER, B.E.E., 1949, University of Minnesota; Bell Telephone Laboratories, 1949—. He was first engaged in maintenance evaluation studies and formulation of system engineering maintenance requirements for electromechanical switching systems. He subsequently supervised a group responsible for system engineering service and maintenance requirements for a military ground-to-air visual communication system. At present, he heads a department responsible for systems engineering requirements and evaluation for operation, administration, and maintenance of switching systems. Member, Tau Beta Pi, Eta Kappa Nu.

LAIMONS FREIMANIS, B.S.E.E., 1951, M.S.E.E., 1952, Michigan State University; Bell Telephone Laboratories, 1952—. Since completing the communications development training program, he has been engaged in peripheral unit development for the ESS trials at Morris, Illinois. On No. 1 ESS he worked on central pulse distributor and related circuits. Since 1966 he has been associated with development of the peripheral decoder for No. 2 ESS. Member, Tau Beta Pi, Eta Kappa Nu, Sigma Pi Sigma.

NEIL M. HALLER, B.S.E.E., M.S.E.E., 1959; E.E., 1961, Massachusetts Institute of Technology; Bell Telephone Laboratories, 1961—. Mr. Haller has worked on maintenance programming and assemblers and other service programs supporting electronic switching development. He is Supervisor of the No. 2 ESS Service and Administration Program Group. Member, Tau Beta Pi, Sigma Xi, Eta Kappa Nu.

JOHN A. HERNDON, A.B., 1952, University of Chicago; M.S. (Physics), 1954 and Ph.D. (Physics), 1957, University of Tennessee; Bell Telephone Laboratories, 1958—. Mr. Herndon was first engaged in exploratory studies on data communication systems. In 1960 he began work on the equipment design of No. 101 ESS, a time-division elec-

tronic PBX. In 1964 he became head of the equipment department responsible for both No. 101 ESS and No. 2 ESS. Since April 1968, he has headed a department developing call processing programs for No. 2 ESS and is responsible for systems evaluation. Senior member, IEEE.

H. Robert Hofmann, B.E.E., 1957, University of Florida; M.E.E., 1962, New York University; Bell Telephone Laboratories, 1957—. Following work on the development of the ferreed switch, Mr. Hofmann was involved in the design of the ferrod used in the No. 1 and No. 2 ESS scanners. Subsequently he worked on the design of the line switching frame and the traffic data transmitter circuit for No. 1 ESS. His latest work is the circuit design of the No. 2 ESS switching network. Member, Sigma Tau.

D. Wayne Huffman, B.S.E.E., 1959, New Mexico State University; M.S.E.E., 1961, New York University; Bell Telephone Laboratories, 1959—. Mr. Huffman worked on the No. 101 ESS and No. 2 ESS in system and program design for maintenance. He is Supervisor of the No. 2 ESS Peripheral System Maintenance and Growth Group. Member, IEEE, Eta Kappa Nu.

Peter J. Kennedy, B.E.E., 1962, Manhattan College; M.S.E.E., 1963, Stanford University; Bell Telephone Laboratories, 1964—. Mr. Kennedy has been a member of the maintenance group for the No. 2 ESS since joining Bell Telephone Laboratories in 1964. He has been primarily responsible for the design of the maintenance monitor programs for the system. Member, Eta Kappa Nu.

Chester W. Lonnquist, B.E.E., 1954, the College of the City of New York; M.S., 1959, University of New Mexico; Bell Telephone Laboratories, 1954–1968. His work at Bell Telephone Laboratories included circuit and equipment design in military switching systems and the No. 101 ESS. More recently he supervised a group responsible for the equipment design of the No. 2 ESS control complex. Since 1968 he has been working in the equipment planning group at AT&TCo. Member, IEEE, Eta Kappa Nu, Tau Beta Pi.

Joseph C. Manganello, B.S.M.E., 1959, University of Pittsburgh; M.S.E.M., 1961, New York University; Bell Telephone Laboratories, 1959—. Mr. Manganello has been engaged in the development of the

Nike-Zeus and Unicom military systems, as well as a variety of electronic telephone switching systems. He supervises a group responsible for the physical design of No. 2 ESS peripheral equipment. Member, Sigma Tau, Pi Tau Sigma.

ERIC NUSSBAUM, B.S.E.E., 1955, M.S.E.E., 1956, Columbia University; Bell Telephone Laboratories, 1959—. Mr. Nussbaum has been engaged in various aspects of electronic switching system development, first on the No. 101 ESS and more recently on No. 2 ESS. Areas of activities have included circuit design, call and maintenance programming, and logic design. He is Head of the No. 2 ESS System Design Department. Member, Tau Beta Pi, Eta Kappa Nu, IEEE.

THOMAS M. QUINN, B.S.E.E., 1960, City College of New York; M.S.E.E., 1962, New York University; Bell Telephone Laboratories, 1960—. Mr. Quinn has been engaged in design work associated with the No. 101 ESS and No. 2 ESS control units. He supervises a group responsible for the No. 2 ESS control unit design and its associated maintenance and administrative programs. Member, Eta Kappa Nu.

PHILIP C. RICHARDS, B.E. (E.E.), 1957, Yale University; M.S.E.E., 1961, New York University; Bell Telephone Laboratories, 1957—. Initially Mr. Richards was engaged in exploratory studies of time division switching and memory design. He later was concerned with the design of interface circuits between the No. 101 ESS and various central offices. More recently he was involved in the initial design of the call processing program for No. 2 ESS. He now supervises a group responsible for designing trunks and service circuits, and programming administrative and peripheral maintenance routines for No. 2 ESS. Member, IEEE, Eta Kappa Nu.

GUY W. RICKER, B.S. (Mathematics), 1953, Wheaton College, M.S. (Mathematics), 1955, Rutgers University; Head of Mathematics Department, Lakewood, New Jersey, High School, 1956–1958; Associate Professor, Jersey City State College, 1958–1964; Bell Telephone Laboratories, 1964—. Mr. Ricker has worked on simulators for No. 2 ESS. Member, Pi Mu Epsilon, Sigma Xi, Phi Delta Kappa, Sigma Pi Sigma, Mathematical Association of America.

L. RALPH ROBERTS, B.S.E.E., 1957, University of Kentucky; M.S.E.E., 1959, New York University; Bell Telephone Laboratories,

1957—. Mr. Roberts was initially associated with the Bell System data processing (BSDP) project where he was concerned with logic circuit design. Joining the No. 101 ESS project in 1959, he was responsible for the development of basic call and special feature programs. Since 1965, Mr. Roberts has supervised a group responsible for development of call programs for No. 2 ESS. Member, Eta Kappa Nu, 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 supervises a group responsible for integrated circuit 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—. His early work was in the development of apparatus for No. 1 ESS, including the printed wire board connector, four-wire ferreed switch, and the distributing frames. He also supervised an equipment design group for the Unicom project and a group involved in exploratory integrated circuit techniques. He heads a department that is responsible for the physical design of both the No. 2 ESS and No. 101 ESS. Member, Sigma Tau, Phi Mu Epsilon, Sigma Xi.

A. E. Spencer, Jr., B.S.E.E., 1951, Drexel Institute of Technology; Bell Telephone Laboratories, 1951—. After assignments in designing circuits for transmission systems and a secure voice communications system, he supervised the initial design phase of the time division switch unit for No. 101 ESS. Later he supervised one of several groups responsible for planning a global military communications system. In 1962 he became Head, Data Switching Engineering Department, responsibile for establishing objectives and requirements for a store-and-forward data communications system. Since 1965 he has been Director of the Local Switching Engineering Center, responsibile for studies of the local switching field and for establishment of objectives and requirements for local switching systems. Member, Tau Beta Pi, Eta Kappa Nu, IEEE, American Association for the Advancement of Science.

ROBERT G. TAYLOR, S.B.E.E. and S.M.E.E., 1957, Massachusetts Institute of Technology; Bell Telephone Laboratories, 1957—. While enrolled in the cooperative course in electrical engineering at M.I.T., Mr. Taylor had work assignments at Bell Laboratories in the Station Apparatus, Outside Plant, and PCM Development Departments. After joining the Laboratories he worked in logic design on Bell System data processing, later working on Nike Zeus communications and on an exploratory study for an electronic PBX. He worked on the circuit development of the permanent magnet twistor store for the No. 101 ESS and supervised a group responsible for system evaluation of the No. 101 ESS. More recently on No. 2 ESS, he has supervised groups responsible for portions of the control complex design and for system test and evaluation. Member, Tau Beta Pi, Eta Kappa Nu, Sigma Xi, IEEE.

WING N. TOY, B.S.E.E., 1950, M.S.E.E., 1952, University of Illinois; Ph.D., 1969, University of Pennsylvania; Bell Telephone Laboratories, 1952—. Mr. Toy's earlier work was concerned with the L-3 carrier, data transmission, and special military systems. Since 1956, his major activities have been in the development of the No. 101 ESS and the No. 2 ESS. He is Supervisor of the System Design Group. Member, IEEE, Eta Kappa Nu.

F. S. VIGILANTE, B.S.E.E., 1957, University of California at Berkeley; M.S.E.E., 1959, New York University; Bell Telephone Laboratories, 1957—. Mr. Vigilante was first engaged in system design and programming for the No. 101 ESS. He was appointed Head of the No. 101 ESS Design Department in 1962 and later became Head, No. 2 ESS Design Department. In 1964 Mr. Vigilante received Eta Kappa Nu recognition as an Outstanding Young American Engineer. Since April 1967 he has been Director of the No. 101 and No. 2 ESS Laboratory with responsibility for the development of hardware and software for several applications of electronic switching. Member, Tau Beta Pi, Eta Kappa Nu, IEEE.

DONALD J. WADSWORTH, B.S.E.E., 1960, University of Washington; M.E.E., 1962, New York University; Bell Telephone Laboratories, 1960—. Mr. Wadsworth has worked on the circuit development and physical design of the No. 101 ESS. He supervises a group responsible for the physical design of the No. 2 ESS control complex. Member, Tau Beta Pi, Phi Beta Kappa.

JOHN E. YATES, S.B.E.E., 1960, S.M.E.E., 1962, Massachusetts Institute of Technology; Research Assistant, Massachusetts Institute of Technology, 1960–1962; Bell Telephone Laboratories 1963—. Mr. Yates designed the logic circuitry of the No. 2 ESS maintenance center. He has also been involved in writing the teletypewriter, and administration (recent change) programs. He is Supervisor of the No. 2 ESS Field Support Group. Member, Sigma Xi.