THE BELL SYSTEM TECHNICAL JOURNAL

DEVOTED TO THE SCIENTIFIC AND ENGINEERING ASPECTS OF ELECTRICAL COMMUNICATION

Volume 53 | January 1974 | Number 1

Automatic Intercept System:

Organization and Objectives

By C. J. BYRNE, W. A. WINCKELMANN, and R. M. WOLFE

(Manuscript received April 12, 1973)

The Automatic Intercept System routes calls to nonworking telephone numbers to a centralized location where a time division network under stored program control connects the calling customer to a series of prerecorded announcements. The customer is told what number he reached and, from information stored in a large-capacity disc file, is given the reason the number is not in service and, if available, the new number at which the called party may be reached. This paper describes the system objectives and organization and also serves as an introduction to the detailed papers which follow.

I. INTRODUCTION

1.1 General

Each customer's line in the telephone switching network is identified by a unique seven-digit number within a three-digit area code. The first three digits designate the local switching center or "office" in which the customer's line terminates, and the remaining four digits identify the customer within that switching center. Of the 10,000 possible telephone numbers associated with an office code, some 1000 or more are not actually assigned at any given time. Many numbers have been disconnected as customers move, other numbers have been changed, and some offices are only partially equipped, with blocks of numbers not existing in a particular office.

When a number assignment is discontinued for any reason, the number cannot immediately be reassigned. If it were, the new customer would receive calls for the person formerly assigned that number, resulting in confusion and irritation to both parties. For this reason, a discontinued number is held out-of-service until the rate of incoming calls has decreased. The period of time can be over 12 months in the case of changed business numbers. Calls to a discontinued number are intercepted in the terminating office, and the calling customer is told why the number is no longer in service and, if possible, given a new number at which the called party may be reached.

In most dial central offices, there are three categories of intercept. Calls to recently discontinued numbers where a new number for the called party is available represent one class of intercept service, which is designated "Regular Intercept." Another class of intercept, called "Machine Intercept," is the result of calls to disconnected numbers where no new number is available, numbers not actually existing in partially equipped offices, numbers which are equipped but have never been assigned, or numbers which have been discontinued for a long enough period of time to be considered unassigned.

The third class, designated "Trouble Intercept," is used when essential customer lines are temporarily out-of-service due to trouble, and alternate means of reaching those customers have been established.

1.2 Present methods

At the present time, most large dial central offices route intercepted calls to a central point over dedicated intercept trunks. The class of call is indicated by the type of supervisory signal used to seize the trunk to the central point. Machine Intercept calls are connected to a recorded announcement, which notifies the customer that he has reached a number which is not in service and asks him to stay on the line if he requires further assistance. If he does not disconnect within a timed interval, he is switched to an intercept operator. Regular Intercept calls are routed to the same point, but are immediately switched to the intercept operator without receiving a recorded announcement. Trouble Intercept calls are routed to a special switchboard location where an operator has been instructed in assisting customers to reach the called party.

When a call is switched to an intercept operator, the operator asks the customer what number he is calling, consults a printed record of nonworking numbers, and gives a verbal report to the customer of the status of the number with any available new number information.

Some small dial offices route intercepted calls to switchboard operators over general-purpose trunks. These offices do not distinguish separate classes of intercept calls, but mark all intercept calls with a burst of audio tone derived from a ringing machine. The operator may serve the customer by reference to an intercept bulletin or may forward the call to an intercept bureau over a dedicated trunk.

Studies of this present method of handling intercept indicated that the development of more automatic means would provide better customer service and would also result in cost savings.

From a cost standpoint, approximately 3500 Bell System operators are required for the time-consuming process of answering calls, searching the intercept record, and giving verbal reports. The record of intercepted numbers, because of the volume of changes, must be reprinted and delivered to each operator position daily. Entries in the record are subject to a time lag as well as to clerical errors.

From the customer's viewpoint, the nonworking number announcement may cause confusion. Customers intending to dial a working number occasionally make an error in dialing. With about 15 percent of the available numbers on intercept at any given time on the average, the customer has a good chance of reaching intercept. If the number is on Regular Intercept, the customer gives the operator the working number he intended to dial. The operator finds no entry in the record for that number and reports "That is a working number. Will you dial it again, please?" If the number dialed happens to be on Machine Intercept, the customer receives an announcement telling him that he has reached a number which is not in service. If he does not disconnect, an operator then asks for the number and tells him it is a working number, contradicting the previous machine announcement.

II. OBJECTIVES

The objectives of the Automatic Intercept System (AIS) are:

(*i*) To improve customer service while saving operating costs by automatically identifying the number actually reached and giving the necessary information to the customer by means of an automated announcement.

The recorded announcement should give all necessary information to the customer and eliminate operator intervention on most calls. In addition to the called number, it should include the status of that number (changed, disconnected, etc.), and the new number with the new area code or new geographic location when required. Operator assistance should be needed only when a customer is not satisfied after hearing an announcement or where calls cannot be handled by the automated announcement.

Figure 1 shows some typical recorded announcements as provided in the AIS. The first announcement sequence is used for a disconnected number with a transfer of calls to a new number. The second announcement is used for a call to an unassigned number. The home area code is included optionally in case a distant customer has reached this number by dialing a wrong area code. The third sequence is used for a number change with a new number and includes the geographic area in which the new number is located.

(*ii*) To save the cost of printing intercept records by providing a mechanized data base.

The means of maintaining the data base should have the capability for rapid record updating, with immediate online insertion, deletion, and verification of record entries.

III. OVERALL SYSTEM CONFIGURATION

Figure 2 is a diagram of a typical Automatic Intercept System. Local terminating offices are designated "ANI" (for Automatic Number Identification) where the local office has been modified to

```
THE NUMBER YOU HAVE REACHED, 642 54 31, HAS BEEN DIS-
CONNECTED.  CALLS ARE BEING TAKEN BY 747 36 45.  PLEASE
MAKE A NOTE OF IT - 642 54 31 HAS BEEN DISCONNECTED.  CALLS
ARE BEING TAKEN BY 747 36 45.  IF YOU NEED ASSISTANCE, YOU
MAY STAY ON THE LINE AND AN OPERATOR WILL ANSWER.

THE NUMBER YOU HAVE REACHED, 368 11 HUNDRED, IS NOT IN
SERVICE IN THE 201 AREA.  PLEASE CHECK THE NUMBER AND DIAL
AGAIN.  368 11 HUNDRED IS NOT IN SERVICE IN THE 201 AREA.
IF YOU NEED ASSISTANCE, YOU MAY STAY ON THE LINE AND AN
OPERATOR WILL ANSWER.

THE NUMBER YOU HAVE REACHED, 432 98 72, HAS BEEN CHANGED.
THE NEW NUMBER IS 741 32 32 IN THE RED BANK AREA.  432 98
72 HAS BEEN CHANGED.  THE NEW NUMBER IS 741 32 32 IN THE
RED BANK AREA.  IF YOU NEED ASSISTANCE, YOU MAY STAY ON
THE LINE AND AN OPERATOR WILL ANSWER.
```

Fig. 1—Typical Automatic Intercept announcements.

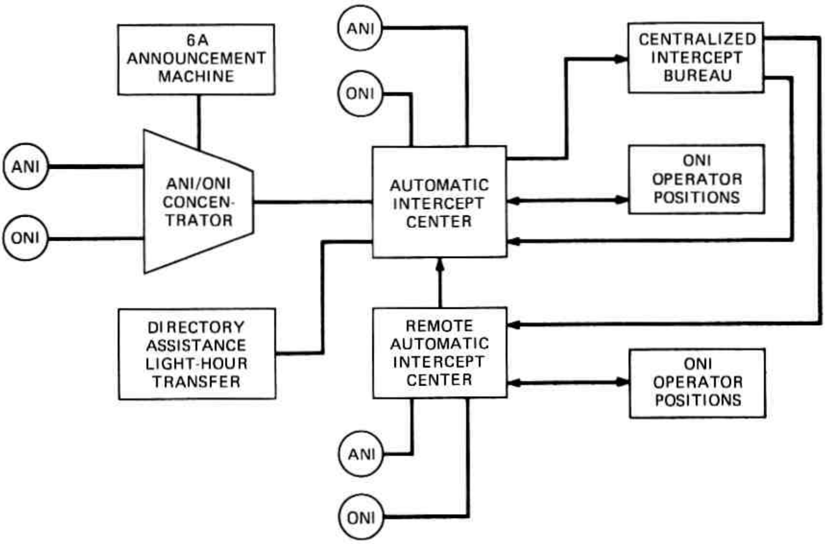


Fig. 2—Overall system configuration of an Automatic Intercept System.

identify and outpulse to the Automatic Intercept Center (AIC) the seven-digit called number with a one-digit prefix indicating the class of intercept (blank, regular, or trouble). Offices designated "ONI" (for Operator Number Identification) are not so modified. Calls from these offices are switched through the AIC to an ONI operator who interrogates the calling customer and keys the called number into the AIC, after which the call is handled as if the called number had been outpulsed from the local office.

The following types of offices can be modified for ANI operation:

No. 1 Electronic Switching System
No. 2 Electronic Switching System
No. 1 Crossbar Dial Offices
No. 5 Crossbar Dial Offices
Step-by-step Dial Offices equipped with certain Automatic Number Identification features used for Automatic Message Accounting
Panel Dial Offices equipped with Automatic Number Identification features used for Automatic Message Accounting.

Modification of electromechanical local offices for ANI is a major cost factor in the Automatic Intercept System, but the cost is recoverable through operator savings. ESS offices are modified simply by exercising a feature option in the generic program. The cost of providing ANI features for all types of new office installations is relatively small.

The AIS is made compatible with the three-class ONI operation to allow installation and cutover of an AIS with unmodified local offices. This provides for early realization of operator savings while permitting orderly scheduling of modification activity at the many local offices connected with each AIC. It also allows offices which are scheduled to be replaced within a few years to use AIS service without modification.

In addition to the ANI and three-class ONI connections to the AIC, one-class ONI operation is provided, which treats all seizures as "regular" ONI intercept calls. This feature is used for calls from local offices not arranged to indicate class of intercept and for operator-forwarded calls.

Calls may be routed to the AIC via direct two-wire trunks or carrier facilities. Trunks from a group of offices located at a considerable distance from the AIC can be concentrated to save on trunk costs. Since intercept traffic from an individual local office is of the order of 0.5 erlang and since several intercept trunks must be provided for an adequate grade of service, allowing for maintenance outage, the average trunk usage is of the order of 20 percent. Concentrating trunks from several local offices results in savings in trunk costs from the concentrator to the AIC.

The No. 23 concentrator (with modifications) can serve a mix of ANI and ONI local offices. It can be equipped for a maximum of 140 incoming trunks and 40 outgoing trunks. Machine Intercept calls can be routed to an announcement machine at the concentrator, with calls from customers who wait through the announcements forwarded to the AIC for operator handling. This feature may be used to deload the AIC network by diverting "machine" ONI calls.

An AIC may also handle Directory Assistance (411) calls transferred from Directory Assistance bureaus during periods of light traffic. These calls are routed to Centralized Intercept Bureau (CIB) consoles with a displayed indication of an incoming Directory Assistance call. Shelf space is provided at the console for directory records. Directory Assistance calls must be routed through dedicated Directory Assistance trunks and cannot be mixed with intercept calls.

Calls may route to a "Home" AIC, which has associated CIB operator consoles, or to up to four "Remote" AIC's which contain all facilities except CIB consoles. Calls in a Remote AIC requiring operator assistance or light-hour Directory Assistance are transferred

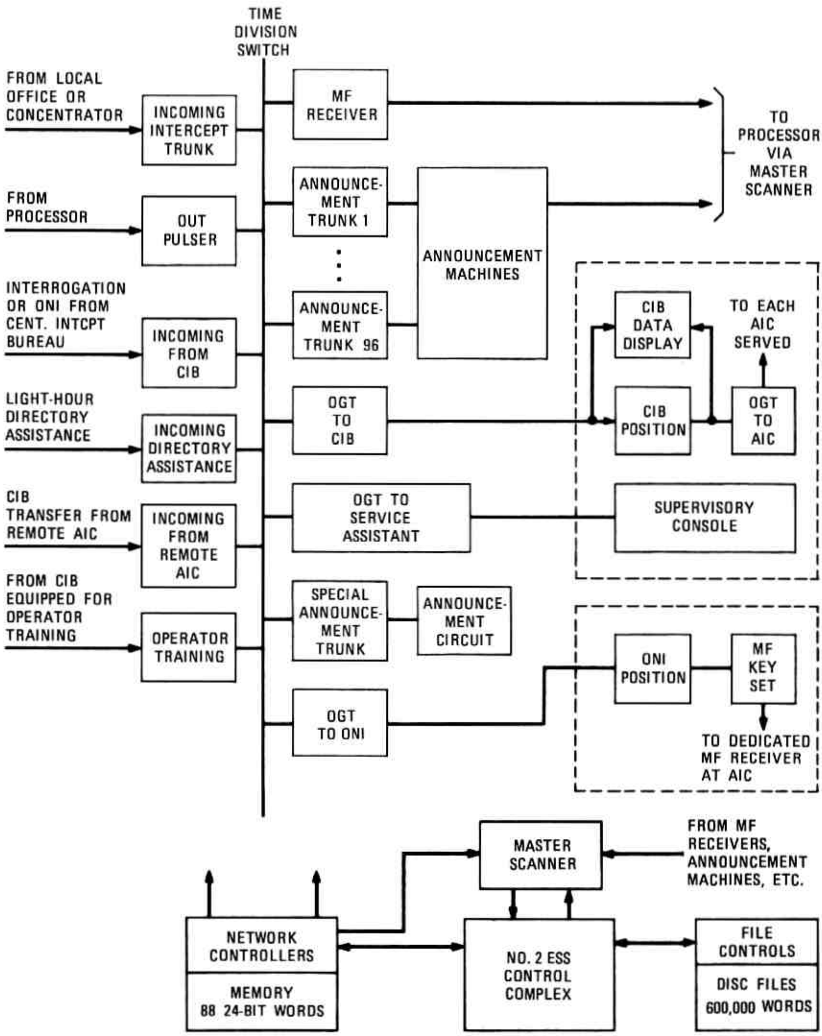


Fig. 3—Block diagram of an Automatic Intercept Center and operator positions.

to the CIB through the Home AIC on a tandem basis. CIB operators have access to each AIC served and may interrogate any AIC using a position keyset to receive a display or announcement based on the information stored in the data file for any seven-digit number.

Calls requiring the ONI function may be routed to special ONI operator positions or may be routed to the CIB positions. Installations with relatively little ONI traffic would find it uneconomic to provide two separate operator teams. For example, an installation which can be served by six ONI operators and six CIB operators can be served by ten operators at CIB positions because of the increase in efficiency at the larger team size. However, the savings in operator costs are partly offset by the cost of the additional CIB positions, which are equipped with an automatic display capability which is not needed for ONI positions.

When ONI positions are equipped, ONI calls in a Home AIC can be routed to the CIB positions during light-traffic periods. A Remote AIC handling ONI traffic must be equipped with ONI positions and cannot route calls to the Home AIC for the ONI operation.

IV. AUTOMATIC INTERCEPT CENTER

The configuration of the AIC is shown in Fig. 3. The AIC consists primarily of a control complex, Master Scanner, switching equipment, announcement machines, and disc files.

4.1 Control complex

The control complex is identical to that used in the No. 2 Electronic Switching System.[1] It is composed of two Control Units and a Maintenance and Administration Center. The Maintenance and Administration Center is described in Section VII. Each Control Unit includes a Program Control, Program Store, Call Store, and Input-Output Control (see Fig. 4).

4.1.1 Program Control

The Program Control is a discrete-component solid state processor with a 3-microsecond cycle time. The Program Control reads and executes 22-bit instructions stored in the Program Store, a read-only permanent magnet twistor memory. Each read requires 6 microseconds. The Call Store is the temporary memory. It is a destructive-readout ferrite sheet memory using 16-bit words.

4.1.2 Input-Output Control

The Input-Output Control shares Call Store access with the Program Control and performs time-critical functions such as digit receiving and data outpulsing. It also provides an interface with peripheral units. The following types of interfaces are provided:

(*i*) A Peripheral Unit Address Bus, shared by all peripheral units, provides for 36-bit (parallel) commands.

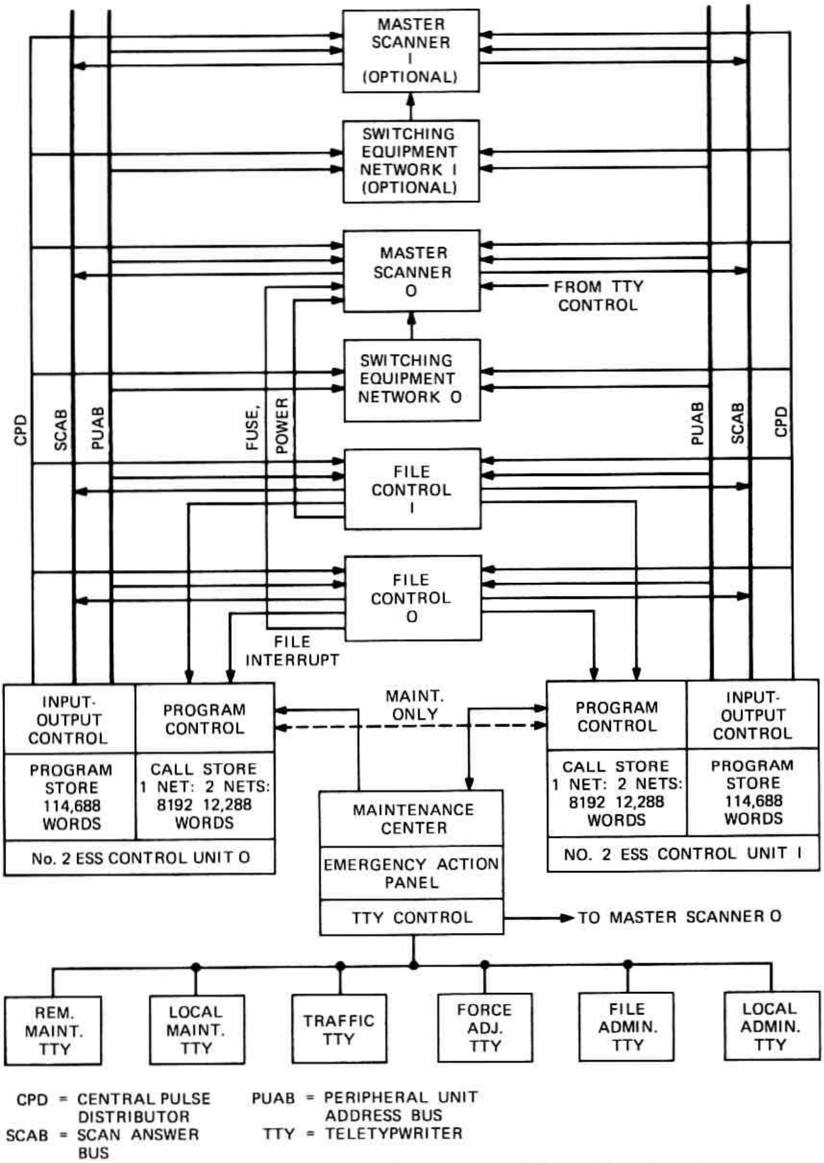


Fig. 4—Processor configuration and interfaces with peripheral equipment.

(*ii*) The Central Pulse Distributor provides communication to peripheral units over dedicated twisted-wire pairs. This device alerts peripheral units to accept commands from the Peripheral

Unit Address Bus. It also transmits single pulses or a short series of coded pulses to peripheral units without use of the Peripheral Unit Address Bus.

(*iii*) The Scan Answer Bus allows the Master Scanner and disc files to reply to a Peripheral Unit Address Bus command. It provides a 16-bit (parallel) word.

4.1.3 Input-output interrupt

An input-output interrupt occurs every 25 milliseconds to allow time-critical programs, primarily associated with peripheral unit access and control, to run.

4.1.4 File interrupts

One interrupt which is initiated by an external signal is provided for each disc file to provide for receiving a block of information directly from the rotating file.

4.2 Master Scanner

The Master Scanner is an input device which allows the processor to interrogate the dc-current-level state of up to 1024 wire pairs by means of ferrods. It carries all data sent from the switching equipment to the processor (including supervisory reports and digits received by multifrequency or *Touch-Tone*® signaling) and all teletypewriter input messages. It also serves a number of miscellaneous functions.

4.3 Switching equipment

The switching equipment consists of a time division switching network, trunk circuits, multifrequency (MF) receivers, data outpulsers, the service observing terminal, and a trunk test frame.[2] The time division switch, derived from the No. 101 Electronic PBX 2A Switch Unit, was selected because its nearly 100-percent solid state design is well-suited to the high switching rate required for announcement synthesis. Most of the calls in progress require connection changes within a 25-millisecond interval every 0.5 second. The switching network connects the incoming trunk to MF receivers, operators, or announcement tracks as required. Each switching network can accommodate 64 simultaneous connections.

The AIC can be equipped with either one or two switching networks, depending on the traffic required. The switching equipment is so designed that trunks and service circuits can be added, deleted, or changed without wiring changes, within frame capacity limits. This means that these changes are completely under Operating Company control.

4.4 Announcement machines

The announcement machines store prerecorded words and phrases on a rotating drum coated with a magnetic material. The audio signal on each track is read and amplified in a separate channel and made available at the switching network.

Announcements are synthesized by switching the customer's trunk from track to track as required. The number of tracks provided is 96; 48 tracks contain words and phrases used for standard intercept announcements and 48 tracks are available for geographic locations recorded specially for each installation. Alternately, the letters of the alphabet which appear on a telephone dial can be recorded on 24 of the last 48 tracks. This feature allows synthesis of telephone number announcements with the alphabet central office codes, where such designations are still in use.

The design of announcements must consider the structure of the announcement machine. An announcement is divided into 1.5-second modules. Each 1.5-second module is subdivided into 0.5-second modules. Stock phrases such as "The number you have reached" or "is temporarily out-of-service" are rendered in 1.5-second intervals and can flow smoothly across 0.5-second boundaries. Individual words (including numbers and letters) are recorded in 0.5-second intervals. Once a long (1.5-second) phrase is used in a given message, all further long phrases must be spaced from it by an integral multiple of 1.5 seconds.

Certain rules must be observed to preserve the naturalness of the message. Among these are:

(*i*) Inflections must be appropriate. For example, all numbers are recorded in two inflections, neutral and falling. The falling inflection is used for the third and the last digit of a telephone number.

(*ii*) Pauses must be inserted to break up the word and phrase units into natural thought groups. They also serve to emphasize adjacent words. Pauses are provided by connecting the customer's trunk to a quiet termination.

The timing of the recorded messages is such that only 50 milliseconds of quiet is provided at the beginning of each track to allow for switching. A clock signal is recorded on the drum and signals the start of a 0.5- or 1.5-second interval to the processor by way of the Master Scanner. During the next input-output interrupt (within approximately 25 milliseconds), connections to the next track in the announcement sequence are made. The design of the switching network is such that the change in track is made in between time division samples, so that no audible click is present.

4.5 Disc files

A record of each equipped intercepted number in the local offices served by an AIC is maintained in duplicated magnetic disc files.[3] This record includes the seven-digit called number, a two-digit status indication, a count of calls to the number, and, when required, a seven-digit new number and three-digit new numbering plan area (NPA) or geographic location.

A disc file is consulted by the processor on all calls of the Regular Intercept class, and the reply from the file is used by the stored program to connect the proper sequence of phrases and digits to the incoming trunk. Delays, transfer to an operator, or connection to a locally recordable special announcement may be included in a call handling sequence.

On calls to numbers on Machine Intercept, Trouble Intercept calls, or on failure to identify the called number (as indicated by a prefixed digit transmitted by the local office) the processor determines the announcement sequence or other method of handling the call without reference to the disc file.

Disc file updating facilities are provided by teletypewriter and data link input with a verification readout to the updating channel.[4] The updating channels also have the ability to obtain a readout of stored information or call counts of an individual number or on all numbers in any local office. The file system also has the ability to match information stored in the two files with a printout of mismatches, or to transfer all stored information from one file to the other.

4.6 Teletypewriters

Operational human interaction' with an AIC is primarily through teletypewriter channels. The following channels are provided:

(*i*) Two maintenance control channels, one local and one remote.
(*ii*) A traffic administration machine for control of entries in the disc file intercept record by keyboard or punched tape.
(*iii*) A plant administration machine for initialization, control, and backup of nongeneric data on the disc file.

(*iv*) Two receive-only machines which print out data on traffic volumes and equipment usage for engineering information and operator force adjustment.

4.7 Equipment

All equipment is mounted on No. 1 ESS type frames. Every effort was made in the development of the AIC to minimize line engineering, manufacturing, installation, and maintenance costs. To achieve this, maximum use was made of frames and apparatus previously designed for other systems, options were kept to a minimum, most interframe cabling was connectorized, the use of plug-in apparatus was emphasized, and fixed floor plans were developed.

V. OPERATOR POSITIONS

CIB operator positions provide a visual display to the operator when a customer is connected. The display presents all relevant information on the history of the call. This saves both operator and customer time and allows the operator to concentrate on the customer's problem rather than asking for information already available.

The information for the display is passed from the processor to the operator position over the position's dedicated outgoing trunk by means of frequency-shift keying. A data outpulser is connected to the trunk through the switching network in order to send the display data. The data outpulser connection is then taken down and the incoming trunk is connected to the operator position.

When the AIC network is used as a tandem switching point to switch calls to the centralized operator positions from Remote Automatic Intercept Centers, the data outpulsing is done by the AIC which first received the call and includes the identity of that AIC.

Operators at CIB positions can access a Home or any associated Remote AIC. They can request a display or announcement on any number by keying the digits over trunks to any associated Remote AIC.

ONI operator positions provide a talking connection to an audio trunk and a keyset equipped for MF pulsing over a separate trunk to a dedicated MF receiver at the AIC.

Both ONI and CIB positions can be remote from the AIC. Carrier facilities may be used if the distances are great.

VI. STORED PROGRAM

The program which controls the processor uses approximately 100,000 words of storage. Of these, 21,500 words are used in common by the AIS and by the No. 2 ESS Local Switching Office. The common code is primarily concerned with processor maintenance, teletypewriters, and utility functions.

The AIS program is totally generic; that is, the program in any installation can provide all optional features and no information which is particular to an individual installation appears in the Program Store. Nongeneric data, such as the assignments and types of individual trunk circuits (translation data) and lists of options, are stored in the Call Store. Since the Call Store is a volatile medium, nongeneric data are backed up on the disc file.[5]

This method of storing nongeneric data allows the administration of nongeneric data to be accomplished locally, by means of teletypewriter messages. There is no requirement for writing permanent magnet twistor cards locally or for a centralized office data assembler facility. This results in a savings in equipment and operational manpower over storing nongeneric data in the permanent magnet twistor memory.

Nearly all programs run at base level. Functional groups of programs are run in sequence until each group has been accessed (through its monitor); then the sequence is restarted. Each sequence is called a main program loop.[5] The time required for a single main program loop can run from about 12 milliseconds to the order of 200 milliseconds, depending on the work to be done.

Time-critical programs (primarily concerned with input-output functions) run in interrupts. Each interrupt stops current base-level work and restarts it when the interrupt program completes its work.

All transient data, including records of calls in progress, are kept in Call Store. Each call is assigned a Call-In-Progress Register (CIPR), which is an eight-word block of Call Store. Nearly all pertinent information about a call is kept in the CIPR. CIPR's in use are examined by call processing programs once each base-level loop to see if further action on the call is required. Counts of calls handled and usage counts are kept in the Call Store and printed on the traffic teletypewriter at 15-minute or half-hourly intervals.

VII. MAINTENANCE FEATURES

Duplicate units are provided for all service-critical components except the switching network, which has a soft-failure mode, in that some worst-case single faults can reduce the traffic capacity by one-half of one network.

The duplicate processors normally run in synchronism, with one online and the other offline.[1] The outputs of the offline processor are inhibited from activating any external equipment, but both units are executing the same sequence of instructions and both Call Stores contain identical information. On each write into Call Store the data being written are matched in the maintenance center. A mismatch causes error detection tests to be run on the online unit. If these tests fail, the unit is switched offline and the other unit is switched online and performs diagnostic tests on the now-offline processor in addition to taking over operation of the system. If the initially online unit passes its detection tests, it runs a diagnostic on the offline unit.

A processor switch may also be caused by other error checking circuits built into the processor.

Single hard faults in the peripheral equipment are rapidly detected by checks in call processing programs, periodic checks of circuit-detected trouble signals, and maintenance exercise programs. Once a fault has been detected, working mode programs isolate the affected unit and remove it from service. Diagnostic programs isolate the trouble to a few circuit packs and print a maintenance teletypewriter message which indicates the packs to be changed. This sequence of actions is performed in short program segments (usually less than 5 milliseconds in elapsed time) run once per main program loop, so that call processing is unaffected after the working mode action.

The maintenance center provides a means for a maintenance man to communicate with the system. It includes lamp displays of registers and equipment status, control keys, and a teletypewriter for two-way communication with the processors. The local maintenance teletypewriter is backed up by a remote maintenance teletypewriter to allow operation of the system from a distant location.

A daily or hourly count of possible trouble conditions, along with base traffic counts useful in normalizing trouble rates, is printed on the plant maintenance teletypewriters.

VIII. ENGINEERING CONSIDERATIONS

Plans for early installations indicate that the capacity of Automatic Intercept Systems will usually be limited by the number of time slots available to establish connections. A study of the AIS priority queue structure predicts that the capacity of each 64-time-slot network is approximately 1900 busy-hour CCS. The capacity can vary about ± 5 percent from this figure depending on the traffic mix.

Assuming that an installation is network-limited, the number of calls which can be handled by the system depends on the average network holding time per call. The holding time in a Home AIC with no associated Remote AIC and no Directory Assistance traffic depends primarily on the following factors:

(*i*) Proportions of ANI and ONI calls.
(*ii*) Proportions of Regular Intercept and Blank Number calls.
(*iii*) Proportions of Regular Intercept calls which have new number information to be announced.
(*iv*) Handling of ONI calls on CIB or ONI positions (CIB positions require a second network connection for keying).
(*v*) Quality of the intercept data base (errors require operator handling).

Studies of the traffic at the first two installations indicate that the average holding time for AIS installations can range from 15 to 35 seconds, depending on the traffic mix. Assuming an average network holding time of 20 seconds (which is conservative for predominantly ANI traffic), the capacity of a two-network AIC is 19,000 calls per busy hour. Clearly, capacity estimates for future installations must be based on careful counts of traffic by types, with an allowance for reserve capacity which reflects any uncertainty in the number and mix of calls as well as peaking and growth factors.

Measurements of processor usage under live traffic confirms that there will be adequate processor capacity at full load.

When an AIC is loaded to capacity, calls enter a priority queuing structure to assure the most efficient use of resources. When the queue capacity is exhausted, calls are held in the Call-In-Progress Registers and served in random order. When the Call-In-Progress Registers are exhausted, requests for service are not acknowledged until registers become available. Local offices are arranged to time out and give reorder tone to customers when excessive delays are encountered. This sequence of overload provisions assures that throughput will not decrease due to overload.

IX. SAVINGS IN NUMBERS OF OPERATORS

A major objective of the Automatic Intercept System is to save operators. Current manual systems require an operator to perform the following functions:

(*i*) Ask the customer what number is being called.
(*ii*) Look up the number in a bulletin.

(*iii*) Tell the customer why the number is on intercept and give a new number if available.

(*iv*) If the customer feels the information is incorrect, fill out a trouble report or transfer the call to a service assistant.

The AIS automates steps (*i*), (*ii*), and (*iii*) above for ANI calls and automates (*ii*) and (*iii*) for ONI calls. In addition, many errors in data base are rapidly and automatically reported to the clerical group responsible for intercept records, reducing the work in step (*iv*).

Figure 5 shows the number of operators required to serve intercept traffic as a function of the number of calls per busy hour. In an all-ANI system with 10,000 busy-hour calls, 85 percent of the operators are saved. In an all-ONI system, 50 percent of the operators are saved.

X. STATUS OF AIS INSTALLATIONS

The first AIS installation was cut over for full-time operation at Hempstead, N. Y., on September 13, 1970. By October 18, 1970, the system was handling 17 ANI and 8 ONI local office entities. On November 8, 1970, intercept traffic for the remainder of Nassau

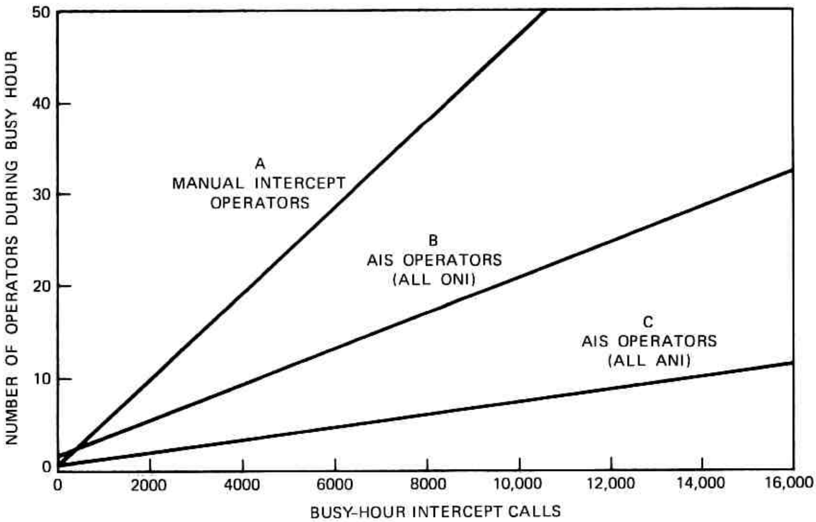


Fig. 5—Number of intercept operators as a function of traffic: curve A, Manual Intercept operators; curve B, AIS operators for an all-ONI system (includes CIB operators and ONI operators at separate positions); curve C, AIS operators for an all-ANI system. The assumptions used in generating these curves are: 40 percent of all intercept calls are of the Machine Intercept class; 10 percent of all calls to manual or ONI operators disconnect while waiting; 5 percent of all intercept calls wait for a CIB operator.

County was transferred to the Hempstead AIS for a total of 33 ANI and 8 ONI entities. At this point, an 18-position Manual Intercept Board was retired.

Between March 7 and April 4, 1971, as local office modifications were completed, all of the Suffolk County traffic was transferred to AIS and a 10-position Manual Board was retired. This AIS in its present configuration serves 3 ONI entities and 71 ANI entities, of which 26 are routed through three concentrators, and also handles Blank Number PBX direct inward dialing for three crossbar tandem units.

The system handles some 90,000 incoming calls on business days, with 250,000 file entries, and 2500 daily changes on disc file intercept records. A force of seven operators handles the busy-hour traffic (which is 97 percent ANI).

The second AIS installation was cut over in Minneapolis, Minn., in October 1971. The third and fourth systems (each equipped with two switching networks) went into service in Cleveland, Ohio, and Manhattan, N. Y., in February and March of 1973. The Indianapolis, Ind., AIS was placed into service in July 1973, and the Baltimore, Md., installation started service in October 1973. Systems are being installed at Newark, N. J., Hammonton, N. J., Miami, Fla., San Francisco, Cal., Washington, D.C., Boston, Mass., Detroit, Mich., Philadelphia, Pa., and other major metropolitan areas.

XI. ACKNOWLEDGMENTS

The design and implementation of the Automatic Intercept System represent the contributions of many groups and individuals at the American Telephone and Telegraph Company, Western Electric Company, and Bell Laboratories. The suggestions of the New York Telephone Company based on their experience with the system also provided valuable information in the design process.

REFERENCES

1. T. E. Browne, T. M. Quinn, W. N. Toy, and J. E. Yates, "[No. 2ESS] Control Unit System," B.S.T.J., *48*, No. 8 (October 1969), pp. 2619–2668.
2. P. J. Brendel, W. K. Comella, R. N. Markson, P. J. Moylan, and J. Orost, "Automatic Intercept System: Peripheral Circuits," B.S.T.J., this issue, pp. 71–106.
3. J. W. Hopkins, P. D. Hunter, R. E. Machol, J. J. DiSalvo, and R. J. Piereth, "Automatic Intercept System: File Subsystem," B.S.T.J., this issue, pp. 107–132.
4. J. H. Carran, K. E. Greisen, W. G. Hall, and D. J. Wells, "Automatic Intercept System: Administering the Intercept Data Base," B.S.T.J., this issue, pp. 133–153.
5. H. Cohen, D. E. Confalone, B. D. Wagner, and W. W. Wood, "Automatic Intercept System: Operational Programs," B.S.T.J., this issue, pp. 19–69.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 53, No. 1, January 1974
Printed in U.S.A.

Automatic Intercept System:

Operational Programs

By H. COHEN, D. E. CONFALONE, B. D. WAGNER, and W. W. WOOD

(Manuscript received May 16, 1973)

The Automatic Intercept System Operational Programs provide the logic for processing calls served by the system. These programs also perform administrative and software correction and recovery functions. Described are program organization, use of temporary memory, and details of call processing.

I. INTRODUCTION

The stored program for the Automatic Intercept System (AIS)[1] directs the operation of equipment which processes intercept calls. The system is designed to meet stringent operational requirements similar to those imposed on other stored program switching systems, such as the No. 1 Electronic Switching System (ESS).[2] AIS provides rapid, reliable, and economical intercept service and supports simple administrative and maintenance procedures.

This paper describes the programs that process calls as well as some of the administrative and maintenance programs. Companion papers cover the remaining parts of the AIS program.[3–5]

Intercept calls differ in several significant ways from calls served by typical local or toll switching centers. These differences have been exploited wherever possible to simplify the system design. Most influential among these factors are the following:

(*i*) The variety of services and number of options required are relatively limited.
(*ii*) There are no subscriber-to-subscriber connections.
(*iii*) Calls have short holding times (the average call is less than

30 seconds in duration, while over 90 percent of the calls to AIS last less than one minute).

(*iv*) Connections to callers receiving announcements are changed at a high rate (every 0.5 or 1.5 seconds).

(*v*) Several hundred thousand intercepted lines (in an area of several million lines) can be served by an AIS.

These factors have most influenced the system design in three areas: use of a standard generic program, novel methods of storing and administering installation-dependent data, and wide use of autonomous circuits. These areas are considered in the following paragraphs.

1.1 Generic program

Intercept service is much more limited in scope than the service provided by a typical telephone switching office. Thus, a single *generic* program can serve all sites. This approach is efficient because the small number of options do not significantly penalize installations where certain options are not used. Use of a single generic program is also desirable because it simplifies testing and support of programs and minimizes the possibility of undiscovered incompatibilities.

1.2 Installation-dependent data

Installation-dependent data consist of:

(*i*) intercept number records, and

(*ii*) records describing installation options.

The intercept number records are stored on disc files.[5] These memory devices provide an economical fast storage medium for the large volume of data required. These data do not describe the way an installation is equipped, but merely describe intercepted lines in local offices served by an AIS. Thus, such records are not considered installation options.

The remaining installation-dependent data, known as *nongeneric data*, describe the way a particular AIS is equipped and operates. Such data include trunk and announcement translation information, equipment and feature options installed, etc. Backup records of nongeneric data are stored on the disc files, but the working record is kept in the temporary memory, *call store*, for rapid access by the operational programs.

1.3 Autonomous circuits

In order to minimize repetitive processing operations and maximize system call capacity, a high degree of autonomous circuit operation is provided. Among the circuits with some autonomous operation are:

(*i*) a time-division switching network, which allows simple path selection and network control and expedites frequent connection changes;[3]

(*ii*) control unit circuits which perform digit receiving and data sending;

(*iii*) associative match circuitry in the disc file control for intercept number lookup;

(*iv*) a trunk scanner associated with the time-division switching network; and

(*v*) multifrequency receiver trunk circuits which perform supervisory wink signaling.

II. PROGRAM ORGANIZATION

The AIS program operates in real time to process intercept calls. The program is structured to perform each task at an appropriate rate so that response times to input stimuli are minimized and processing capability is maximized.

Parts of the AIS generic program were developed for use by both AIS and the No. 2 Electronic Switching System.[6] Since these systems use the same control complex, programs which perform functions such as control unit maintenance or teletypewriter input-output are substantially application-independent and thus can be used in common by both systems. The remaining programs, used solely for the AIS application, include all call processing and administrative programs as well as application-dependent maintenance programs. The total program consists of almost 100,000 program store words. Figure 1 illustrates the breakdown between commonly used and application programs as well as the functional division.

In addition to certain operational programs, several *service programs* are shared by the AIS and No. 2 ESS developments.[7] These progams, written for a general-purpose computer, are used to prepare and test operational programs and to produce program documentation. Among the service programs are a macro assembler, a linking loader, and a control unit simulator.

2.1 Main program loop

The *main program loop* is a series of programs executed in a repeating cycle at *base level*, that is, when no interrupts are in effect. Programs in this loop (Fig. 2) perform five functions at the same frequency: call processing, maintenance, teletypewriter, call store audits, and file

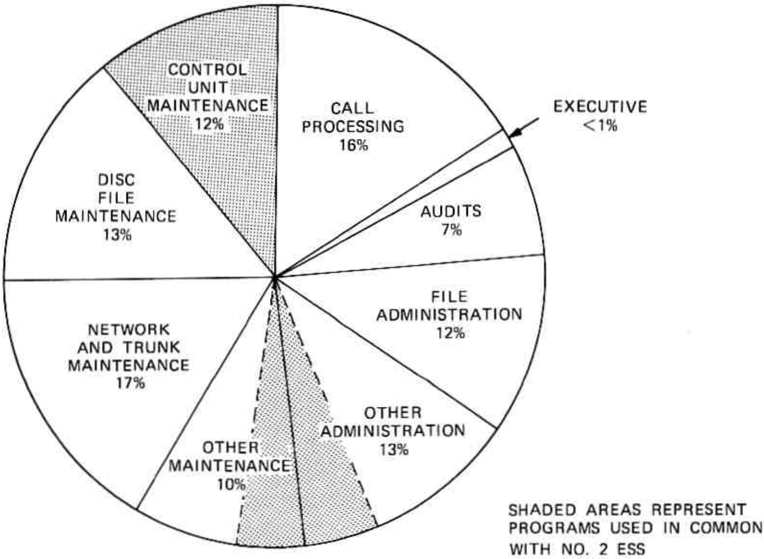


Fig. 1—Functional division of AIS programs.

administration. Following is a description of these programs and their control.

2.1.1 Executive control program

Sequencing of base level programs is performed by the *executive control* program. Programs which are directly subordinate to executive control are known as *monitor* programs. Each monitor program normally is entered once per main program loop. However, the executive may bypass one or more monitors during a particular main program loop based on indicators of certain abnormal system activity. These abnormal modes of operation include:

(*i*) reloading of nongeneric data from a tape backup (Section 7.2).* In this mode, call processing activity is suspended and the call processing monitor is not entered and

(*ii*) system initialization, when only call store audit (Section 5.4) and file administration[5] programs are entered to initialize nongeneric call store data.

* Note that should disc file intercept number records be lost, they are reloaded from a backup record. During this period, call processing continues and system programs are executed normally.

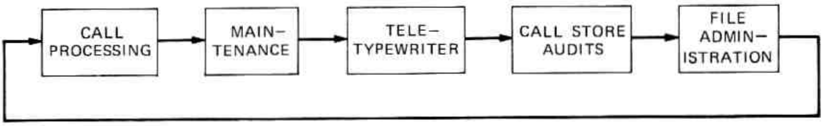


Fig. 2.—Main program loop.

Additionally, the executive schedules programs for execution based on time of day and elapsed time. To perform these services, the executive maintains a highly accurate time-of-day clock in call store memory.

2.1.2 Monitor programs

The call processing monitor (Section IV) responds to requests for service detected by interrupt level programs. It also controls the execution of client task programs which advance calls through various stages. The teletypewriter monitor controls the receiving and sending of data over the six AIS teletypewriter channels. Character input and output functions are performed at an interrupt level. The call store audit monitor (Section V) controls execution of audit programs. The file administration monitor[5] controls programs which perform disc file access and which administer the intercepted number data base.

The maintenance monitor controls all system maintenance activity. Based on the maintenance states of system equipment and requests for maintenance action, this monitor selects for execution those maintenance jobs immediately vital to continued operation of the system. Other functions of the maintenance monitor include:

(*i*) reacting to changes of status of the control units;

(*ii*) scheduling maintenance action on all system equipment;

(*iii*) responding to maintenance and utility requests input manually via a maintenance teletypewriter or the maintenance center control panel;

(*iv*) testing for and reacting to abnormal conditions indicated by the states of alarm-indicating master scanner ferrods; and

(*v*) initiating recovery action after detecting that the main program loop is not cycling properly.

The scheduling function of the maintenance monitor is complex and relies on a preselected set of priorities associated with each class of maintenance test actions. The relative priorities of the classes have been determined by considering the consequences of deferring each class. For example, diagnostic testing of a control unit resulting from an error detected by a check circuit is given priority over daily routine testing of the switching network.

The maintenance scheduler is entered once per main program loop. During each entry, the scheduler permits a portion of a maintenance test to be executed. Maintenance tests are segmented since, in general, they would require hundreds of milliseconds if executed continuously. The maintenance scheduler generally permits only a single maintenance function to be executed until it is completed to prevent interference between maintenance programs which may temporarily leave equipment in an arbitrary state.

Flexibility is afforded in maintenance scheduling by prematurely terminating, or *aborting*, a maintenance function which may be in progress when a higher priority function is requested. An aborted function may be resumed when all higher priority functions have been completed; also, a new request is held until all higher priority requests have been honored. In addition, the scheduler can disallow particular functions when indicators do not permit that function to be executed. For example, when the control units are operating in synchronism, diagnostic testing of the offline control unit is not allowed.

Manual requests via teletypewriter for maintenance tests permit the requestor to specify that the test be performed in one of three modes: once, repetitively, or in *step mode* (once on each operation of a key from a manual remote test facility). In repetitive or step modes, the maintenance scheduler controls the repeated execution of the test and can cause the result of the last test (pass or fail) to be indicated on lamps at the maintenance center display panel and the remote test facility. These features are most helpful in repairing faulty equipment units.[3,4]

Some programs which are subordinate to the maintenance monitor are known as subsystem maintenance monitor programs and are covered in companion papers.[3,4] Maintenance of the No. 2 ESS control complex is also covered elsewhere.[8]

2.2 Interrupt structure

Multilevel interrupt facilities are provided in the control unit to permit entry to programs immediately on demand. Interrupts have a priority structure such that the highest interrupt level demanded is executed. When an interrupt signal is generated, a program is entered at a fixed address corresponding to that interrupt level. When the interrupt is completed, control is returned to the program that was interrupted.

Table I — Interrupt structure

Interrupt	Source	Purpose(s)
High-Priority Maintenance	Demand by control unit	(*i*) Control unit mismatch recovery (*ii*) Digit scanning error recovery (*iii*) Utility request processing
Disc File 0 Disc File 1	Demand by disc file	Disc file block transfer input–output
Input-Output 25-Millisecond	Timed by control unit	Other input–output
Low-Priority Maintenance	Demand by control unit	Continue control unit mismatch recovery

Three types of interrupts are used in AIS, and they are assigned at five priority levels. A list of interrupts showing their relative priorities is contained in Table I.

2.2.1 25-millisecond interrupt

A periodic 25-millisecond interrupt, IO25, is generated by circuits within the control unit. This interrupt handles:

(*i*) Many input-output functions requiring execution more frequently than once per main program loop. Among these tasks are detection of trunk supervisory changes, processing of multi-frequency digits, and disc file lookups (see Section IV).

(*ii*) Tasks requiring precise timing between entries, such as teletypewriter input-output, or maintenance testing the duration of timing signals provided by peripheral circuits where the signal may last hundreds of milliseconds.

The IO25 interrupt programs are sequenced by a portion of the executive control operating at IO25 interrupt level. The control structure used in this interrupt is kept simple, since inefficiencies here would be multiplied by the high frequency of execution and thus would be wasteful of processing time. Hence, task programs generally interface directly with the IO25 executive.

The IO25 executive controls execution in one of four modes. Normally, all task programs are executed, but in other (abnormal) modes of operation, only essential tasks are executed. These other modes are:

(*i*) recovery from control unit mismatch;
(*ii*) reloading of nongeneric data from paper tape backup; and
(*iii*) system initialization.

Table II contains a list of IO25 interrupt functions in their order of execution and indicates the modes in which each is executed.

The duration of the IO25 interrupt is normally much less than 25 milliseconds. Should an occasional interrupt last longer than 25 milliseconds, "jitter" could be introduced in the timing of precisely timed tasks.

2.2.2 Disc file interrupts

Two *disc file interrupts* are provided, one per disc file, at adjacent priority levels. These interrupts are used to transfer large blocks of data between a control unit and a disc file (other than for routine call lookups) because the data cannot all be buffered in file control registers.[4] Data transfers requiring file interrupts are used for:

(*i*) administration and auditing of the intercept number data base[5] and the backup nongeneric data;
(*ii*) auditing of nongeneric data in call store; and
(*iii*) maintenance of the disc file subsystem.

Further details of the disc file interrupts are discussed in a companion paper.[5]

2.2.3 Maintenance interrupts

Two maintenance interrupts are used to process certain maintenance and manual requests. The *high-priority maintenance interrupt*, the highest-priority system interrupt, is used for:

(*i*) processing of manual utility requests from the maintenance center;
(*ii*) recovery from errors occurring in autonomous multifrequency digit scanning performed by the control unit; and
(*iii*) recovery from mismatches of the control units.

Processing requests for (*i*) and (*ii*) is rapid and, therefore, can be completed at this interrupt level. However, in case (*iii*), testing which ensues after a control unit mismatch requires up to 220 milliseconds. If all this testing were performed in this interrupt, all other system programs, including lower-priority interrupts, would not be executed during this period; this would most likely result in mishandling of some calls. Consequently, after several milliseconds at this interrupt level, the remaining testing is performed at *low-priority maintenance interrupt* level, the lowest-priority system interrupt. During this time, only base level work is delayed, permitting call processing input-output tasks to continue.

Table II — IO25 interrupt functions

Description	Mode of Operation			
	Normal	CU Mismatch Recovery	Tape Mode	System Initialization
Update system time	X	X	X	X
High-precision timing for teletypewriter maintenance	X		X	
Data outpulsing	X	X		
Teletypewriter input/output processing	X		X	
Autonomous scanner processing	X			
Announcement connections	X	X		
Disc file input/output	X		X	X
Digit receiving	X	X		
Service observing	X	X		
Directed scan processing	X			
Low-precision timing for announcement machine maintenance	X		X	

III. USE OF CALL STORE MEMORY

The call store in the AIS control complex is a direct-access memory consisting of modules of ferrite sheets operating on a 6-microsecond read-change-write cycle.[6] Each module is comprised of 4096, 16-bit words. Areas used by the operational programs may be classified into five categories:

(*i*) records of calls currently in progress;

(*ii*) maps which record the busy, idle, or maintenance states of peripheral equipment, trunks, and service circuits;

(*iii*) records of nongeneric data particular to an Automatic Intercept Center (AIC);

(*iv*) buffers, hoppers, and registers which receive (and transmit) data from (to) peripheral equipment; and

(*v*) traffic and plant administration and call queuing data.

Since the overall AIS call store requirements are small (compared with other stored program switching systems such as No. 1 ESS or

No. 2 ESS), enough call store is reserved at every installation to meet the maximum call capacity of an AIC. If a particular AIC is not fully equipped, the call store words which would have been associated with the nonexistent facilities are simply not used. This dedication of all call store areas eliminates any need for redefining call store layouts to accommodate changes in an operating AIC; thus, it is completely compatible with the generic program concept.

3.1 Classes of call store data

Data stored in the call store memory can be classified into one of three categories according to their use and frequency of alteration: transient data, semipermanent data, and stable data.

3.1.1 Transient data

Transient data words are changed frequently and are associated with the processing of a particular call or maintenance action. For example, the digits received by the AIC from a local office, and the linkages between call-active equipment control registers, are transient data.

3.1.2 Semipermanent data

Semipermanent data call store words contain the nongeneric data associated with a particular AIC. The semipermanent data in call store is backed up on both disc files and is altered only by plant changes (Section 7.2). An example of nongeneric data is the information describing the equipping of a trunk network.

Certain call store words contain both transient and semipermanent data. This permits more efficient call processing, but increases the complexity of initializing and auditing these words. Many of the equipment control registers (Section 3.3) are examples of shared data type words.

3.1.3 Stable data

Stable data words contain information which is retained by the system over an extended period of time, which may be updated periodically, and which, if lost, is very difficult to reconstruct. They are similar to transient data words, except that they are not related to the major call processing and maintenance functions of the system. Thus, it is unlikely that errors in these words would cause severe software problems; hence, they are never reinitialized by an automatically triggered system initialization (Section VI). The time of day is an example of stable information. Once entered into the system, it is frequently updated and remains in the system almost indefinitely.

3.2 Call-in-progress register

An eight-word control register, known as a *call-in-progress register* (CIPR) (Fig. 3), is assigned to every call as it "enters" the system. The same CIPR stays with the call until it is disconnected. It is used to maintain an up-to-date record of the state of the call and controls the processing of the call in the system.

There are 128 CIPRs per switching network. Since each switching network can connect 64 calls simultaneously, approximately 64 additional calls can enter the system in various queues before the AIS begins to reject originations from the local offices (Section 4.2).

Word 0 of each active CIPR contains the state of the associated call, known as a *progress mark*. The state is encoded such that it equals the address of the beginning of the program routine which handles this state. Access to progress marks is facilitated by a special-purpose "macro-type" control unit command that causes a transfer based on the nonzero contents of the first word of the CIPR.[6] If the contents of the word are zero, no call is associated with this CIPR and the command advances to the next CIPR.

Once during every main program loop each CIPR is accessed and control is transferred to the program indicated by a nonzero progress mark. That program then determines if there is any new information

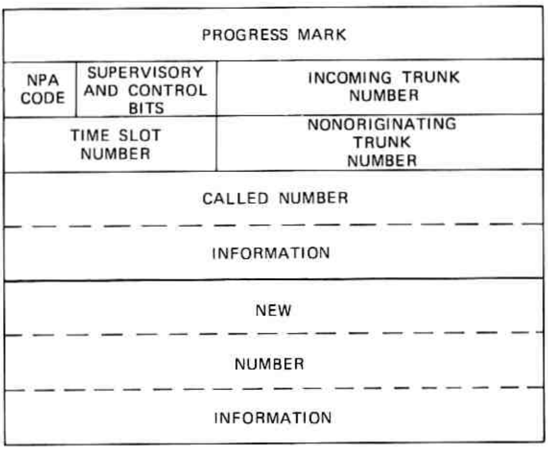


Fig. 3—Call-in-progress register (CIPR) format.

about the call and, hence, if any new action is to be performed on this call. When there is no action to be taken, the progress mark is not changed and the next CIPR is looked at. If there is new information, the program takes the appropriate action; if this action changes the stage of the call, a new progress mark is assigned to the call reflecting the new stage, and the progress mark is written in the CIPR. The action taken on a progress mark entry advances the call to a point where no further processing is possible without a real-time break. Typically, the break is necessary when waiting for unavailable facilities, for response from a peripheral circuit, or for a timed period to elapse.

The efficiency of this nonzero progress mark approach for all active calls results from two characteristics of intercepted calls:

(*i*) most intercepted calls have a short holding time, and
(*ii*) active calls in the system are usually not in a stable state (e.g., announcement connections are changed every 0.5 or 1.5 seconds).

Hence, the little processing time wasted monitoring stable connections is justified by the simplicity achieved in performing the call processing function.

The second and third words of the CIPR are used to store the originating trunk equipment location and the equipment location of the facility to which it is connected, as well as supervisory information for both trunks and the number of the network time slot in which the connection is made. The rest of the CIPR contains the called number dialed by the customer or keyed by the operator, a digit indicating the call class, and the file reply for calls requiring a disc file lookup.

3.3 Facility control registers

In addition to the controlling CIPR, a time slot and various other hardware facilities are needed at different stages of the call to process it properly. These hardware facilities are known as *nonoriginating equipment* since they are used to help process calls and do not originate work. When active on a call, both the incoming trunk and the various items of nonoriginating equipment are linked to the controlling CIPR. To do this a call store facility control register is associated with each piece of equipment and the number of the controlling CIPR is recorded in this register.

These control registers are arranged by facility type and are generally one or two words long. In addition to recording the number of the controlling CIPR of an active facility, these facility control registers

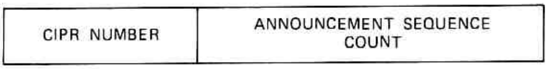


Fig. 4—Time slot word format.

may contain semipermanent data pertinent to the piece of equipment involved. All incoming switching network trunks and time slots have associated facility control registers.

3.3.1 Time slot words

The AIC interconnects trunks via a time-division switching network that provides 64 time slots in which simultaneous connections are made.[3] Associated with each time slot is a call store *time slot word* (Fig. 4). When a time slot is in use, the time slot word contains the number of the CIPR controlling the associated call. Additionally, when a call is in the announcement stage, an *announcement phrase count* is kept of the number of 0.5-second time periods which have elapsed since the announcement sequence began. This count is used to determine the next announcement connection to be made for the associated call.

3.3.2 Primary trunk words

Most of the 512 trunks or service circuits (except the 96 announcement trunks) can appear at any equipment location on the switching network. Therefore, a detailed layout of switching network equipment is provided in a contiguous block of call store known as *primary trunk words* (Figs. 5a and b). Primary trunk words are ordered according to the equipment location of the associated trunks (i.e., numbered position on the network). For example, trunk number 106 (group 1, vertical 0, horizontal 6) would have its corresponding call store word in the 106th (octal) slot of the primary trunk block.

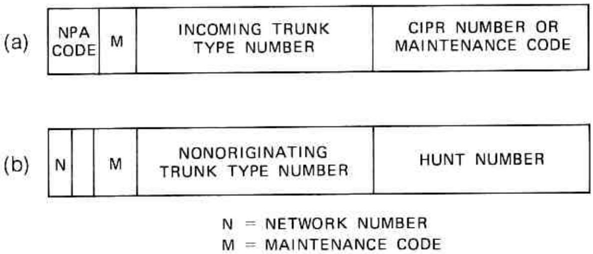


Fig. 5—Primary trunk word format.

Each primary trunk word contains the *type number* of the associated trunk. This type number designates the function of the trunk or, in some cases, distinguishes trunks with different supervisory arrangements. Each word also contains two bits indicating the present maintenance state of the trunk.

The remaining bits of a primary trunk word vary according to the type of trunk. In general, if the word is for an originating trunk it contains the number of the associated CIPR or a code describing a current maintenance condition. For incoming intercept trunks, also included is a code used to indicate from which of four possible NPAs served by the AIC this trunk originates. For trunks and service circuits which are selected, a trunk selection number (or *hunt number*), assigned when the equipment is installed, is recorded in the primary trunk word. For outgoing trunks to Central Intercept Bureau (CIB) operator positions, an indication of the network on which the associated incoming trunk from the CIB position appears is also included.

3.3.3 Secondary trunk words

The primary trunk word for a nonoriginating trunk (other than a test trunk and *Touch-Tone*® receiver) provides translation from the trunk equipment location to its hunt number. The *secondary trunk word* (Fig. 6) provides the reverse translation, that is, from the hunt number to the trunk equipment location. Each type of huntable trunk and service circuit has an associated block of secondary trunk words.

Each secondary trunk word contains the equipment location of the corresponding trunk. When the trunk is active on a call or placed in a maintenance state, the number of the controlling CIPR or a maintenance code specifying the exact condition, respectively, is also stored therein.

In a typical AIS, a small number of operators are needed. Therefore, outgoing trunks to operator positions are multiplied to both networks. Thus, each operator position in a two-network installation has the same outgoing trunk appearance on each network and only one secondary trunk word.

3.3.4 Facility selection words

Hunted trunks and service circuits, CIPRs, and network time slots are assigned facility selection words known as *hunt words*. Each block of these words records the current busy/idle status of facilities of that type. These selection words are divided into categories which depend upon the extent to which traffic is to be distributed evenly over that

TRUNK EQUIPMENT LOCATION	CIPR NUMBER OR MAINTENANCE CODE

Fig. 6—Secondary trunk word format.

group of facilities. Usage of outgoing trunks to operator positions and to service assistant consoles is precisely equalized by using ordered facility selection words. Usage of other facilities is not equalized and nonordered facility selection words are used.

3.3.4.1 Nonordered facility selection words. The busy/idle state of an individual facility is represented by one bit in a facility selection word of that type. The bit number within the facility selection words corresponds to the hunt number of the facility. When an idle facility is to be selected, a linear search is made of the busy/idle bits until an idle facility is found. For cases where it is not desirable to select the same facility in periods of light traffic, a random starting point is used for the search. This also results in roughly equalizing usage of the facilities.

3.3.4.2 Ordered facility selection words. When use of facilities of a type is to be precisely equalized, ordered selection words are used. Each ordered huntable facility is assigned two bits, a busy/idle bit and an *order bit*. The order bits are searched for facilities next in line for selection; the busy/idle bits are used to record which facilities have become idle after selection. Hence, the true availability state of each facility is represented by a combination of the two bits.

3.4 Buffers and hoppers

Base level and IO25 interrupt programs communicate with each other via call store *buffers* and *hoppers*. Generally, buffers are used to pass information from base level programs to interrupt programs, whereas hoppers are used to pass information to base level programs.

3.4.1 Peripheral order buffers

A *peripheral order buffer* (POB) is associated with each network time slot. Each POB contains the equipment locations of the incoming trunk and the announcement track trunk to be connected during the next announcement sequence (Fig. 7).

POBs are loaded by a call processing program at base level and are unloaded at the time the corresponding announcement connections are made at IO25 interrupt level. Constructing announcement connections

ANNOUNCEMENT CONTROL	ANNOUNCEMENT TRACK EQUIPMENT LOCATION
TIME SLOT NUMBER	INCOMING TRUNK EQUIPMENT LOCATION

Fig. 7—Peripheral order buffer (POB) format.

at base level saves IO25 interrupt time. The connections are made at IO25 interrupt level to properly synchronize announcement connections machine phrase timing. All other orders to make connections are issued at base level upon demand because the timing requirement is not as stringent.

3.4.2 Autonomous scan hoppers

During the IO25 interrupt, a program interrogates the master scanner ferrods associated with each autonomous scanner to determine if a trunk supervisory change of state has been reported. If a message is present, the IO25 program loads it into the first vacant entry of the *autonomous scan hopper* (Fig. 8). There is one autonomous scan hopper per network, each containing 10 entries. To conserve IO25 interrupt time, each message stored in this hopper is recovered and processed at base level.

3.4.3 Directed scan hoppers

A *directed scan* is used generally to determine on demand the supervisory state of a trunk. In addition, a directed scan is performed to obtain the type of call intercepted on a three-class operator number identification (ONI) trunk (i.e., whether the call is a regular intercept, blank number, or trouble intercept). This is possible since three-class ONI trunks can inform the AIC via dc signaling of the class of intercept; this dc signal is registered on two master scanner ferrods when the trunk is directly scanned.[3]

Requests for directed scans of trunks are initiated at base level by placing the number of the CIPR associated with the request into a *directed scan hopper*. In a subsequent IO25 interrupt the request is

M	CONTROL INFORMATION	OFF-HOOK BIT	ON-HOOK BIT	EQUIPMENT LOCATION OF TRUNK THAT CHANGED STATE

M = MESSAGE PRESENT FLAG

Fig. 8—Autonomous scan hopper format.

removed and a directed scan order issued to the associated connector and scanner. The answer to this order is retrieved about one millisecond later during the same IO25 interrupt and is stored in the CIPR.

3.5 Input-output registers

Certain input-output functions requiring frequent attention are performed autonomously by control unit circuits. The performing of these functions via direct program actions would consume excessive real time. The information obtained from and used by these circuits are stored in call store *input-output registers*.

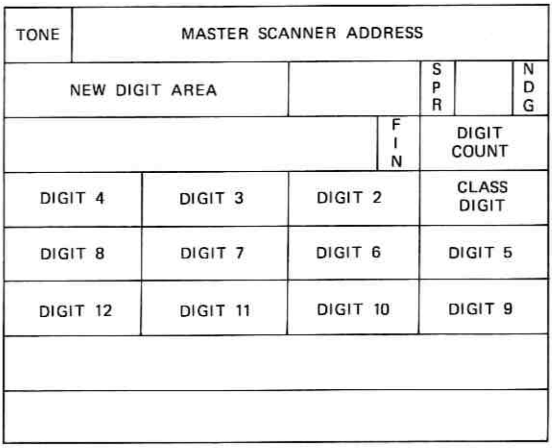
3.5.1 Originating registers

One *Touch-Tone* receiver and several multifrequency receivers are provided in an AIC for digit receiving. The scanning of digit receivers for new digits is an input-output function that is performed at a 10-millisecond rate autonomously by control unit circuits. These circuits monitor the ferrods assigned to the digit receivers and, when digits are received, store the new digits in call store records called *originating registers* (ORs). During alternate IO25 interrupts, the digit receiving program stores the new digits that have been placed into ORs, permitting another new digit to be received in each OR. When a digit receiver is connected, software linkage is established between a CIPR and the OR corresponding to the digit receiver.

One OR is *dedicated* to each digit receiver and it can store all digits received from a local office or an operator position on any intercept call. The first word of each OR contains information which identifies the master scanner row and ferrods associated with the digit receiver permanently assigned to that OR. The format of an OR is shown in Fig. 9.

3.5.2 Data outpulsing buffers

A 64-word call store *data outpulsing buffer* is used for the storage and transmission of up to 16 data outpulsing messages (eight per switching network) for display at CIB operator positions. These messages are loaded by an IO25 interrupt program and transmitted by control unit circuits at a rate of about 800 bits per second. A data message is stored vertically, one bit per word, in storage slots referred to as *data channels*; one data outpulser is permanently associated with each channel. Circuits transmit one word every 1.251 milliseconds, where each word contains one bit for each of the 16 data channels. The transmission



TONE = MF OR TOUCH-TONE® RECEIVER
SPR = SIGNAL PRESENT FLAG
NDG = NEW DIGIT FLAG
FIN = RECEIVING FINISHED

Fig. 9—Originating register (OR) format.

over each data channel is continuous; when no message is present, nulls are transmitted.

Since the circuits are arranged to transmit only 64 bits per message, the transmission of the 94-bit data message to a CIB operator position must be done in two parts. The first part of the message is loaded into the outpulsing buffer slots of a channel. The remaining bits are stored vertically in a supplementary outpulsing buffer. After approximately 40 bits have been transmitted (i.e., two IO25 interrupts later), the rest of the message is constructed by appending the bits in the supplementary outpulsing buffer to the remaining untransmitted bits in the outpulsing buffer.

3.6 Queue structure and control

Most calls bidding for facilities are placed in queues. This allows facilities to be distributed in a predetermined order of priority based on the importance of the call type. Exceptions to this rule are the assignment of CIPRs which is done on a first-come, first-served basis and the assignment of multifrequency (MF) receivers and time slots to new call originations on which digit receiving is expected (Section 4.2.2.1).

Each switching network has a block of 128 call store words reserved for call queuing. This block is divided into 12 queue categories which

N	TIME SLOT NUMBER	P	CIPR NUMBER	H

N = NETWORK NUMBER
P = INDICATION OF TRAFFIC PEG COUNTER INCREMENTED
H = ON-HOOK/OFF-HOOK BIT

Fig. 10—Queue word format.

are used for different classes of calls. The length assigned to the various queues is dependent upon the traffic mix handled by the AIC. In addition, in the call store queue block of the first network, a 13th queue is used to store disc file lookup requests for calls appearing on incoming trunks appearing on either switching network. Except for the disc queue, queues are assigned an order of priority. Higher-priority queues have access to facilities before lower-priority queues bidding for the same facilities. Within a queue, calls are served on a first-come, first-served basis when facilities become available. Each queue entry is one word and contains several items of information as shown in Fig. 10. Associated with each queue is a four-word control block shown in Fig. 11.

3.7 Other semipermanent data areas

3.7.1 NNX-ABX translation table

The NNX-ABX translation table gives an AIC the capability of announcing local office codes as two letters and a digit (ABX) instead of as three digits (NNX). Each entry in this table contains the three NNX digits, a code which is translatable into the Numbering Plan Area (NPA) of the local office, and control bits which indicate the circumstances when ABX should be announced. Each entry also contains the equipment locations of the associated alphabetic announcement tracks for the NN digits of the local office code.

3.7.2 Miscellaneous nongeneric data

In addition to the previously mentioned data, several other items of nongeneric data are needed to define an AIC. These include codes

QUEUE WRITE ADDRESS
QUEUE READ ADDRESS
QUEUE FIRST LOCATION
QUEUE LAST LOCATION

Fig. 11—Queue control block format.

which identify the AIC and the NPAs served by it, indications of the optional features provided for at the AIC, and indications of the amount of optional equipment at the AIC.

3.8 Nongeneric backup data

Since the AIS stored program is totally generic, each installation has nongeneric data which define the way it is equipped. Under normal operating conditions, the data which are stored in the semipermanent data areas of call store are durable. However, system design errors or hardware faults could lead to circumstances which could destroy the duplicated call store data. Therefore, it is not prudent to depend on call store alone for storing the nongeneric data. In an AIC two electrically alterable disc files are otherwise required to store records of intercepted numbers. Accordingly, a small section of both of these files is used to provide backup for the nongeneric data. In case nongeneric call store data are mutilated, they can be reinitialized quickly from a disc backup (Section VI). This backup data base also provides a convenient and expedient method for making changes to nongeneric data (Section 7.2).

When nongeneric data are placed in call store, considerable redundancy is introduced to increase the efficiency of call processing programs. The format of the file backup version of the data is quite different from that of call store. Rather than providing an image of call store, file information is organized in a rigid, virtually nonredundant format. When reinitializing call store semipermanent data, many disc file accesses are required to obtain the backup data. The time required to perform each access is an order of magnitude greater than the processing time required to reinitialize call store data from the file data obtained during that access. Thus, by eliminating redundancy, the time necessary to restore the system from backup data is minimized. Furthermore, this data organization lowers the probability of inconsistencies within the backup data structure, which could lead to conflicting and inconsistent action by the audit programs (Section V).

Nongeneric data are stored on one track of each disc file using a maximum of twenty-three 20-word blocks, each word containing 42 data bits. The number of blocks required for a given AIC depends on the number of switching networks equipped and the number of NNX-ABX translations. All but two blocks contain trunk or NNX-ABX translation information, so that packing schemes have been developed to most efficiently pack these classes of data. The remaining blocks contain all other nongeneric data.

3.9 Register linkages

Many call store registers, some associated with hardware facilities, are needed to successfully process intercept calls. Many of these registers are closely related and permanently linked to each other, as shown in Fig. 12. In addition as calls are processed other transient relationships develop between the registers. Pointers, redundant for call processing expediency, are introduced into call store to link these registers together. An example of these transient linkages for operator-subscriber connections is also shown in Fig. 12.

IV. CALL PROCESSING PROGRAMS

The call processing programs control the processing of intercept calls routed from local switching offices. This section describes the functions performed by these programs.

4.1 Call processing program organization

4.1.1 Basic call processing functions

Call processing functions are accomplished by a series of highly interrelated programs, each of which performs a specialized task. These tasks include:

- (*i*) detection and processing of trunk supervisory changes (Section 4.2);
- (*ii*) reception and analysis of sequences of multifrequency digits (Section 4.3);
- (*iii*) communication with the disc files to determine the disposition of intercept numbers (Section 4.4);
- (*iv*) determining and establishing sequences of announcement connections to incoming intercept trunks and operator positions (Section 4.5);
- (*v*) connecting incoming trunks to operator positions and reacting to operator keying actions (Section 4.6);
- (*vi*) performing call disconnects (Section 4.7); and
- (*vii*) performing functions ancillary to call processing (Section 4.8).

The relationships among these functions are best illustrated by examining a typical call (Fig. 13). Consider an intercept call which is to be routed from a local office equipped with *automatic number identification* (ANI) features, i.e., equipped to automatically identify the called number.* A trunk to the AIC is seized at the local office. When

* This differs from the standard use of ANI, which is to identify the calling number.

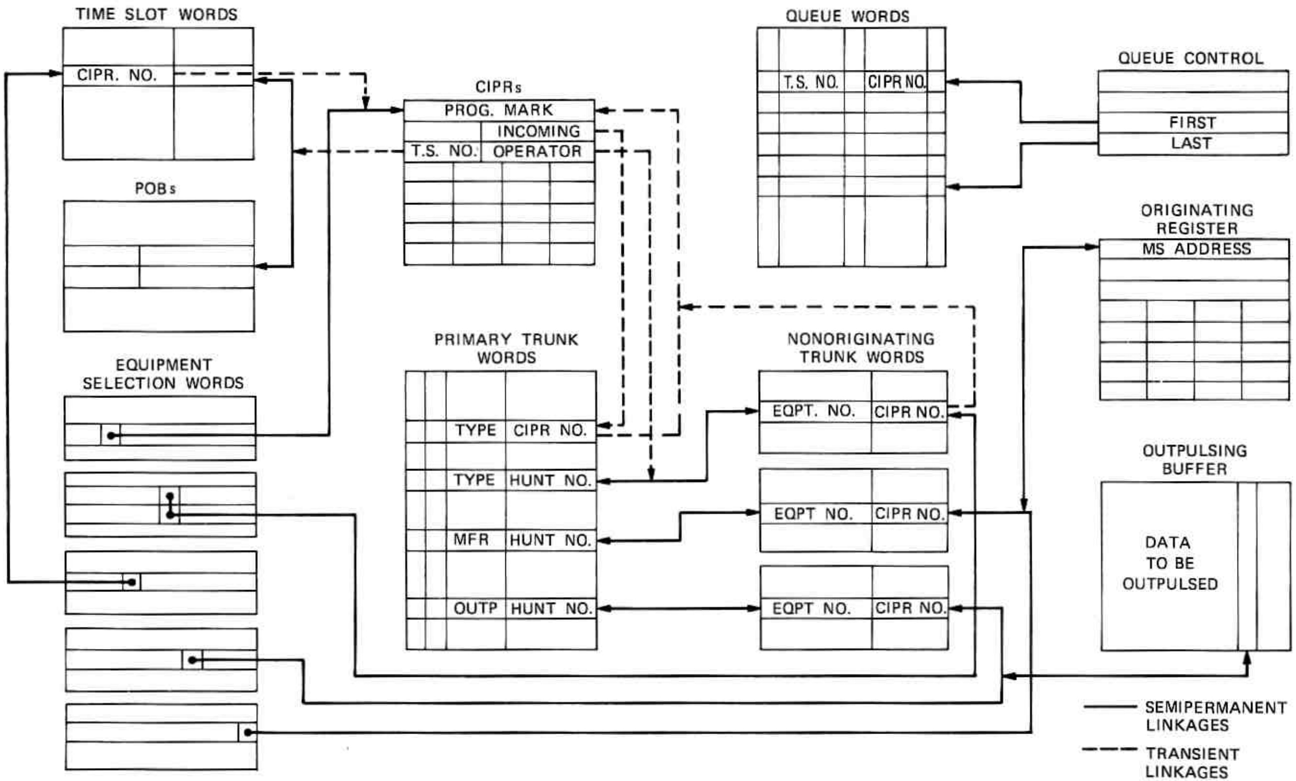


Fig. 12—Semipermanent call store linkage structure and transient linkage structure for connection between operator position and incoming intercept trunk.

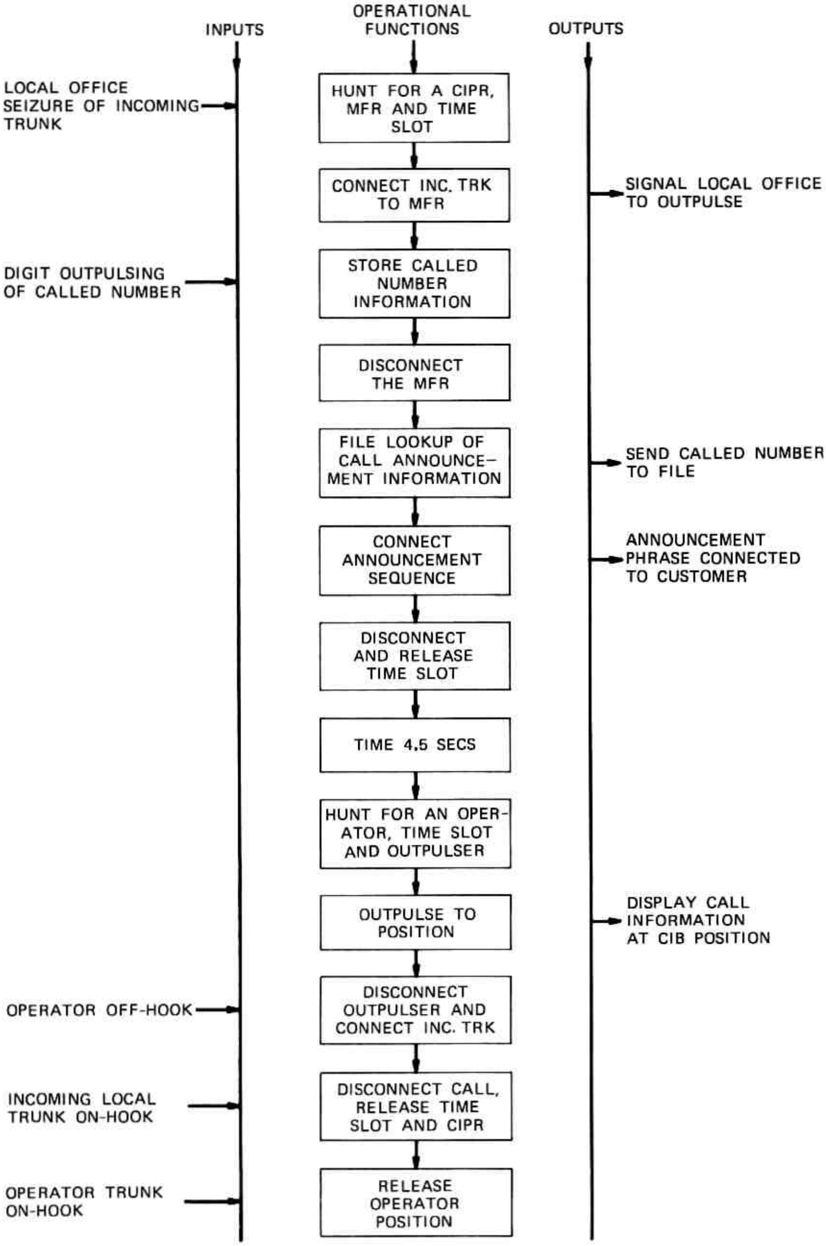


Fig. 13—Typical ANI call sequence.

the seizure is recognized at the AIC, an MF digit receiver is connected in a network time slot to the incoming intercept trunk. This connection causes the MF receiver trunk to send a *wink* signal[3] to the local office, indicating that the MF receiver is connected. The local office then outpulses a stream of MF digits, identifying the called number and the class of intercept call.

When all digits have been received, the MF receiver is disconnected and the digits are analyzed. If required, a disc file lookup is performed to determine the intercept number disposition. If the lookup indicates that an announcement sequence should be connected, the incoming trunk is connected to a series of 0.5- and 1.5-second announcement phrases. This informs the caller of the intercepted number status and of a new number, if available.

After the announcement, a delay of 4.5 seconds is provided to permit disconnect. If the caller remains on the line, the intercept trunk is connected to a CIB operator position for operator assistance. All details regarding the intercepted number are outpulsed and displayed to the operator. Access to additional information via the disc files is available to the operator upon keying a suitable sequence of MF digits. The call is disconnected when the incoming intercept trunk goes on-hook.

4.1.2 Call processing program structure

The structure of the call processing programs is depicted in Fig. 14. These programs are executed at 1025 interrupt level and at base level. The 1025 interrupt programs interface the software with autonomous

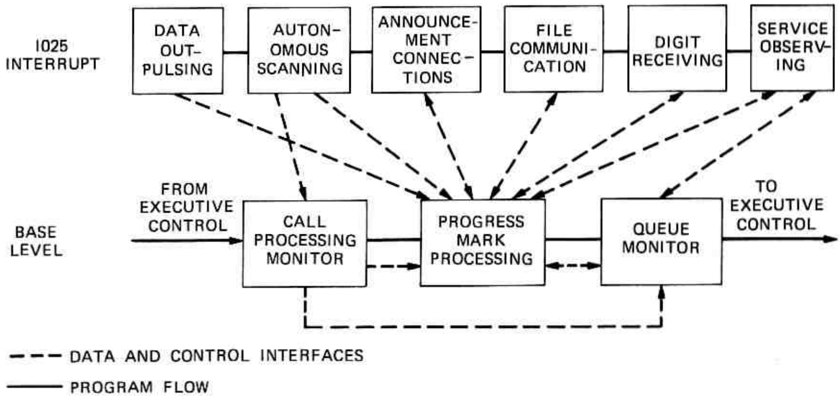


Fig. 14—Call processing program structure.

hardware. In general, base level programs process input signals passed from IO25 interrupt programs and generate outputs to be transmitted by IO25 interrupt programs.

The call processing monitor controls the flow of the three base level call processing programs: trunk supervisory change, progress mark processing (Section 3.2), and queue administration. These programs are sequenced to minimize the elasped time before trunk supervisory changes are processed and to maximize the first-come, first-served use of available facilities. Accordingly, trunk supervisory change processing is performed first; this may involve setting up initial connections, placing calls in queue, or passing inputs via the CIPRs to the progress mark programs. Progress mark routines are executed next, before the queue administration program, so that facilities released by the progress mark routines can be used to serve calls in queue. The queue administration program executes last. It selects available facilities for calls in queue, and then updates the progress marks for further processing actions in the next main program loop. This sequence insures that calls in queue will have access to facilities before new originations.

The IO25 interrupt programs are sequenced so that critical time-dependent functions are executed first. As an example, Section 3.5.2 relates the close interaction between circuits and IO25 interrupt programs to accomplish data outpulsing. This interaction is precisely timed to insure that the second segment of the outpulsed message is properly appended to the first segment.

4.2 Supervisory change detection and processing

Supervisory changes of state (between on-hook and off-hook) are detected on incoming trunks from local offices and from remote AICs, on trunks from and to operator positions, and on trunks to home AICs. Such changes inform the program of call originations and disconnects, and of changes of service state of operator positions and trunks to home AICs. The program reacts to each change of state by performing an appropriate processing action.

4.2.1 Detection of supervisory changes

Changes of supervisory state are reported by a combination of hardware and software actions. The autonomous scanner in the time-division network monitors trunk supervisory states by sequentially comparing the present states of trunks with those recorded in its last look memory.[3] When a change is detected, the autonomous scanner stops and indicates on master scanner ferrods the equipment location of the trunk and the direction of the change.

An IO25 interrupt program interrogates the master scanner once every 25 milliseconds for autonomous scanner messages. If a message is present, the information is placed in the autonomous scan hopper for further processing (as described in Section 3.4.2). In this manner, one change of state per switching network can be reported in each 25-millisecond interval. This provides for sufficiently rapid trunk scanning, independent of the length of the main program loop.

4.2.2 Processing of supervisory changes

Initial processing of supervisory changes is performed during the IO25 interrupt. In general, the change of state is kept in the autonomous scan hopper for base level processing and the autonomous scanner is restarted with its last look memory updated.[3] However, if the change is an off-hook indicating a new seizure of an originating trunk (such as an incoming intercept trunk), some additional processing is performed. A CIPR is selected and call store linkage is established from the primary trunk word to the CIPR. In the event that no CIPR is available, the origination cannot be accepted. In this case, the autonomous scanner is restarted but the last look memory is not updated. The origination may be detected again on a later autonomous scan cycle. Delaying the acceptance of new call originations at the AIC provides a natural defense against overloading system processing capabilities.[1]

Further processing of trunk supervisory changes is performed by base level programs. The actions taken by these programs depend on:

(*i*) the type of trunk that changed state;
(*ii*) the direction of the supervisory change; and
(*iii*) whether or not the trunk was associated with a call and, if so, the type of call and the stage to which it had progressed.

Base level actions are described in more detail in the following sections.

4.2.2.1 Processing changes on incoming trunks. Incoming trunk changes from off-hook to on-hook are reported by the base level trunk supervisory program in the CIPR associated with the trunk. This action results in the progress mark program disconnecting the call and releasing the CIPR for another call (Section 4.7).

The action taken on on-hook to off-hook changes of state (call originations) depends on the trunk type. In general, a time slot is assigned to the call and a connection is made between the incoming trunk and nonoriginating equipment (e.g., digit receiver or outgoing trunk to an operator position). Table III contains details of actions taken on call originations, including a number of exceptions to the above.

4.2.2.2 Processing changes on outgoing trunks to operator positions. The actions taken on changes of state of outgoing trunks to operator positions depend on whether or not a CIPR has been assigned. If the trunk state changes from on-hook to off-hook with no CIPR assigned, this indicates the operator has placed the position in a *made busy* state and the appropriate bit in a facility selection word is set indicating this position is temporarily not available.

Changes of state on trunks that have a CIPR assigned are reported in the associated CIPR for action by the progress mark programs. When an off-hook to on-hook change of state is reported, the position is eventually idled (Section 4.7). Conversely, an on-hook to an off-hook change of state usually verifies that the operator is talking on a connection (Section 4.6).

4.2.2.3 Processing changes on outgoing trunks to home AIC. Calls requiring CIB operator assistance at remote AICs are routed via outgoing trunks to the home AIC. When a connection is made at the home AIC between an incoming trunk from a distant AIC and a CIB position, a signal is passed which causes the outgoing trunk at the remote AIC to go off-hook. The call processing program at the remote AIC then outpulses the information necessary for the CIB display. After a timed interval, the incoming intercept trunk is connected to the CIB operator position. After the call has been completed, disconnect actions are similar to those performed at a home AIC.

4.3 Digit receiving and analysis

Digit receiving programs accumulate, decode, and analyze sequences of MF and *Touch-Tone* digits. Analysis of a sequence of digits is complex, since the digits can be received from various sources and be associated with various stages of calls. Sequences are received at the AIC from:

(*i*) ANI-equipped local offices, when MF outpulsing the intercept number;

(*ii*) CIB and ONI operator positions, when MF keying the called number on calls intercepted in non-ANI local offices (Sections 4.6.1 and 4.6.3.2);

Table III — Actions taken on call originations

Type of Trunk	Initial Actions		
	All Facilities Available	Time Slot Available; Service Circuit or Operator Position Not Available	Time Slot Not Available
Incoming Intercept (ANI)	Connect to MF receiver (Section 4.3)	Queue for time slot and MF receiver	Queue for time slot and MF receiver
Incoming Intercept (1-Class ONI)	Connect to ONI operator position (Section 4.6.1)	Queue for ONI operator position; apply audible ring	Queue for time slot and operator position
Incoming Intercept (3-Class ONI)	Regular intercept—connect to ONI operator position	Queue for time slot and ONI operator position	Queue for time slot and ONI operator position
Determine Call Class by Directed Scan (Section 4.6.1)	Trouble intercept—connect to trouble operator position	Queue for time slot and trouble operator position	Queue for time slot and trouble operator position
	Machine intercept—connect to blank number announcement	Not applicable	Queue for time slot
Incoming from Remote AIC	Queue for CIB operator position (Section 4.6.3)	Queue for time slot and CIB operator position	Queue for time slot and CIB operator position
Incoming from CIB Operator	Connect to MF receiver (Section 4.3)	Queue for time slot and MF receiver	Queue for time slot and MF receiver
Incoming Operator Training	Connect to *Touch-Tone* receiver (Section 4.8.3)	Wait for *Touch-Tone* receiver and release time slot	Wait for *Touch-Tone* receiver and release time slot
Incoming Directory Assistance	Queue for time slot and CIB operator position (Section 4.6.3.2)	Not applicable	Not applicable

(*iii*) CIB operator positions, when keying MF digits requesting additional displays or announcements of intercept number information (Section 4.6.3.1.) or when connecting a service assistant on the call (Section 4.6.4);

(*iv*) a trainer when entering *Touch-Tone* digits while training a CIB operator (Section 4.8.3);

(*v*) CIB operator positions, when keying MF digits upon vacating or reoccupying a position (Section 4.6.3); and

(*vi*) ANI/ONI concentrators, when identifying the type of ONI call (Section 4.6.1).

In response to certain types of trunk supervisory state changes, an appropriate digit receiver is connected in a time slot, causing a signal to be returned by the digit receiver trunk to the transmitting source. This indicates that a receiver is connected and outpulsing can begin.

4.3.1 Digit reception

Each MF digit transmitted to the AIC is indicated in a 2-out-of-6 code on master scanner ferrods and subsequently placed in an OR as described in Section 3.5.1. *Touch-Tone* digits are handled similarly except that they are transmitted in a 1-out-of-4, 1-out-of-4 code. Checks are performed at IO25 interrupt level on the digit sequence to determine whether or not the sequence is valid.

While the IO25 interrupt digit receiving program is storming digits in an OR, the base level digit analysis program is checking for the completion of digit receiving. When the start pulse (ST) is received indicating the end of the digit stream, the digit receiving program notifies the analysis program that all digits have been received. The digits are then transferred to the associated CIPR. If the digit outpulsing is not completed within 24 seconds or if the IO25 interrupt digit receiving program has indicated an error in receiving, the call is placed in queue for a CIB operator. When the call is connected to the operator position, an indication, AIC FAILURE, is displayed at the position. The error is also reported to maintenance programs so that diagnostic tests may be performed on the digit receiver. If no error has occurred the digits are analyzed.

4.3.2 Digit analysis

The interpretation of a digit message is done at base level by progress mark digit analysis programs. Analysis depends on the source of the digits, the class digit contained at the beginning of the digit stream and

Table IV — MF and *Touch-Tone* keying sequences

Source	MF Sequences	Explanation and Disposition
Local office via ANI trunk	(KP) 0 (ST)	Machine intercept; give blank number announcement.
	(KP) 1 (ST)	Trouble intercept; connect to trouble operator position for assistance.
	(KP) 2 (ST)	ANI failure; connect to CIB operator position for assistance.
	(KP) 3 NXX XXXX (ST)	Regular intercept; perform disc file lookup to determine called number disposition.
Local office ONI calls via ANI/ONI concentrator	(KP) 5 (ST)	Announcement previously given at ANI/ONI concentrator; route to CIB operator for post-announcement assistance.
	(KP) 6 (ST)	Regular ONI intercept; connect to ONI operator position for number identification.
	(KP) 7 (ST)	Machine intercept; give blank number announcement.
	(KP) 8 (ST)	Trouble intercept; connect to trouble operator position for assistance.
CIB operator position	3 (ST)	To SA key depressed; establish connection to service assistant console.
	3 01X (ST) 3 01XNXX (ST)	Operator-keyed trouble report; generate printout at maintenance teletypewriter.
	3 1YX (ST)	Test call.
	3 NXX XXXX (ST) 3 NYX NXX XXXX (ST)	If KP ONI key depressed, regular intercept call identified by operator; if KP ANN key depressed, CIB operator requesting additional announcement; in either case, perform disc file lookup and give announcement.
	4 NXX XXXX (ST) 4 NYX NXX XXXX (ST)	KP DISP key depressed; display additional information to CIB operator.
	9 (ST)	Operator has vacated position by removing head set.

Table IV (continued)

Source	MF Sequences	Explanation and Disposition
ONI operator position	3 01X (ST) 3 01XNXX (ST)	Operator-keyed trouble report; generate printout at maintenance teletypewriter.
	3 NXX XXXX (ST) 3 NYX NXX XXXX (ST)	Regular intercept call identified by operator.
	Touch-Tone Sequences	
Trainer while training a CIB operator	(KP) 0 (ST) (KP) 0 NNX XXXX (ST)	Machine intercept; give blank number display at training position.
	(KP) 1 (ST) (KP) 1 NXX XXXX (ST)	Trouble intercept; connect to training position with trouble intercept display.
	(KP) 2 (ST)	ANI failure; connect training position with ANI failure display.
	(KP) 3 NXX XXXX (ST)	Regular intercept; connect training position with post-announcement display.
	(KP) 5 (ST)	Directory assistance call; connect to training position with directory assistance.
	(KP) 6 (ST)	AIC failure; connect to training position with AIC failure display.
	(KP) 7 (ST)	Post-announcement assistance on ONI-originated call; connect to training position with proper display.
	(KP) 8 (ST)	ONI intercept call; connect to training position with display signifying ONI function to be performed.

Key: N = 2, 3, 4, 5, 6, 7, 8, 9
X = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Y = 0, 1

possibly on the first two or three digits of the message itself. Table IV lists the possible keying sequences from each source and gives their interpretation.

Once analysis is completed and the proper call action ascertained, the progress mark is updated. For example, after the MF digit sequence (KP) 3 NXX XXXX (ST) is transmitted from a local ANI office and the IO25 interrupt digit reception program has accepted the digits, the base level analysis program begins interpreting the digit stream. Because the digit source is a local ANI trunk, the class digit alone determines the action to be taken. In this case, the class digit 3 indicates a regular intercept call and the progress mark of the call is changed so that a file lookup is initiated. This is described in Section 4.4.

4.4 Call processing disc file communication

4.4.1 File access

Regular intercept calls require a disc file lookup based on the seven-digit called number to determine the status and any new number information associated with the call. All calls requiring a file lookup are placed in the call store disc queue (Section 3.6) containing calls waiting for a file lookup. These requests are entered in this queue by the progress mark routines during base level call processing and are removed at IO25 interrupt level. After it is determined that a file is available for a lookup, the IO25 call processing file communication program removes an entry from the disc queue and sends the called number to the file where the lookup is performed associatively by addressing the file with this number.[4]

4.4.2 Data retrieval and analysis

The average file response time for a lookup is 160 milliseconds.[5] After initiating a file lookup, a program reads a status register in the file control circuit during every subsequent IO25 interrupt to determine if the lookup is completed. If the lookup is successful, the file reply is contained in the input-output register in the file control circuit and the file is made available for other use. The reply is placed in the associated CIPR; this information normally consists of:

(*i*) a two-digit *status* indicating the disposition of the intercept number (for example, the called number is disconnected);
(*ii*) a new number associated with the intercept number, if appropriate; and
(*iii*) new NPA or new locality information, where appropriate.

A file lookup may fail for a variety of reasons. For example, the intercept number may not appear on the file, in which case the caller is connected directly to an operator position with an indicator, NOT IN FILE.

When both files are unavailable due to maintenance conditions, callers are given a blank number announcement (Section 4.5). This announcement is used to provide limited service when the status and new number information are unavailable.

4.5 Announcement connections

The announcement sequence connected on a call depends on the call class and, where appropriate, the results of a disc file lookup. Three types of announcement sequences can be connected:

(*i*) The *blank number announcement* informs the caller that the number he reached is not a working number. It is applied on machine intercept calls, and on regular intercept calls requiring a file lookup when the lookup cannot be performed because neither disc file is available for service.

(*ii*) The *working number announcement* informs the caller that the number he intended to dial is a working number. It is applied on ONI calls when a file lookup results in a not-in-file, a condition most likely caused by a dialing error.

(*iii*) The *"customized" intercept announcement* informs the caller of the disposition of an intercept number as defined in the disc file record. The type of announcement is based on the intercept number status; details of the announcement include the called number and, possibly, a new number. These announcements are applied to regular intercept calls, both ANI and ONI.

Announcements are constructed from 0.5-second digits and 1.5-second phrases. About 50 milliseconds of "quiet" time is provided between sequential announcements on a track and is synchronized between tracks. Announcement connection track switching is performed within this period to prevent "clipping" of announcement phrases. Reference 3 contains a list of the phrases and digits recorded on the announcement tracks.

Timing of announcement tracks is indicated on master scanner ferrods. At the end of each 0.5- and 1.5-second announcement phrase (i.e., during the quiet period), an IO25 interrupt program detects the indication and sends connection orders to the switching network. The identities of the trunks to be connected were previously calculated by a base level program and stored in POBs (Section 3.4.1). When the IO25 interrupt program completes sending the connection orders, it indicates to the base level program that the next POB orders can be constructed. Base level progress mark routines perform this by using the intercept number status to reference the appropriate *connection table*. Each table entry defines the actions to be taken after each 0.5-second period. The table is indexed by the announcement phrase count stored in the time slot word (Section 3.3.1). The information obtained from the table indicates one of the following:

- (*i*) no action (e.g., in the middle of a 1.5-second phrase);
- (*ii*) the identity of the new announcement track;
- (*iii*) a specific reference to called number, new number, NPA, or geographic location information which is required to determine the next track; or
- (*iv*) completion of the announcement sequence.

4.6 Operator position connections and actions

Operator positions are provided in AIS installations to assist callers by performing functions not accomplished by automatic equipment. Four types of positions may be provided:

- (*i*) *ONI operator positions,* used to manually key called number information;
- (*ii*) *trouble operator positions,* used to serve trouble intercept calls;
- (*iii*) *CIB operator positions,* used to perform a variety of functions such as assisting callers not satisfied with the recorded announcement; and
- (*iv*) *service assistant consoles,* used to handle problems which cannot be resolved by CIB operators.

These positions are provided at home and remote AICs as shown in Table V.

4.6.1 ONI operator functions

In an AIS installation with a large volume of ONI traffic, ONI operator positions may be furnished to receive calls from local offices not equipped with ANI. The ONI operator asks the caller the number he dialed and then keys the called number into the AIC via an MF keyset. The MF receiver used is permanently associated with the operator position; hence, it is *not* necessary to select an MF receiver

Table V — Types of operator positions provided with AICs

Position	Home AIC	Remote AIC
CIB	Must be provided	Never provided
ONI	Optionally provided	Optionally provided
Trouble	Optionally provided	Optionally provided
Service Assistant	Must be provided	Never provided

and connect the ONI position to it through the switching network. Digit receiving and analysis programs store the keyed digits. When the ST digit is received, the ONI operator position is disconnected and thereafter the call is handled like an ANI call.

ONI calls are received from local offices on three types of trunks:

(*i*) one-class ONI trunks;
(*ii*) three-class ONI trunks; and
(*iii*) trunks from ANI/ONI concentrators.

All calls on one-class ONI trunks are routed directly to ONI operator positions. On calls received on three-class ONI trunks, a directed scan (Section 3.4.3) is performed to determine the call class. If a regular ONI call is indicated, the call is routed to an ONI operator position. Otherwise, the call is handled like an ANI call of the same class.

Calls received through ANI/ONI concentrators are first connected to MF receivers. For ONI calls, the concentrator outpulses a digit sequence identifying the call class. Regular ONI calls thus identified are routed to ONI operator positions. Other calls are handled like ANI calls.

4.6.2 Trouble operator functions

Calls to important customer lines, such as hospitals or firehouses, may be intercepted and routed to trouble operator positions if the line has been "plugged up" because of a trouble. The operator consults a printed bulletin and notifies the caller of the disposition of the line. In most cases, the trouble operator offers an alternative method of reaching the desired customer. When trouble operator positions are not furnished or are unoccupied, this function can be performed at CIB operator positions.

4.6.3 CIB operator functions

CIB operator positions are provided only at home AICs. They serve the following functions:

- (*i*) further assisting callers on calls receiving recorded announcements or requiring immediate connections to operator positions;
- (*ii*) keying the called number and providing a verbal response for ANI-failure calls;
- (*iii*) handling directory assistance calls; and
- (*iv*) performing ONI and/or trouble operator functions when such positions are not furnished at an AIC.

Calls at a remote AIC may also require CIB operator assistance. Such calls are routed to a CIB operator at a home AIC, using the home AIC as an intermediate switching point. Note, however, that ONI calls to a remote AIC must be served by ONI operators at that AIC.

When a call is to be connected to a CIB position, a data outpulser is connected to the outgoing trunk to the position and a data message is transmitted (as described in Section 3.5.2). This results in a console display containing all available details of the call. When transmission is complete, the outgoing trunk goes off-hook, signaling call processing programs to disconnect the outpulser and connect the associated incoming trunk.

A CIB position is equipped with trunk access to each AIC it serves (one home AIC and up to four remote AICs). Using these trunks, the CIB operator may obtain further information about an intercept call. To do this, the operator depresses a key, causing the trunk to the destination AIC to go off-hook. The call processing programs at this AIC recognize the change of state and connect the incoming trunk from the CIB operator position to an MF receiver. The operator then keys a sequence of MF digits appropriate to the request.

In addition, some operator actions cause MF digits to be outpulsed automatically by circuits within the position. For example, when a CIB operator vacates a position, removal of the headset causes a digit sequence to be outpulsed indicating that the position is now unoccupied.

4.6.3.1 Post-announcement assistance. Callers not satisfied with the recorded announcements may wish assistance from an operator. In such cases, the incoming intercept trunk is connected to a CIB operator position 4.5 seconds after completion of the announcement. The console display indicates to the CIB operator the call class (regular intercept or operator-keyed), the called number and NPA, the call status, the identity of the originating AIC, and the new number and NPA, if available. While assisting the caller, the CIB operator may desire an announcement or another display of information concerning this (or another) intercepted number. The operator seizes a trunk to an AIC and keys the digit sequence (shown in Table IV). The class digit indicates whether an announcement or display is desired. A file lookup is performed and an appropriate announcement or display results. The display is handled similarly to the initial display. The announcement, which is heard by both the caller and the operator, is connected to the incoming trunk from the CIB operator position. The announcement is disconnected when it is complete or when the operator trunk goes on-hook.

4.6.3.2 Other intercept assistance. Certain types of calls require immediate connection to a CIB operator position. For example, when ONI positions are not furnished or are all unoccupied, CIB positions may receive ONI calls. This function is quite similar to that performed by ONI operators (Section 4.6.1). When an ONI call has been connected to a CIB position, the console display indicates the call class (ONI) and the identity of the originating AIC. Unlike ONI operation, incoming trunks from CIB positions require connection to a hunted MF receiver in a second time slot. This second connection is controlled from a second CIPR. When operator digit keying is complete, the digits are transferred from the originating register to the CIPR associated with the incoming intercept trunk. The facilities used for the second connection are then released.

4.6.4 Transfer of call to service assistant

If the caller requires assistance beyond that provided by the CIB operator, the call may be connected to a *service assistant* (SA). To do this, the CIB operator depresses the TO SA key. A second connection is then established between the incoming operator trunk and the SA trunk. The CIB operator may release from the call by operating the POSITION RELEASE key. This causes a transfer of call data from the CIPR associated with the original connection; a new connection is established between the incoming intercept trunk and the outgoing trunk to the SA console. The CIB position is made available for another call.

4.7 Call disconnect

Certain types of trunk supervisory changes (Section 4.2) stimulate call disconnects. In general, when a trunk goes on-hook, the associated call is disconnected. The additional actions taken in response to an on-hook depend on the type of trunk that changed state, and the type and supervisory state of the trunk to which it is connected, if any.

When a connection has been established between an incoming trunk (from a local office or a remote AIC) and an outgoing trunk to an operator position, an on-hook report on either trunk causes the call to be disconnected. If the incoming trunk goes on-hook first, all associated facilities, hardware and software, are idled except for the operator position. The outgoing trunk to the operator position is placed in a made-busy state until it goes on-hook. Normally, this on-hook occurs momentarily; however, should the position have been manually made busy, the trunk will not go on-hook until the position is again ready to receive calls.

The operator may release a call from the position manually by operating the POSITION RELEASE key. This causes the outgoing trunk to the operator position to go on-hook first. In this event all facilities are released except the CIPR and the incoming trunk. The CIPR is held until the incoming trunk goes on-hook or until two minutes have elapsed. After two minutes, a permanent signal on the incoming trunk is suspected, and maintenance programs are notified to perform testing of the trunk.[3]

On all connections other than caller to operator, disconnect actions are based only on the state of the incoming trunk. For example, when an incoming trunk connected to an MF receiver goes on-hook, the supervisory state of the MF receiver trunk is ignored. The call is simply disconnected and all associated facilities are idled immediately. Similarly, if an incoming trunk goes on-hook when it is not connected in a time slot, associated facilities are released and the call is removed from any queue in which it may be entered.

4.8 Ancillary call processing functions

Several functions are performed by call processing programs which are not central to the job of processing intercept calls. These functions are incorporated into system design in a way that does not interfere with expedient handling of intercept traffic.

4.8.1 Service observing

Service observing provides a means for monitoring the performance of an AIS, in general, and of CIB operators, in particular, to determine the quality of service offered. This allows telephone companies to ensure that intercept service standards are being achieved.

One call (at a time) to a CIB or ONI operator position may be monitored at a remote *service observing* (SO) *desk*. The observer receives via a data link three types of information concerning the monitored call. These are:

(*i*) all data which are displayed at the operator position; these are printed on a paper tape at the SO desk;
(*ii*) reports of all actions taken by the operator such as MF keying; these are shown on lamps and paper tape at the SO desk; and
(*iii*) the conversation which transpires between the caller and an operator and between the caller and a service assistant, if any.

When a period of monitoring calls is to begin, the service observer indicates this to the AIC by key operation. In response, when monitoring is active and the SO desk is idle, the progress mark program selects a call to be monitored from a CIB operator queue. At this time a *position called* signal and all data to be displayed at the CIB position are sent to the SO desk. After the incoming trunk is connected to the operator position, a *position attached* signal is sent to the SO desk. A special, three-way connection[3] is also established to allow the observer to monitor the conversation. All additional actions taken on the call are transmitted to the observer until either the call is disconnected or the observer releases the position. Another call can then be sent to the SO desk for monitoring.

4.8.2 Operator-keyed trouble reports

Problems encountered by CIB and ONI operators while attempting to assist callers may point to trouble conditions in the AIS or connecting offices. If these conditions are reported quickly and accurately to maintenance personnel, the reports can be used in conjunction with other indications to resolve the problems. For this reason, these operators are able to key reports of ten different types of trouble conditions into the AIC. The format of the digit sequence and a list of the trouble conditions are given in Table IV and Table VI, respectively.

After the reception and analysis of the digits, a base level progress mark program generates a printout on a maintenance teletypewriter. This printout contains the information keyed by the operator, the type and number of the operator position, and all pertinent information about the connection involved.

Table VI — Trouble conditions reported by operators

Keyed Code	Description
010	Caller reports or operator detects difficulty in hearing or being heard.
011	Operator detects abnormal noise on connection.
012	Caller reports or operator detects an announcement phrase that is not clear.
013	Partial or mutilated display.
014	Position seizure with valid display. No caller response to operator challenge and no room noise heard.
015	Caller on line with display from previous call, and no zip tone.
016	More than one caller on line.
017	Operator hears another operator on position.
018	For no apparent reason, the connection between the caller and the operator is broken during conversation.
019	Code assigned for local use in analyzing special trouble conditions.

4.8.3 Operator training

Facilities are provided at the AIS for training CIB operators to perform all appropriate functions. Up to two positions at a CIB may be equipped on a plug-in basis with standard 12-key *Touch-Tone* card dialers. Activation of training is indicated to the program by a start-training message which is entered via the card dialer.

On each training call, the trainer enters via the card dialer a number used only for operator training. The digit sequence is received and analyzed and the call is routed directly to the trainee with the proper display. All call conditions normally encountered by a CIB operator may be simulated in this manner. On a call where an announcement is normally required, a file lookup is performed but no announcement is constructed; the post-announcement display is simply sent to the position. Training continues until a stop-training message is received.

4.8.4 System status indicators

A *lamp signal cabinet* is provided at both the CIB and ONI position locations to indicate to the operators traffic and trouble conditions in the AIS.[3] For example, whenever calls are waiting for operators, a lamp is lighted indicating the condition.

These lamps are lighted and extinguished by the call processing programs by means of data outpulsing. An outpulser is connected to a data trunk outgoing to the lamp signal cabinet. A code is loaded into the data outpulsing area and outpulsed as described in Section 3.5.2. Decoding circuits at the cabinet then light and extinguish the lamps indicated.

V. CALL STORE AUDITS

5.1 Audit philosophy

The AIS program depends heavily on data in its call store memory to record the states of calls and of system hardware and software resources. Hardware errors, program bugs, and incorrect manual operations can mutilate data in call store, causing calls to be mishandled and leaving system resources in unusable states. In addition, data errors could propagate throughout the call store data causing service to degenerate, possibly creating the need for a system initialization (Section VI) to recover from the errors.

Some of these errors are eliminated by defensive programming techniques. However, some types of errors would require a prohibitive amount of processor time to prevent and still other more subtle errors could not be readily found using defensive programming techniques. Hence, *audit programs* are needed to protect the AIS from the effects of data mutilation. These programs detect and correct errors in the critical transient and semipermanent call store memory areas such as call-in-progress registers, queues, records of connections in the network, and trunk translation data.

The audit programs employ the philosophy of using the external hardware states and nongeneric data from the files as references. This eliminates the need to make any assumptions about the correctness of call store data before auditing and simplifies corrective techniques since normally many more errors occur in call store than in the external data. For example, the audit of the basic call processing linkage structure starts by reading the connector and scanner time slot memory to determine the equipment locations of the trunks actually connected. Call processing software records are then compared with the time slot readout information for consistency. If errors are found, audits report them via teletypewriter messages and correct them by returning the call store memory and hardware to consistent, viable states. It should be noted that certain inconsistencies reflect errors in the external information and the audits can detect and report these. A high rate of such errors can trigger maintenance action on the associated peripheral circuit.[3]

Other audit programs depend on the nongeneric data stored on disc file or on trunk supervisory state scanning for external information. Since various audits use different reference points and limit the areas of memory which they audit, the combined action of several audits, some triggered by others, may be necessary to clear a data problem and halt error propagation.

5.2 Types of audits

Audits may be divided into three groups: *transient data audits*, *semipermanent data audits*, and *timeout audits*.

5.2.1 Transient data audits

Transient data audits generally detect and correct linkage errors in call store registers associated with calls in the system. This is done by comparing redundant information. Redundant information is that which is represented in different forms within the call store memory for call processing expediency or indicated by the states of peripheral circuits; this is distinct from semipermanent information for which a backup copy exists. A brief description of a transient data audit follows.

The *time slot memory audit* checks the correctness of the basic call processing software linkages (Section 3.9). A word is read from the connector and scanner time slot memory. The equipment locations of the connected trunks are also read from the call-in-progress register indicated by the time slot word. Then, the facility control words corresponding to these trunks are read. If the linkages, as shown in Fig. 15, are not consistent, the call is disconnected and the associated trunks and CIPR are released. In this way, the call records are checked using the redundancy inherent in the call store linkage structure and

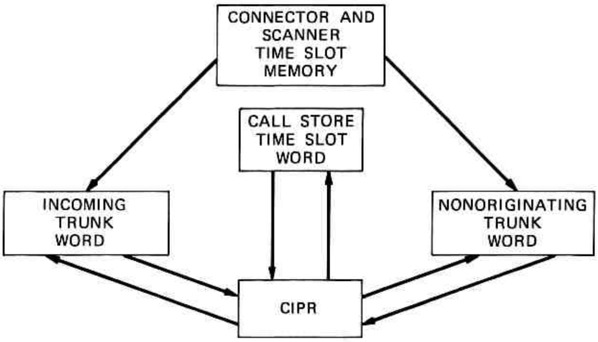


Fig. 15—Abstract representation of linkages audited by time slot memory audit.

the peripheral circuits. In addition, the audit also checks the status of the time slot facility selection words with the contents of the time slot memory; any inconsistencies found are corrected.

Three other transient areas of call store are audited. The *nonoriginating equipment audit* compares the facility selection words for all nonoriginating equipment with the actual state of the associated equipment to prevent equipment from becoming lost to the system or multiply used. The *queue audit* checks the linkage structure of calls in queues to prevent these calls from being mishandled. Finally, the *connector and scanner maintenance words audit* checks call store words associated with the state of the connector and scanner circuits with the actual states of the circuits.

5.2.2 Semipermanent data audits

Semipermanent data audits detect and correct errors in the areas of call store where nongeneric data particular to an AIC are stored. This is done by comparing call store records with a disc file backup record. For example, call store records of NNX-ABX translations are periodically compared with a disc backup record. When a disagreement is found, the file record is assumed correct and call store is corrected.

When nongeneric data are altered via plant changes, the same audits are used to regenerate semipermanent call store data without disrupting service. This is accomplished by overwriting only that information which has been changed on the disc file backup record (Section 7.2). In addition, the semipermanent data audits are used to place nongeneric data in the call store memory during phase B and C system initializations (Section 5.4). The use of the semipermanent data audits and their relation to other operational programs is shown in Fig. 16.

Another audit compares the nongeneric data on the two disc files and makes some internal consistency checks on the data. If a discrepancy occurs, the error is reported and manual correction is required; semipermanent data audits are suspended until the errors are corrected. This audit enhances the integrity of the disc file backup records of nongeneric data.

5.2.3 Timeout audits

Timeout audits monitor continuous use (without release) of a software or hardware facility. If a predetermined time period is exceeded, a timeout audit either verifies the correct use of the facility or releases it. For example, a check is made to determine if a CIPR is active for a period of time longer than the maximum holding time for the type

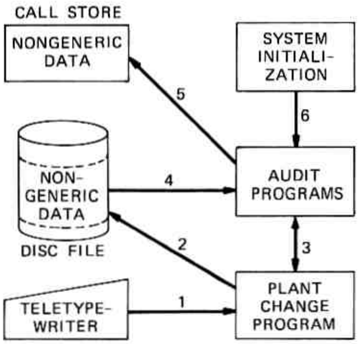


Fig. 16—Interrelationship between audit, plant change, and system initialization programs.

of associated call. If the time is exceeded, the system may have lost control over that CIPR and action is taken to verify the CIPR usage or correct the situation.

5.3 Audit control

Audits are executed once every main program loop. Upon gaining control from the executive control program, the audit monitor causes any requested client audit to be executed based on a predetermined order of priority. The progress of a client audit is then controlled through a segmenting structure until it has completed.

Individual client audits are requested by the monitor on a routine, periodic basis. The rate at which different audits are run was chosen to maximize the call processing, administrative, and maintenance needs of the system while minimizing the risk of data mutilation and error propagation. In particular, transient data audits are executed at least once every 40 seconds. At the chosen rates, routine audit requests do not overlap, and an audit segment is executed during about half of the main level loops.

Semipermanent data audits are routinely requested at a rate lower than transient data audits. This is done to prevent excessive file access requests which could delay call processing and hinder use of the disc files by file administration routines. To compensate for this less frequent routine requesting, the transient data audits typically request the semipermanent data audits when suspect semipermanent data are encountered. Other system programs also request audit programs if suspect call store data are encountered.

Most audit programs can be initiated manually via teletypewriter request. Options are available for controlling the detail of error printout, the extent of the audit, and whether the audit is to be executed once or repetitively. In addition, messages exist to inhibit and restore all detailed error message printouts and the execution of all audits or a particular client audit.

Executing audits as a major main program loop function rather than as a maintenance program allows the audits to run as asynchronously as possible with respect to all other system programs. This enhances the possibility of detecting all errors, regardless of their sources, and before they propagate.

5.4 Audits during system initialization

The audit programs are used during phase B and C system initializations (Section VI) to place the nongeneric data in call store memory. This is accomplished by running the semipermanent data audits consecutively until the semipermanent portions of call store are completely reinitialized from a disc file backup record. This consecutive running of the audits is called *stitching* the audits together. The stitched mode execution of the audits is requested by the initialization program. During this period, no other system programs are running except those needed to perform file block reads. After all the requested audits have been run, control is returned to the initialization program.

VI. SYSTEM INITIALIZATION

In a complex program-controlled system such as AIS, hardware or software malfunctions can occur which result in improper call processing actions. While circuit redundancy and software corrective techniques such as audits are applied to minimize the effects of such actions, an occasional problem arises which is so serious that severe recovery action, known as *initialization,* is necessary. In addition, similar actions are used for initial startup of the system.

When initialization occurs, all other system activity is suspended for the duration of the initialization. The severity of an initialization determines the degree to which intercept service is disrupted. Three levels of severity, or *phases,* are provided so that increasingly drastic initialization actions can be performed until proper operation is resumed. This structure minimizes over-reaction to troubles which can be cleared without seriously disturbing calls in the system.

In general, the initialization strategy provides a working configuration of a control unit and peripheral units, and brings the call store memory into agreement with the states of system equipment. A control unit switch accompanies all but manually requested initializations so that a potentially faulty control unit can be switched offline. This control unit then performs all initialization actions. The action taken to initialize the call store memory depends on the phase being executed. Transient, stable, and semipermanent areas of call store can be affected (Section 3.1).

6.1 Causes of initialization

Initialization actions can result from both automatic and manual sources. Automatic requests for initialization can be triggered by failure of either hardware or software checks. Among these are:

(*i*) A circuit which times the length of the main program loop and causes an initialization if a maximum time is exceeded. This provides a defense against infinite program looping.

(*ii*) A program which checks that the main program monitors are executed in the correct sequence and causes an initialization if improper sequencing is detected.

(*iii*) A program which frequently tests the online control unit to see that its circuits are operating properly and causes an initialization if a test fails while the control units are not operating in synchronism.

Thus, in general, an initialization results from program insanity or from failure of the online control unit when the control units are not operating in synchronism. Troubles in peripheral units do not normally result in automatic initialization.

Manual requests for initialization are made via the emergency action panel at the maintenance center. Three keys must be operated in sequence so that inadvertent operation of a single key does not result in initialization. In addition, two keys are provided to permit the following options:

(*i*) immediate execution of the most severe initialization phase and zeroing of long-term call store (stable) records not affected by automatic initializations, and

(*ii*) placing the system in a noncall processing mode during which file backup records of nongeneric data can be initialized or restored from a punched paper tape record (Section 7.2).

6.2 Phase structure

The initialization phase structure and sequencing strategy represent a compromise between maximizing speed of recovery and minimizing disruption of intercept service. Three phases are provided: A, B, and C. They are designed so that most troubles are cleared by a phase A initialization. If phase A does not recover the system, the more drastic phase B initialization may be triggered. Similarly, if phase B is unsuccessful, a phase C initialization may be executed. If an initialization is triggered within about 20 seconds after a previous initialization is completed, the initialization phase may be escalated.

When a rapid sequence of initializations occurs, phase A is executed once, phase B is executed twice, then phase C is executed twice. The sequence ABBCC is then repeated until recovery is successful. During each initialization, a different configuration of the control unit with each peripheral unit is established so that a faulty communication path does not prevent a successful initialization. For example, data on the disc files are read during phases B and C to initialize semipermanent data in call store. On four consecutive initializations (two phase B, then two phase C), if both disc files are accessible, all four control unit/disc file access paths are used.

6.2.1 Phase A initialization

The lowest level initialization is phase A. This phase lasts less than 100 milliseconds and cannot be triggered by manual request. After a phase A initialization, the program is restarted from a fixed point and all maintenance activity is aborted. While no calls are disconnected, the delay of certain critical input-output tasks may result in problems which are later detected and cleared by call processing and audit programs; this might result in the mishandling of a few calls.

6.2.2 Phase B initialization

Phase B is the second level of initialization. During this phase all but operator-to-subscriber calls are disconnected. Phase B can be requested manually or automatically, and lasts from four to eight seconds. This short duration is made possible because almost all peripheral circuits are electronic and, therefore, can be initialized at program speeds. For example, a call in the network can be disconnected in tens of microseconds.

When a phase B initialization is executed after a phase A has failed to recover the system, the most likely problem is mutilation of call store data. Consequently, semipermanent areas of call store are initialized during phase B by the audit programs (Section 5.4) using a disc file backup record of nongeneric data. Several rapid checks of file operation, data validity, and plant changes in progress are performed to minimize the possibility of using incorrect data. Using the results of these checks, a preferred file is selected. If plant changes were in progress they are removed from the system. If neither file can be used, the semipermanent data previously in call store remain there.

All transient areas of call store are zeroed during phase B. Some call records are re-established when operator-to-subscriber connections are identified. This is done by reading the equipment numbers of connected trunks from the connector and scanner time slot memory. These are compared with the semipermanent trunk data which define the types of trunks assigned at each network equipment location. Transient call records are then re-established for all calls which are identified as operator to subscriber. Transient areas of call store are also initialized to reflect the service states of peripheral circuit and operator positions.

The order of activities during phase B is critical. For example, subscriber-to-announcement calls must be disconnected quickly to prevent improperly repeating the current announcement phrase or digit. In summary, the sequence of actions during phase B is:

(*i*) initialize peripheral circuits;
(*ii*) disconnect subscriber-to-announcement calls;
(*iii*) zero transient areas of call store;
(*iv*) initialize semipermanent areas of call store;
(*v*) disconnect all but operator-to-subscriber calls; and
(*vi*) initialize state data for peripheral circuits and operator positions.

The main program loop is then entered at a fixed point (at the beginning of the call processing monitor).

6.2.3 Phase C initialization

Phase C is the highest level of initialization. During this phase, all calls are disconnected; otherwise, it is much like phase B. However, phase C is slightly shorter in duration than phase B since subscriber-to-operator calls need not be identified and maintained.

The primary differences between phases B and C result from the two options which may be invoked when phase C is requested manually from the emergency action panel. If either option is requested, phase C is entered even when no recent initialization has occurred.

The *stable* option causes stable data areas of call store (Section 3.1) to be zeroed. This results in loss of records such as time of day and plant and traffic measurements. No automatic initialization of stable data is provided because troubles in stable data are unlikely to result in improper call processing actions, and because such data cannot be automatically re-established. Since the initialization is requested manually, the requestor can simply re-enter the time of day via a teletypewriter following the initialization.

The *tape* initialization option is used when the integrity of both file backup records of nongeneric data is in doubt and the records must be restored from a backup punched paper tape. In this case, following initialization, the system is placed in a *tape mode* of operation in which no call processing is performed (Section 7.2).

Entry of the punched paper tape at a teletypewriter typically requires 10 to 20 minutes. After this period, normal operation is resumed. Tape mode is also used for initial system testing, since many maintenance and administrative functions can be executed without reference to nongeneric data. This mode also permits initial disc file loading of such data.

A summary of the initialization phase structure is shown in Table VII.

VII. ADMINISTRATIVE FUNCTIONS

7.1 Plant and traffic measurements

In administering an AIS, system facilities must be adequately engineered and maintained if satisfactory service is to be provided. *Plant measurements* are made which reflect the health of all system components by recording counts of system activities, failures, and maintenance states. These measurements are automatically printed at the maintenance teletypewriters and reset once every 24 hours.

In addition, facilities are provided at the AIC for recording *traffic measurements* needed for operator force adjustment and traffic engineering and equipment administration. These measurements consist of *peg counts* which are incremented each time an event occurs and *usage counts* which are incremented every 10 seconds. The counts for each type are printed half hourly at a dedicated traffic data receiving teletypewriter. Those counts associated with operator force adjustment are printed additionally at a dedicated traffic force adjustment teletypewriter.

Table VII — Summary of initialization phase structure

Phase (And Options)	Request Sources	Approximate Execution Time (Seconds)	Effect On Call Processing
A	Automatic	0.1	Very little
B	Automatic or Manual	4.0–8.0	All but caller-to-operator calls disconnected
C	Automatic or Manual	4.0–8.0	All calls disconnected
C Stable	Manual	4.0–8.0	All calls disconnected
C Tape	Manual	0.5	All calls disconnected; call processing inhibited

7.2 Administration of nongeneric data

Prior to placing an AIC in service, nongeneric data describing the AIC are punched manually on a paper tape and are placed on one track of each of the two disc files and in call store by reading in this tape from a teletypewriter while in tape mode following a phase C tape initialization (Section 6.2.3). Thus, backup copies of the nongeneric data are maintained on the files and also on punched paper tape.

When changes are made in equipment or options, the associated nongeneric data changes are made online by a telephone company employee via a teletypewriter. In performing some of these changes, such as adding or deleting trunks, coordinated changes to the physical equipment may be required. Plant changes are performed by entering a series of teletypewriter messages. As new data are received, the corresponding changes are made in the backup data on one file.

Due to the sensitivity of the system to errors in the nongeneric data, the program makes checks for data completeness and reasonableness as the new data are being received. If the data are found to be incorrect, the request is rejected and a teletypewriter message is printed containing a code which indicates the reason for rejection.

When all plant changes have been entered, a message indicating completion signals the audit programs (Section 5.2.2) to *regenerate* the nongeneric data in call store; this results in making the new data operational. Following regeneration, a period of testing ensues to insure proper system operation with these data. If a trouble condition is detected, the original data can be restored from the unmodified file on manual request. After the new data are found to be acceptable, the telephone company employee requests the program to punch a backup paper tape which incorporates the new data. During this process, the program provides a printed record of the nongeneric data in the same format as it appears on the tape; thus, a listing of the tape for use as an office record is available. Finally, a request is made to read the nongeneric data from the modified file and write it onto the other file, bringing the two backup records into agreement.

VIII. SUMMARY

The foregoing discussion has provided the organization and structure of the operational programs, an explanation of the various functional tasks performed by the operational programs, and a detailed description of the call processing functions to show the manner in which autonomous circuits and software interact to process intercept calls. Novel features of the AIS have been emphasized: a single generic program, use of autonomous circuits, use of both hardware and software as the basis for audit techniques, and the methods of storing and administering nongeneric data. The authors have attempted to provide insight into the techniques and considerations used in the development of operational programs.

REFERENCES

1. C. J. Byrne, W. A. Winckelmann, and R. M. Wolfe, "Automatic Intercept System: Organization and Objectives," B.S.T.J., this issue, pp. 1–18.
2. J. A. Harr, Mrs. E. S. Hoover, and R. B. Smith, "Organization of the No. 1 ESS Stored Program," B.S.T.J., *43*, No. 5 (September 1964), pp. 1923–1959.
3. P. J. Brendel, W. K. Comella, R. N. Markson, P. J. Moylan, and J. Orost, "Automatic Intercept System: Peripheral Circuits," B.S.T.J., this issue, pp. 71–106.
4. J. W. Hopkins, P. D. Hunter, R. E. Machol, J. J. DiSalvo, and R. J. Piereth, "Automatic Intercept System: File Subsystem," B.S.T.J., this issue, pp. 107–132.
5. J. H. Carran, K. E. Greisen, W. G. Hall, and D. J. Wells, "Automatic Intercept System: Administering the Intercept Data Base," B.S.T.J., this issue, pp. 133–153.
6. T. E. Browne, T. M. Quinn, W. N. Toy, and J. E. Yates, "[No. 2 ESS] Control Unit System," B.S.T.J., *48*, No. 8 (October 1969), pp. 2619–2668.
7. M. E. Barton, N. M. Haller, and G. W. Ricker, "[No. 2 ESS] Service Programs," B.S.T.J., *48*, No. 8 (October 1969), pp. 2865–2896.
8. H. J. Beuscher, G. E. Fessler, D. W. Huffman, P. J. Kennedy, and E. Nussbaum, "[No. 2 ESS] Administration and Maintenance Plan," B.S.T.J., *48*, No. 8 (October 1969), pp. 2765–2815.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 53, No. 1, January 1974
Printed in U.S.A.

Automatic Intercept System:

Peripheral Circuits

By P. J. BRENDEL, W. K. COMELLA, R. N. MARKSON,
P. J. MOYLAN, and J. OROST

(Manuscript received April 20, 1973)

The Automatic Intercept System forms intercept announcements from phrases recorded on a magnetic drum by means of a sequence of connections through a time division switching network. Automatic fault recovery and diagnostic maintenance features include a trunk test circuit that enables maintenance programs to look in at the system on trunk pairs and test the autonomous scanner and time division network as well as the trunk and service circuits.

The Automatic Intercept System time division switching network, announcement machines, operator positions, trunk and service circuits, and the maintenance strategy for these units are described in this paper.

I. INTRODUCTION

An Automatic Intercept Center (AIC) consists of a processor and associated peripheral circuits as shown in Fig. 1. An Automatic Intercept System (AIS) may consist of up to four of these AIC's.[1] One AIC is designated home and the others are termed remote. The home AIC has associated with it a Centralized Intercept Bureau (CIB) which may be located in either the same or a distant building. The peripheral circuits of the AIC and CIB and the maintenance programs for these circuits are described.

1.1 Peripheral circuits of an AIC

Trunk circuits interface local offices, operator positions, and service circuits with the time division network of the AIC. Maintenance programs automatically test the trunk and service circuits using facilities of the trunk test frame.

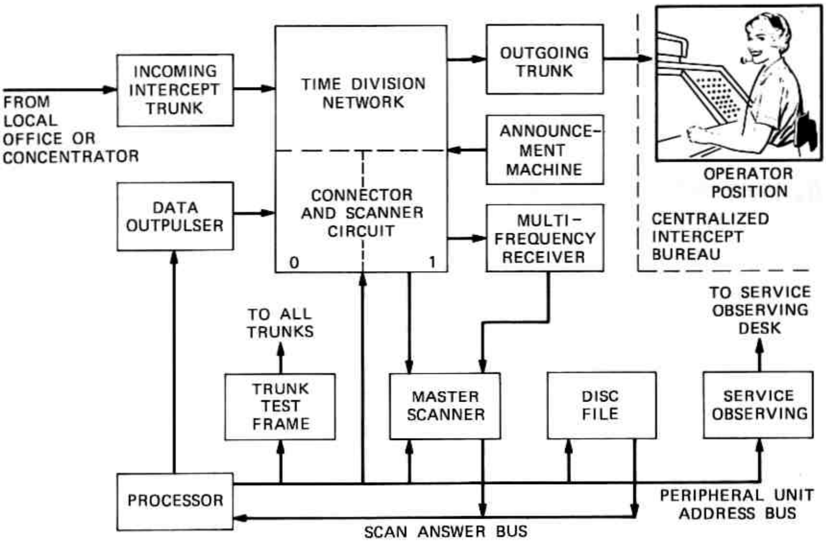


Fig. 1—Automatic Intercept Center.

An announcement machine contains prerecorded words and phrases which are combined to form a spoken message notifying the customer of the status of an intercepted number.

An AIC may be equipped with either one or two time division networks. A time division network is comprised of two connector and scanner circuits, each connector providing 32 talking time slots. The scanner autonomously scans and records the on- or off-hook status of all trunks, reporting changes of state to the processor via the master scanner.

The master scanner used by the processor to input data is the same as that used in the No. 2 Electronic Switching System (ESS) and has been previously described.[2]

1.2 Maintenance programs

Failures in the peripheral circuits are automatically detected through hardware and program checks. In general, the strategy is similar to that used by No. 2 ESS.[3] Major circuits are duplicated and if a failure occurs a working configuration is automatically established in which the faulty circuit is removed from service. Automatic diagnostics then locate the trouble within the faulty circuit to expedite repair, minimizing the interval during which the system operates on only half of a duplicated pair.

II. TRUNK AND SERVICE CIRCUITS

Each trunk circuit in the AIC is packaged on a single plug-in circuit pack. The trunk circuits interface the tip and ring of intercept lines, service circuits, and operator position to the time division network. The switching network is distributed; most of the time division switches (TDS) are contained in the trunk circuits. Trunk test arrangements provide access to the tip and ring allowing maintenance programs to test through the trunk circuit into the network. A busy condition can be returned to the local office preventing calls on the trunk while it is under test or out-of-service.

The AIC has both incoming (originating) and outgoing (nonoriginating) trunks. The trunks are classed according to the type of service which they perform. A "type number" in a call store area dedicated to each trunk indicates the class of the trunk. Table I lists the various originating and nonoriginating trunk types.

2.1 Originating trunks

Traffic to the AIC is generated when a call terminates at an intercepted line in a local office. The line number is identified and outpulsed to the AIC which determines the status of the number from

Table I — AIS trunk types

Type Number	Trunk Type
Originating	
01	Incoming Intercept ANI
02	Incoming From CIB Operator
03,12,13,14	Incoming Directory Assistance
04	Incoming From Remote AIC
05	Incoming Intercept 3 Class ONI
06,07	Spare
10,15,16,17	Incoming Intercept 1 Class ONI
11	Incoming Operator Training
Nonoriginating	
20,31,33,34,35	Spare
21,22	OGT to Trouble Operator
23	OGT to Supervisor
24	OGT to ONI Operator
25	Operator Training *Touch-Tone®* Receiver
26	OGT to Home AIC
27	OGT to CIB Operator
30	MF Receiver Trunk
32	Data Outpulser
36	Miscellaneous and Test Trunks
37	Unequipped or Unassigned Trunk Location

information stored in the disc file and returns an appropriate announcement to the customer. Under this method of operation, called Automatic Number Identification (ANI), the number *actually* reached by the customer is announced.

There are three main classes of intercept: machine, trouble, and regular. Unequipped line numbers are placed on machine intercept; essential lines which are temporarily in trouble are placed on trouble intercept; changed or disconnected numbers are placed on regular intercept. On the ANI trunk these classes are indicated by a class digit prefixed to the called number by the local office.

2.1.1 ANI trunk

Figure 2 shows a typical trunk circuit used in the AIC. Each trunk has a unique address based on its physical appearance on the network. Trunks appear in groups; vertical and horizontal numbering is used to distinguish trunks within a group. In order to establish a connection, a coincidence of pulses is required on the G, V, and H (group, vertical, and horizontal) leads from the connector translator. This causes the bilateral switch to close completing the connection to the time division network. The connection between the trunk and the time division bus is established repetitively for a short period of time called a time slot. Energy is interchanged by the resonant transfer principle[4] with another trunk circuit which is selected during the same time slot.

On-hook and off-hook information is received over the trunk pair and detected by the trunk current detector. Any transient hit on the trunk is filtered by this detector. A different time slot, known as the

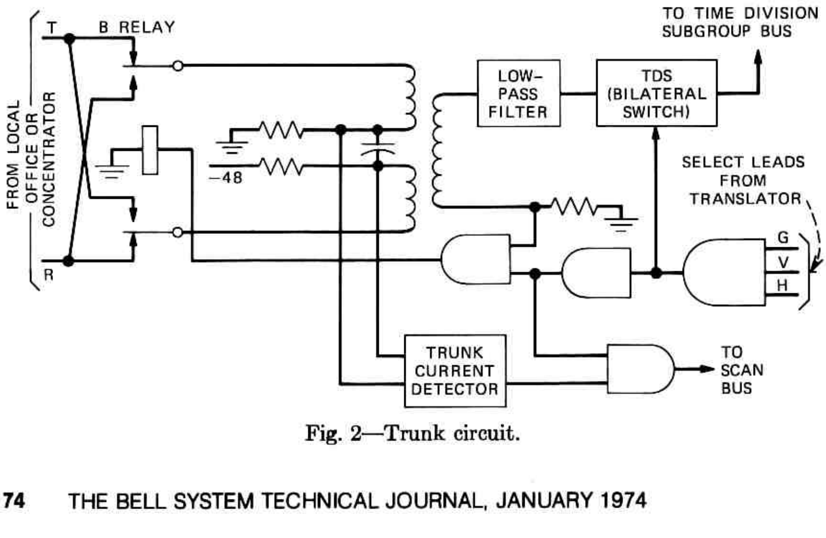


Fig. 2—Trunk circuit.

scan interval, is used to pass this information over the scan bus. In contrast to other switching systems, the talk selection and scan selection are integrated into a common set of leads (G, V, H). This results in a less expensive and more reliable arrangement. However, this also made it necessary to inhibit the bilateral switches during the scan interval to prevent a double access which would affect the resonant transfer and cause a high-pitched whistle in the talking connection.

When an intercepted line is reached, the local office transmits the called number digits to the AIC by multifrequency (MF) pulsing. The ANI trunk uses reverse battery supervision. Outpulsing is started by means of a wink signal which is controlled by the B relay of the trunk. This relay operates when the trunk is placed in a time slot and sends a battery reversal to the local office. After a timed interval the multifrequency receiver (MFR), which has been connected to the trunk in the same time slot, sends a signal through the time division network, causing the B relay to release completing the wink. The start of the wink signal tells the local office the AIC has attached an MFR; the end of the wink signal indicates that the MFR is ready to receive pulsing.

2.1.2 Operator Number Identification (ONI) trunk

Some offices do not have the ability to automatically identify the called number. For these offices an arrangement is provided whereby an operator at the AIS may ask the customer for the called number and key this number into the AIC. These calls are brought into the AIC via an ONI trunk which responds to three distinguishing signals to differentiate the three classes of intercept. This signaling is accomplished by means of various battery conditions on the tip and ring conductors. Two additional scan buses are used to convey these three classes of intercept to the processor. After an off-hook has been received via the regular scan bus the processor must initiate a directed scan over the other two buses to fully identify the class of intercept. The processor will then act to set up the proper connection. Blank number intercept calls may be immediately given a blank number announcement. A regular intercept call must first go to an ONI operator. This operator will inquire as to the called number and key this number into the AIC. The call will now proceed as if it were an ANI call. To accommodate offices which are not arranged to forward the class of intercept, a one-class ONI trunk is also provided which gives all calls regular intercept treatment.

The ONI arrangement has the drawback that the announcement that the customer receives is for the number he *thinks* he dialed and not the number which he actually reached. The customer is also affected by operator keying errors. In addition, since about 60 percent of the traffic from an ONI office is regular intercept and each of these calls requires the intervention of an operator, this type of traffic is expensive to handle.

2.1.3 Miscellaneous

Other incoming trunks used at the AIC are for directory assistance, CIB inquiry, operator training, and incoming trunks from remote AIC's. The operator inquiry trunk interfaces the CIB with the AIC for such reasons as disc file interrogation and calls to the supervisor. The operator training trunk is used to simulate an intercept call to a trainee.

Since the remote AIC does not have a CIB, traffic intended for operators is sent to the home AIC which tandem switches this traffic to the CIB. This traffic is handled at the home AIC over incoming trunks from the remote AIC.

2.2 Concentrators

The AIC is arranged to accept calls from local or distant concentrators. The concentrator used with the AIC is the 23 type.[5] These concentrators have been modified to pass ANI or ONI type traffic to the AIC. For ANI traffic the concentrator repeats supervisory signals in both directions. In the case of ONI traffic, the concentrator converts the dc signal which identifies the class of intercept into MF tones and passes these tones to the AIC.

2.3 Outgoing trunks to operators

Outgoing trunks provide access to ONI positions, CIB positions, trouble operators, and the supervisor.

In most cases the circuit pack used for these trunks is the same as that used for the ANI trunk previously described. However, a special arrangement is used to provide full access to a single operator hunt group in a two-network AIC. As shown in Fig. 3, a back contact of the B relay in the trunk circuit on network 1 is used to transfer to the trunk circuit in network 0. The B relay operates when the trunk is placed in the time slot. Since only one of these two trunks is placed in a time slot at a given time, the position is automatically associated

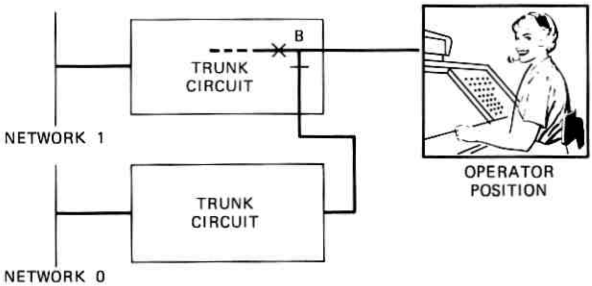


Fig. 3—Transfer arrangement for operator trunks.

with the appropriate network and the other trunk circuit is cut off from the pair.

2.3.1 Outgoing trunk to home AIC

The remote AIC has outgoing trunks to the home AIC. As previously described, these trunks provide access to the CIB operators via the switching network of the home AIC.

In contrast to the operator trunks, if the remote AIC has two networks, full access is not provided to these trunks. Instead, two hunt groups are provided, one per network. This requires a few more trunks between the remote and home AIC than a single large group would. However, two hunt groups use fewer appearances on the remote AIC networks than would be required if full access to a single large group were provided using the transfer trunks which are used for outgoing trunks to operators as described above.

2.4 Service circuits

Service circuits send and receive data and perform other specialized functions. Among the service circuits used in the AIS are the MF receiver, *Touch-Tone*® receiver, data outpulser, and audible ring circuit.

2.4.1 Multifrequency receiver

The MF receiver circuit is a one-digit receiver and is used to receive class of intercept and called number information from the local office and to receive similar information from the CIB positions. Multi-frequency pulsing uses voice-frequency tones in a 2-out-of-6 code. These tones are detected by the MFR which forwards the coded information for each digit to six ferrods on the master scanner (Fig. 4). The master scanner provides a means to input information to the

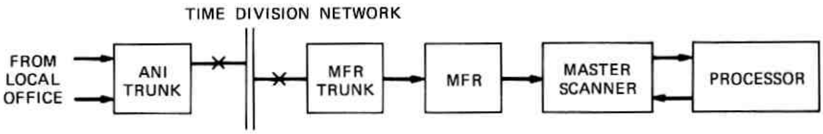


Fig. 4—ANI trunk to MFR connection.

processor. Each MFR has a dedicated register in call store memory. The input-output (IO) circuit within the processor autonomously looks for digits on these ferrods every 10 milliseconds and transfers them to the register in call store.

A diagnostic program is provided to aid in multifrequency receiver maintenance. This program, which runs under the control of a trunk and service circuit maintenance control program described below, can be requested to run by various methods. Particular types of errors encountered by the digit processing program will automatically cause a diagnostic to be run on the MF receiver involved with the error. Examples of these errors are insufficient or too many digits received for a given class of call, failure of a 2-out-of-6 tone check for each digit, or no digits received. In addition to these MF receiver error detection checks, the receivers are periodically exercised by a trunk exercise program which continuously tests most trunks in the system. The MF receiver diagnostic tests the ability of the MFR to receive all six tones and to correctly supply the wink control signal used for signaling the local office to start outpulsing. It also checks for any false outputs and makes power checks.

2.4.2 Data outpulser

Prior to connecting a customer to the operator, the processor connects a data outpulser to the outgoing trunk to the CIB position and sends all known information about the call to a display circuit located in the position. The data are encoded in a BCD format and are sent in a serial train of 94 bits using frequency-shift pulsing at frequencies of 1150 and 1850 Hz. A buffer area in call store is dedicated to each outpulser. The program loads information for each outpulser into the appropriate buffer and the IO circuit within the processor autonomously sends the information in the buffers one bit at a time over a private pair to a flip-flop located in each outpulser, as shown in Fig. 5. A 1.25-millisecond clock gates the data from this flip-flop to a register. Depending on the state of this register, the frequency-shift modulator will send one of two frequencies corresponding to a zero

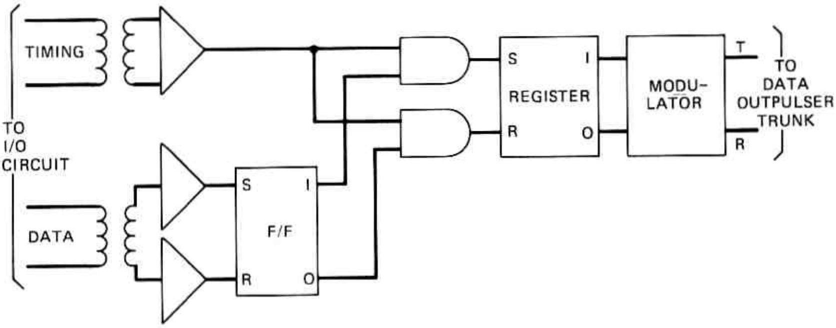


Fig. 5—Data outpulser.

or a one. After the message is received on the operator's display, the data outpulser will be momentarily connected to the operator's telephone circuit. This generates a "zip tone" or altering tone apprising the operator of a new call.

2.4.3 Miscellaneous

The *Touch-Tone* receiver is used in conjunction with the operator training circuit and is quite similar in its operation to the MF receiver circuit. This receiver is required since the *Touch-Tone* card dialer used in operator training emits *Touch-Tone* frequencies, as described in a subsequent section.

The audible ring circuit is used to send ringing tone to a customer who is waiting for an operator. This action tells the customer that his call is proceeding normally.

2.4.4 System monitor circuit

As a backup check on the ability of the AIC to process calls, a wired logic circuit known as the system monitor circuit periodically originates a call. This time period is optionally either 30 or 60 seconds in duration and the length of the calls is optionally either 5 or 10 seconds. During the time that the call is originated, this circuit will check the ability of the AIC to find an MF receiver and generate a wink signal. Each time this circuit successfully completes its call, a peg count is scored. If a call is not able to complete, an error register is scored and a major alarm sounded. This alerts the maintenance personnel if the AIC has not automatically recovered from some failure which prevents calls from being handled.

2.5 Trunk maintenance

The trunk test circuit provides the hardware features required to perform various kinds of program-controlled trunk maintenance.

2.5.1 Trunk test circuit

The trunk test circuit provides access to the tip and ring of each trunk circuit. Figure 6 shows an incoming trunk and its associated MB and TT relays. The operation of the MB relay splits the tip and ring from the trunk and connects a make-busy condition toward the local office, preventing origination of new calls on the trunk. Operation of the TT relay connects the signaling and termination circuit to the tip and ring of the trunk. This circuit is multipled to all 415 trunks through their corresponding TT relays. A 1-out-of-N check circuit controls two ferrods which indicate to the program that the MB and TT relays of one and only one trunk have operated. Once the signaling and termination circuit is connected to a trunk, marginal tests can be applied to that trunk. Off-hooks, on-hooks, various battery conditions, and 1000-Hz tone can be applied to the trunk. A ferrod (SIG

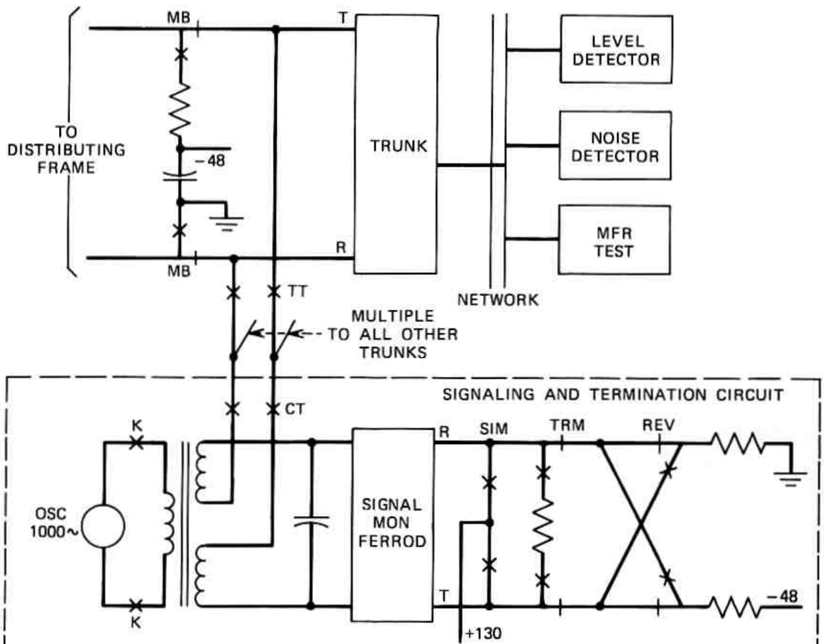


Fig. 6—Trunk testing.

MON) allows the program to monitor dc currents flowing in the tip and ring of the trunk under test.

Three test trunks, the noise detector, level detector, and MFR test circuit, have separate network appearances and are used in testing the trunks and MF receivers. The level detector can be connected through the network to the trunk under test and indicates the presence of tone via a ferrod. The level detector and tone supply are set to detect a trunk which develops a loss of more than 3 dB. Likewise, the noise detector is used to detect noise in the trunk or the network. The MFR test circuit is connected through the network to an MFR under test and can outpulse both normal and illegal multifrequency codes and test the MFR under marginal operating conditions.

2.5.2 Trunk maintenance programs

Trunk diagnostics can be automatically initiated in response to detected failures or they can be manually requested by means of appropriate input messages from a maintenance teletypewriter. When no other maintenance has been requested, routine trunk exercises are run. This background trunk exercise, which tests every trunk every few hours, also provides a powerful detection mechanism for network troubles. The sequence of trunks which is exercised is determined by a pattern based on the network appearance (GVH). This pattern was chosen to test many different portions of the network in a short period of time. The trunk maintenance programs also permit maintennance personnel to remove and restore trunks and service circuits and to run selected portions of diagnostics called blocks and segments.

2.5.2.1 Maintenance control hierarchy. The trunk maintenance programs run under a trunk and service circuit maintenance control program. The trunk control program is a client of the switching equipment maintenance monitor which runs under the system maintenance monitor (see Fig. 7) described elsewhere in this issue.[6] The switching equipment maintenance monitor controls the scheduling of maintenance activities for the master scanner, network, and trunk and service circuits; buffering requests for these programs; and executing them in order of priority.

2.5.2.2. Trunk diagnostic table. Different trunk diagnostics must be provided for each trunk type (ANI, ONI, operator, etc.). A table-driven approach was utilized to facilitate the implementation of these different diagnostics. There is a different diagnostic table for each type of trunk or service circuit. As shown in Fig. 8, the trunk network

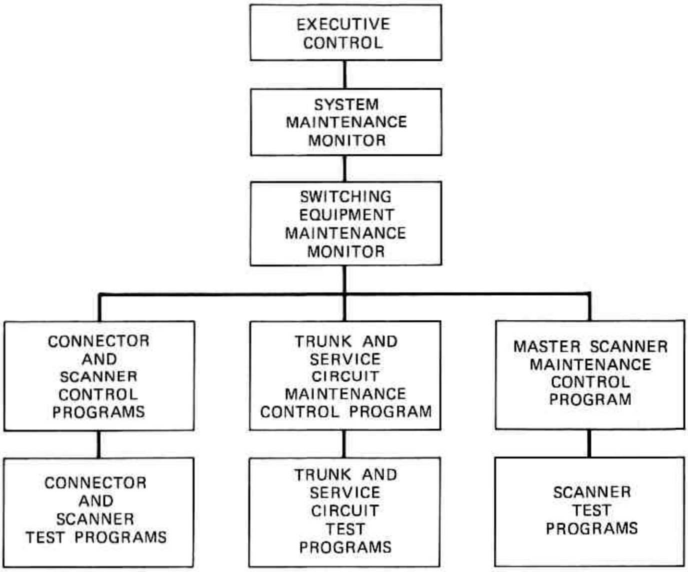


Fig. 7—Maintenance control structure.

appearance (GVH) is used to index a table of office parameters to determine the type of trunk. The type number is then used to obtain the address of the diagnostic table pertaining to that trunk type.

Each table consists of a list of tests or functions to be performed. The trunk control program executes each function in the order in which it appears in the table. Each function consists of a small program, similar to a subroutine, which performs a single task such as connecting a trunk to the level detector. Figure 9 shows a diagnostic table and the linkages between it and the function programs and subtables described in a following section. As the tasks performed by most functions are generic, they may be used as building blocks in many different diagnostics. Functions which check results make use of a data word which is also placed in the table as a separate entry immediately following the function entry. This technique permits a function to be used in many different test situations.

Diagnostic failures are identified by a block and segment number. A typical diagnostic consists of about nine blocks of tests arranged in numerically increasing order, each block containing several segments. Each function is assigned a unique segment number regardless of the order in which it appears. The segment number identifies the specific function that failed within the block.

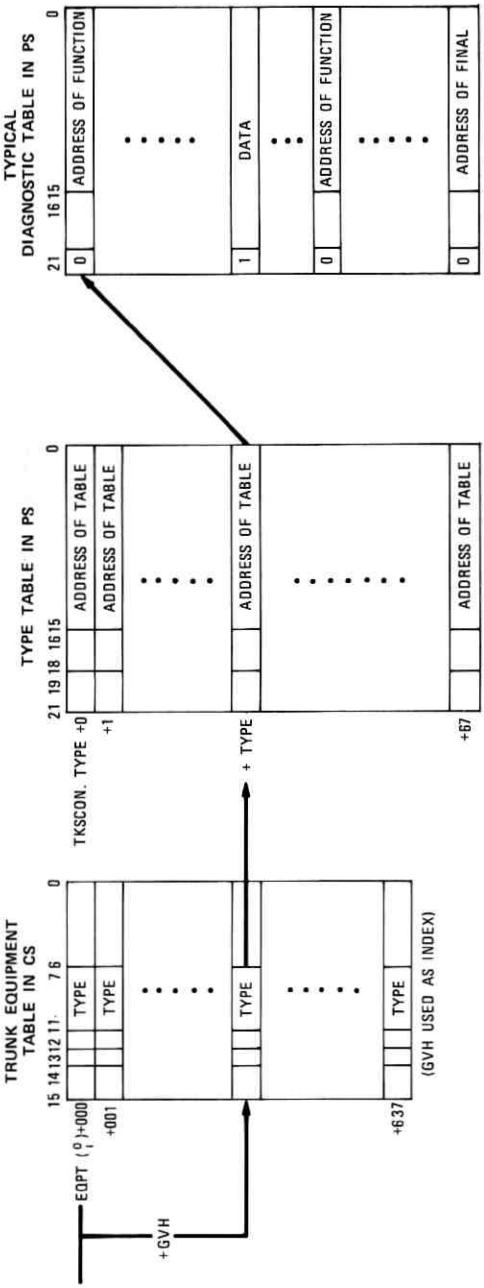


Fig. 8—Trunk diagnostic table selection.

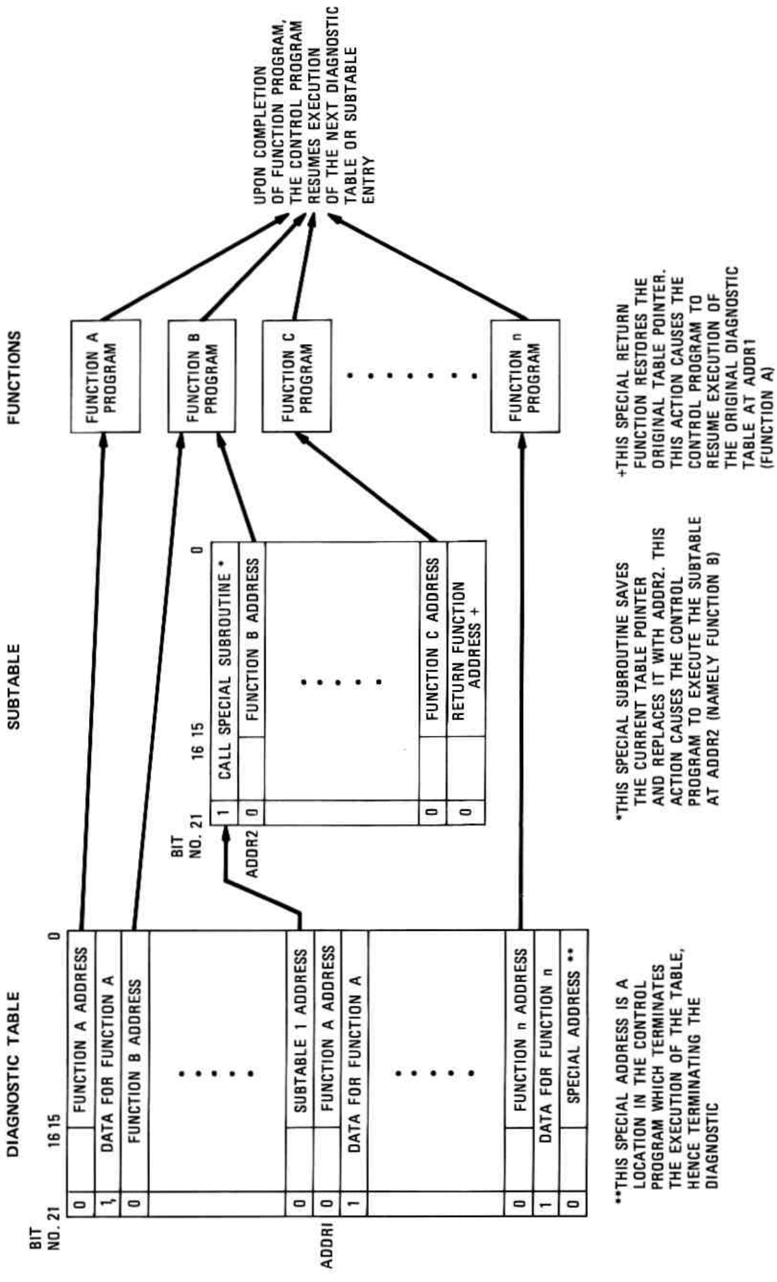


Fig. 9—Trunk diagnostic table execution.

The diagnostics are ordered so that small portions of the circuit are checked and verified and may be assumed good for subsequent tests which check features of increasing complexity. For example, some checks are made on the trunk test frame before attempting any tests. Continuity of tip and ring are verified before attempting transmission checks. Transmission is checked before attempting scans of supervision as the transmission check verifies the translators of the trunks, and scans will not work unless the translators are good. The ordering of tests allows fairly definite conclusions to be drawn as to what is causing a particular fault.

2.5.2.2.1 Subtables. A subtable feature is incorporated in the trunk and service circuit maintenance control program to permit the use of a common sequence of functions in several different diagnostic tables. Common test sequences are useful when the nature of the test is independent of the trunk type; for example, the transmission check. A subtable consists of an ordered list of functions similar to the diagnostic tables previously described. To cause a subtable to be executed, a function is used which changes pointers in a call store area, directing the control program to the subtable (see Fig. 9). A similar technique is used to return from the subtable to the diagnostic table.

2.5.2.3 Permanent signals. If a trunk remains off-hook for more than two minutes after the operator has released, a permanent-signal condition is suspected. The trunk is diagnosed to verify that the trouble is not within the trunk circuit at the AIC. If this diagnostic passes and the trunk is still off-hook, the trunk is placed in the permanent-signal state. When permanent-signal trunks go on-hook they are automatically returned to service.

2.5.2.4 Pumping trunks. Since the ANI trunk previously described uses reverse battery supervision, certain types of faults can occur which require a special detection mechanism. For example, a grounded ring will cause the trunk to go off-hook, but when it is placed in a time slot and the battery reverses it goes back on-hook. Thus a pumping condition develops in which a trunk generates rapid seizures of short duration, tying up equipment and creating unproductive work for the system. To detect this situation, a special subroutine has been included. Whenever a trunk goes on-hook shortly after being placed in a time slot, call processing calls this subroutine. The subroutine keeps a record in call store of the previous trunk it was called for and, if this matches the present trunk, it is considered to be pumping. This call store record is erased every six seconds and so for a trunk to be considered pumping, it must go on-hook shortly after being placed in the time slot twice within a six-second interval. In addition to outputting a special "PUMP" message, these trunks are removed from the time slot and placed in the suspected permanent-signal state, as described above, from which they will be diagnosed and eventually placed in the permanent-signal state.

2.5.2.5 Trunk out-of-service states. When an automatically requested diagnostic fails on a trunk, an attempt is made to remove it from service. For intercept trunks a threshold constant is checked. If the total number of automatically made-busy trunks equals that threshold, usually set to one less than the smallest trunk group, then no more intercept trunks are automatically made busy. When a trunk is removed from service, the MB relay is operated, sending a make-busy condition on the tip and ring to the local office.

By means of an appropriate TTY input message, any trunk, service circuit, or operator position can be removed from service or restored to service.

2.5.2.5.1 Trunk out-of-service displays. The status of AIS intercept trunks is displayed on lamps located on the supplementary maintenance frame key and lamp panel as shown in Fig. 10. This panel contains status lamps pertaining to the networks, files, and trunks. Red lamps indicate power-off conditions on the networks and files. White lamps indicate active connector, scanner, announcement machine, clock, and file. The row of lamps indicating trunk status provides indications of any off-normal conditions as well as any excessive trouble conditions. These lamps indicate if any (OS) or an excessive number (XOS) of incoming intercept trunks are out-of-service either manually or automatically, if any (PER) or an excessive number (XPER) of trunks are permanent signals, and if any (AMB) or an excessive number (XAMB) of trunks have automatically been made busy. In addition, operation of the MB relay of a trunk causes a lamp on the faceplate of the trunk circuit pack to light, facilitating location of the circuit pack on the frame.

III. ANNOUNCEMENT MACHINES

Natural-sounding voice messages which inform the customer of the status of an intercepted number are formed by piecing together prerecorded words and phrases. These words and phrases are recorded

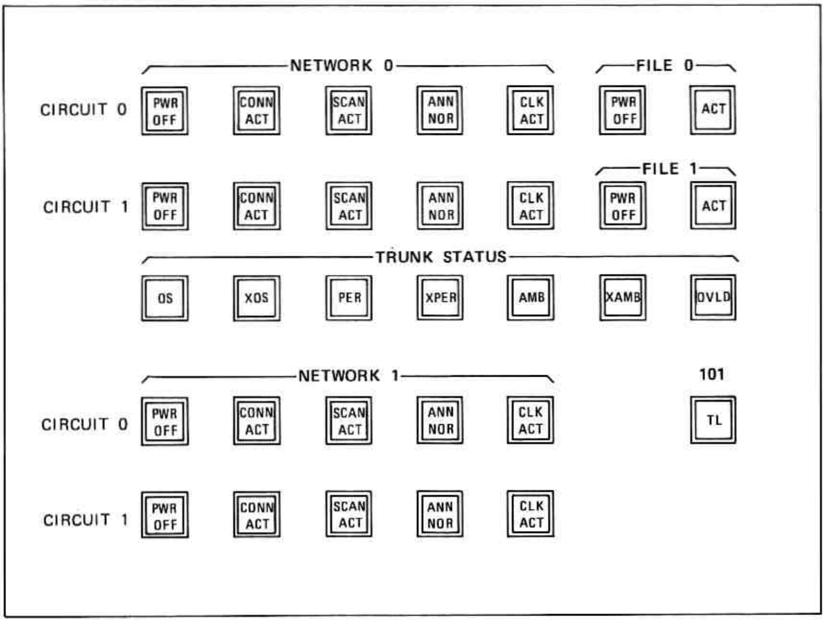


Fig. 10—Supplementary maintenance display.

on the rotating drum of a 96-track announcement machine. Two announcement machines are provided for reliability.

3.1 Announcement trunks

The output amplifier of each track is connected directly to an announcement trunk on the network. For a two-network AIS the same output is multipled to an announcement trunk on the other network, as shown in Fig. 11. Since many incoming trunks, using different time slots, can be connected to one announcement trunk, the time division switch in the announcement trunk is interrogated at a rate which varies, whereas an incoming trunk which is used on a single connection in one time slot is interrogated at a fixed rate. Therefore the resonant transfer principle cannot be used in the announcement trunks; instead, the audio signal from the announcement machine is sampled. This trunk is designed to insure high echo and singing return losses, low crosstalk, and high talk-through suppression.

3.2 Message synthesis

The announcement is assembled by establishing a sequence of connections through the connector and scanner between the incoming

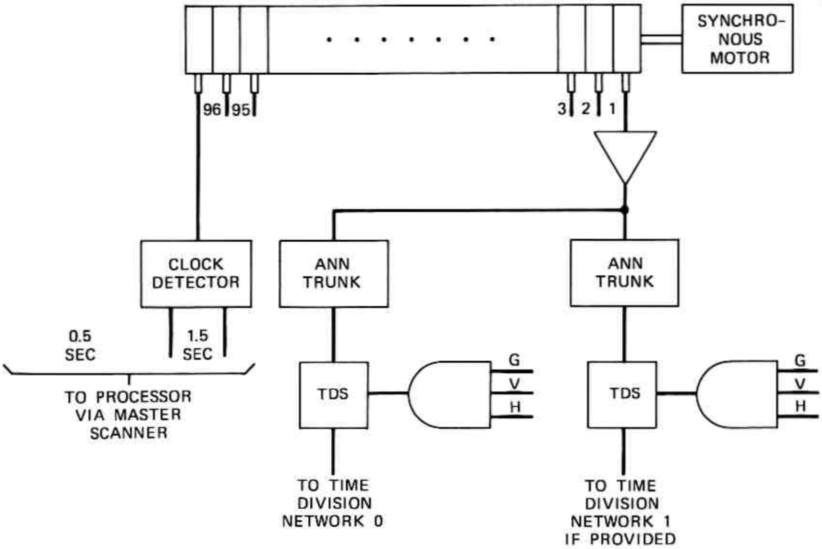


Fig. 11—Announcement machine.

trunk and the appropriate announcement trunk. By using the time division network in such a role, expensive hardware message synthesizers are not required, thereby reducing the cost and simplifying the maintenance of the announcement machine.

3.2.1 Vocabulary

The 96 tracks consist of either 0.5-second or 1.5-second phrases and provide a vocabulary of about 170 words. The 1.5-second tracks are used for phrases such as "The Number You Have Reached" or "Calls Are Being Taken By." The 0.5-second tracks are used for the digits 0–9, letters, and words such as "Hundred" or "Thousand." Two tracks are provided for each digit, one recorded with a neutral inflection, one with a falling inflection. The falling inflection is used at the end of a string of digits preceding a pause to enhance the naturalness of the message.

Tracks 1 to 48 are standard and are the same for all installations (see Table II). The recordings on tracks 49–96 are specified by the telephone company when a system is ordered. These tracks may be used to record locality information or to record letters if all-number calling is not being used.

Table II — Announcement machine tracks

Track No.	Phrase or Digit
1	The Number You Have Reached
2	Has Been Changed
3	The New Number Is
4	To a Nonpublished Number
5	For Incoming Calls
6	In Area Code
7	Has Been Disconnected
8	To a Nonlisted Number
9	Temporarily
10	At The Customer's Request
11	Is Being Changed
12	The New Number
13	Is Not Yet Connected
14	(Pause)
15	Calls Are Being Taken By
16	Is Not In Service
17	Is a Working Number
18	Please Check The Number
19	And Dial Again
20	If You Need Assistance
21	Please Make a Note of It
22	You May Stay on the Line
23	And An Operator Will Answer
24	Thousand
25	Hundred
26 to 35	Digit 0 to 9 With Neutral Inflection
36 to 45	Digit 0 to 9 With Falling Inflection
46	(Reorder Tone)
47	Area Code
48	Will You Dial It Again, Please
49 to 96	Locality name as required, for example, "In Freehold"

3.2.2 Phrase timing

The drum rotation period is 4.5 seconds. There are three repetitions of a 1.5-second phrase around the periphery of the drum and nine repetitions of the 0.5-second phrase. All tracks are synchronized and a 97th track is recorded with tones which indicate the start of each phrase. These tones are converted to dc signals which are read via the master scanner; when a new phrase begins, a new network connection is made by the processor.

3.3 Maintenance

Maintenance programs monitor power and fuse alarms, the rotational speed of the drum, and the output level of each track. Power failures or blown fuses are detected via master scanner ferrods. The rotational speed is checked by a program in the timed 25-ms input-output (IO25) interrupt which times the phrase-synchronizing clock pulses. Faults in individual tracks are detected by announcement trunk exercises which are interleaved with other trunk exercises. An individual track is connected through the network to the level detector which was previously described. The test program synchronizes with the beginning of a phrase and checks the output of the detector every 25 ms for the duration of the phrase. This is necessary since the detector indicates level present only during peaks in the voice signal. To insure that a failure is reported only when a failure actually exists, the level detector is set so that a phrase containing a small amount of energy, such as the digit "eight," will not be reported as bad unless the output level falls by at least 7 dB. Upon detecting either a power, timing, or track level failure, the faulty machine is removed from service and the duplicate machine handles all announcements.

3.4 Special announcement machine

A special announcement machine has been provided to permit local recording of announcements to meet special situations. Calls may be directed to this machine by entering the appropriate status for the numbers in the disc storage file.

IV. NETWORK

The AIC interconnects trunks using time division switching circuitry similar to that developed for No. 101 ESS.[7] The AIC can have either one or two time division switching networks depending upon the amount of traffic. A network may be equipped with 415 trunks for use with incoming intercept trunks, service circuits, operator positions, etc., in addition to 96 duplicated announcement trunks. Each network provides 64 time slots and contains two connector and scanner circuits, each capable of controlling 32 connections and monitoring the supervisory state of all 415 trunks of a network.

4.1 Network topology

Each network interconnects trunks via a grid of time-shared subgroup buses and two intergroup buses as shown in Fig. 12. Full access is not required between the two networks as only the announcement machines and operator positions serve both networks. As previously described, announcement tracks are multipled to both networks and full access to operator groups is achieved through a

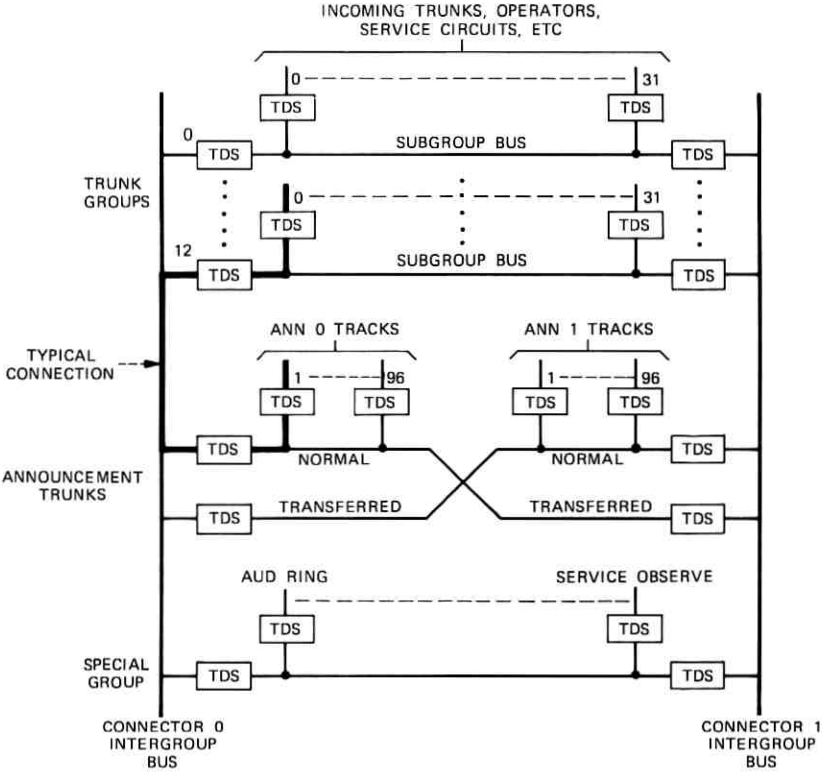


Fig. 12—Time division network buses.

transfer arrangement in the operator trunks. A talking path from one trunk to another is established in a time slot by operating appropriate time division switches. The time division switches of 32 trunks are connected to a subgroup bus. Thirteen subgroup buses are required to accommodate the 415 trunks; each bus has 2 TDS allowing it to be connected to either intergroup bus. Each network has two intergroup buses, each associated with a connector circuit which controls 32 talking time slots.

Trunks of the two announcement machines appear on separate buses. A program-controlled flip-flop within the connector selects the announcement machine to be used by a particular connector. In addition, a special group is provided so that a three-way connection can be established to include service observing or audible ringing.

4.2 Connector and scanner circuit

A network contains two connector and scanner circuits, two clocks, and one transfer and alarm circuit. The transfer and alarm circuit and one of the duplicated connector and scanner circuits are shown in Fig. 13. The transfer and alarm circuit is used (*i*) to remove a faulty connector and its clock and scanner from service; (*ii*) to change the announcement configuration; and (*iii*) to select the online clock and scanner.

A 3.25-MHz clock is used to control all data manipulating, gating, and time division switch closure signals. The two connectors are interlocked to insure that only one connector is establishing a connection at a particular instant of time. The clock which is placed on line by the transfer and alarm circuit drives both connectors. A time slot counter driven by the clock establishes a number of time intervals or time slots (TS) which are assigned to three functions as shown in Fig. 14: (*i*) connection control (TS 0–31); (*ii*) processor communication (TS 32); and (*iii*) autonomous trunk scanning (TS 33 and 34). Each time slot interval is 2.464 μs long. As shown in Fig. 14, the connector whose scanner is on line recycles in 86.24 μs; the other connector recycles in 83.78 μs because it skips one of the

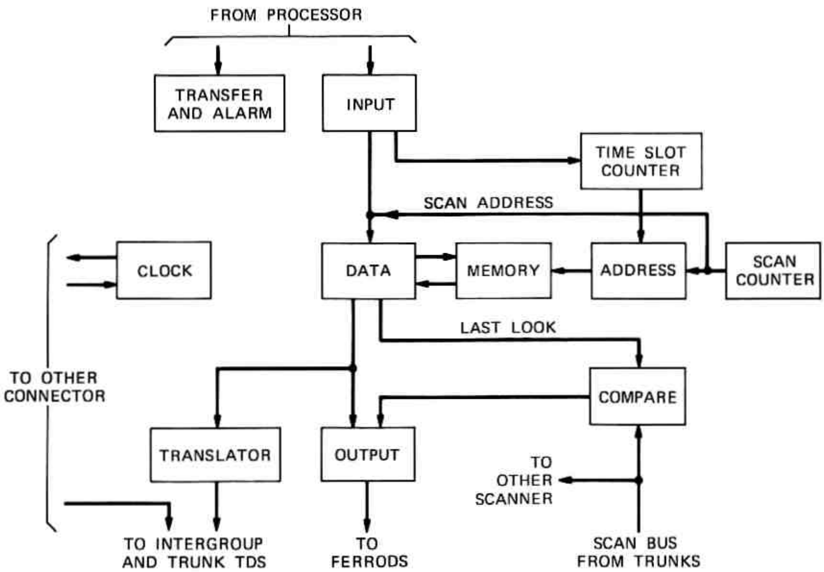


Fig. 13—Connector and scanner circuit.

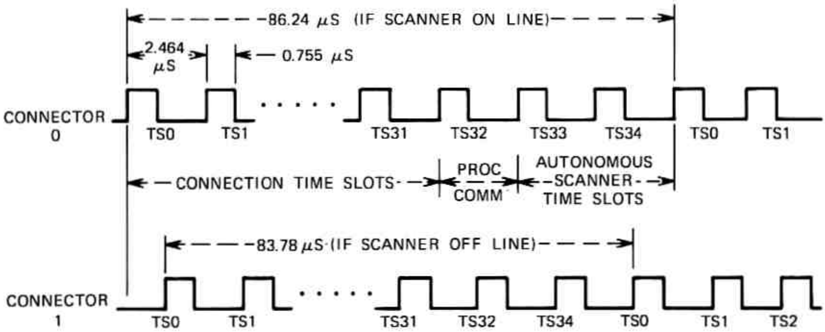


Fig. 14—Time division switch timing.

autonomous scanner time slots, TS 33. This causes the time slots of the two connectors to precess, preventing crosstalk.

4.2.1 Connection control

During TS 0–31 the connector uses data stored in the connector memory to establish talking connections. The memory contains 32 time slot words, one for each talking time slot, and 26 words used by the autonomous scanner described later. Each word is 24 bits long. The connector reads the trunk numbers which are to be connected together from one of the 32 time slot memory locations selected by the time slot counter (Fig. 13). A trunk number consists of group, vertical, and horizontal digits, each made up of three bits. Each nine-bit trunk number is translated into three 1-out-of-8 select signals (G, V, H) which operate the appropriate TDS. Figure 12 shows a connection to an announcement trunk which requires the closure of four TDS. While the connection is in progress, a test is made to verify that the proper number of TDS have closed. A connector closes the TDS for 0.755 μs of the 2.464-μs time slot interval. The remainder of the 2.464-μs interval is used for data manipulation and for TDS closures by the other connector. To prevent crosstalk from one time slot to the following time slot, the time division buses are discharged by clamping to ground after each TDS closure.

4.2.2 Processor communication

During TS 32, the connector checks the input register for requests from the AIS processor. The processor can request that a new connection be written into the memory, request a readout of a memory location, or request a directed scan of a particular trunk to determine its present state (on-hook or off-hook).

4.2.3 Autonomous scanner

The autonomous scanner of one connector monitors the supervisory states of all trunks on the network. The scanner contains a nine-bit scan counter which identifies the trunk to be scanned. During TS 33 and 34 one trunk is scanned by gating the scan counter to the data register, which causes the translator to select the appropriate trunk (Fig. 13). In response, the selected trunk sends back a status signal to the scanner over the scan bus. There is a last-look bit in the memory corresponding to each trunk. The last-look bits are stored 16 to a word; thus 26 words are needed to accommodate the 415 trunks. The five high-order bits of the nine-bit scan counter are used to read the memory at one of the locations. After comparing a trunk's present state with its corresponding last-look bit, the autonomous scanner advances to the next trunk if no change has occurred. If a change is detected, the scanner stops scanning, the last look memory is updated, and a message identifying the trunk and its present state is loaded into the output register which controls ferrods on the master scanner. Call processing programs check for autonomous scanner messages once every 25 ms. When a call processing program has read the scanner message, it signals the scanner to continue scanning. The scanner can scan one trunk during each TS 33, which occurs once every 86.24 μs. Thus, if no change is detected, the 415 trunks can be scanned in 36 ms.

The connector and scanner maintenance programs initialize the last-look and time slot memory whenever it may be in error. A scanner switch or a clock switch will be followed by a memory update which uses call processing records in call store to reinitialize the connector's memory. After the update is completed, the scanner is switched on line, and it begins scanning from the first trunk.

4.3 Connector and scanner fault detection and recovery

Connector and scanner faults are detected by call processing, audits, and trunk maintenance programs as well as connector and scanner maintenance programs. When a failure is detected, a connector and scanner working mode program establishes a working configuration and requests the appropriate diagnostic.

The equipment in the peripheral system is duplicated and connected to the processor's communication buses, which are dedicated

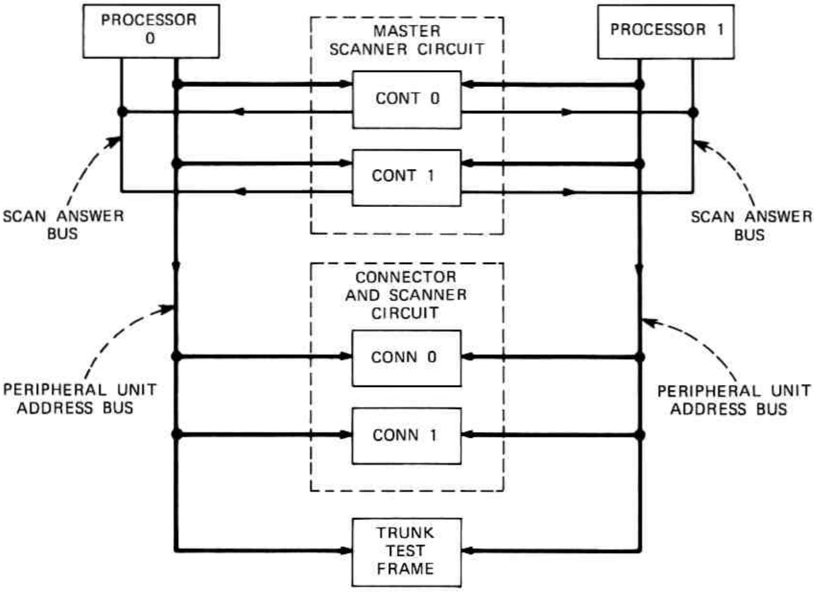


Fig. 15—Duplication oι important network circuits.

to the corresponding processors as shown in Fig. 15. Because of the dedicated busing, a peripheral unit fault recovery or working mode program must switch the processor online configuration in order to change the bus access to a peripheral unit. If the working mode program identifies a problem in the interface between a processor and a peripheral unit, the processor is removed from service and the appropriate processor-peripheral unit diagnostics are requested.

Figure 16 shows the strategy used by the connector and scanner working mode program to establish a working configuration and request the proper diagnostic. When a connector and scanner working mode program is called with a failure, the working mode program retries the failing operation and, if there is no error on the retry, it increments a transient error count. If the retry fails, the online processor is placed off line and the standby processor goes on line. With the new processor on line, the order is retired again. If it fails again, the connector is removed from service and an online connector diagnostic is requested. If the second retry passes, the offline processor is removed from service and an offline connector diagnostic is requested. Since the connector is functioning properly with the online processor, the connector remains in service.

Some operations cannot be retried, others have the property of always passing when retried if certain faults are present. To cover these cases the working mode program monitors the error rate (see Fig. 16). When the rate becomes excessive, the equipment is reconfigured and the rate is monitored again. If a connector has an excessive error rate with both processors, the working mode program removes the connector from service and requests an online diagnostic. If a connector has an excessive error rate with a single processor, that processor is removed from service and an offline diagnostic is requested for the connector.

4.3.1 Fault detection—call processing

Call processing programs can detect errors on most operations they request the connector and scanner to perform. The reaction to a failure of any of these requests is similar to the two described below.

Call processing programs check the connector and scanner each time they establish a connection or scan a trunk. When data are sent to the connector, the circuit sends an enable verify signal back to the processor to indicate that it has received the data. If the circuit fails to send back this signal, the program calls in a connector and scanner working mode program.

When the autonomous scanner sends a message to the call processing program, the program compares the trunk status change with its records. An unexpected change of state (e.g., an off-hook from an unequipped trunk) will cause the call processing program to call a working mode program which will increment an autonomous scanner error counter. The autonomous scanner operations cannot be retried, and so the working mode program monitors the rate at which errors are detected. When the autonomous scanner causes ten errors within a ten-minute interval, the working mode program removes the circuit from service and requests an online diagnostic.

4.3.2 Fault detection—audits

An audit program is able to detect connector and scanner errors as it checks the data associated with the intercept connections.[6] The program can detect when data fail to be transmitted to the connector memory properly. It also can detect memory addressing errors, which can cause data to be overwritten by a subsequent connection.

If a time slot readout error is detected, the time slot readout working mode program retries the failing operation by retransmitting

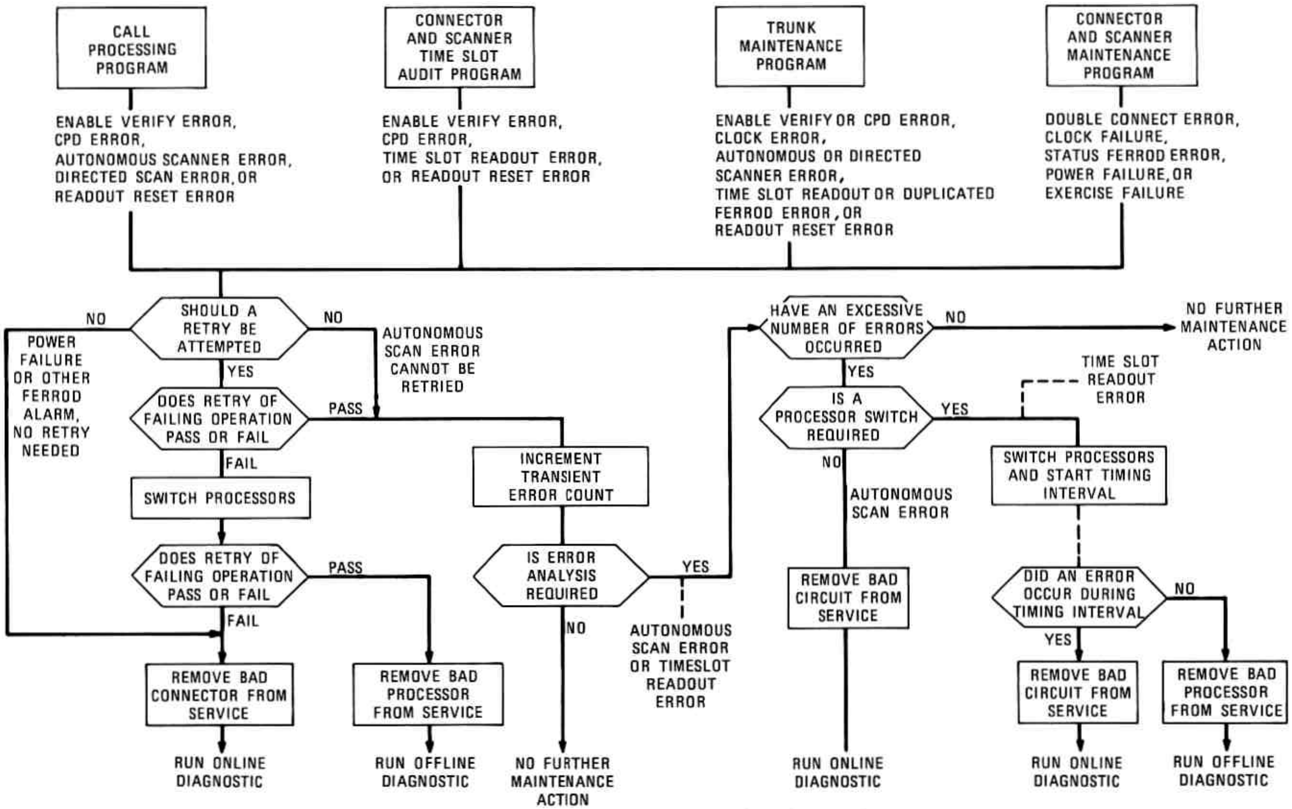


Fig. 16—Connector and scanner fault detection and recovery.

the connection data and then rereads the time slot memory. If the retry passes, a time slot readout transient error counter is incremented. If the failure persists, the processors are switched and the retry is repeated again and a connector and scanner diagnostic is requested.

If memory addressing faults are causing data to be overwritten, retries of a specific failure will always pass. Therefore, the time slot readout working mode program monitors the transient error rate. If the program detects four transient errors within ten minutes, it will cause a processor switch. If no further errors occur for over two minutes, the offline processor is removed from service and the connector offline diagnostic is requested. If errors continue to occur, the connector is removed from service and an online diagnostic is requested.

4.3.3 Fault detection—trunk maintenance

The trunk maintenance programs systematically test the ability of the network to scan trunks and to transmit audio signals. These programs force the trunk under test off-hook and on-hook by applying and removing seizures using the trunk test circuit. The trunk state changes reported by the autonomous scanner are verified by a directed scan of the trunk. If a trunk state change occurs but it is not detected by the autonomous scanner within four seconds, the trunk maintenance program calls the autonomous scanner working mode program. On the other hand, if both the directed scan and the autonomous scan results are incorrect, the failure is attributed to the trunk being tested.

Failure of TDS clock signals can cause excessive noise or transmission failures. Therefore, if the trunk and service circuit diagnostic detects these failures it will switch clocks and repeat the failing test. If the test continues to fail with the new clock on line, the failure is attributed to the trunk. If the test fails with only one clock, that clock is marked out-of-service. Because of the close relationship between trunk and network operations, the trunk and service circuit maintenance programs are an important mechanism for detecting problems in the interface between the connector and scanner and the trunk and service circuits.

4.3.4 Fault detection—connector and scanner maintenance

Alarm ferrods report closure of an improper number of TDS, voltage alarms, and the configuration of the clock, scanner, and connector. Connector maintenance programs monitor these ferrods and remove and diagnose the appropriate circuit if a trouble is indicated. Once a day the entire diagnostic is run in an exercise mode on each connector to insure that all maintenance and redundant circuitry is functioning properly.

4.4 Connector and scanner exercise and diagnostic programs

There are three test control programs: exercise, online diagnostic, and offline diagnostic (see Fig. 17). There are nine connector and scanner maintenance test blocks which are used by the control programs to test the connector and scanner. These tests can be requested automatically by a working mode program or manually via the teletypewriter. The test blocks are run in an order such that a circuit used to test another circuit has itself been previously tested. For example, the time slot and autonomous scanner last-look memory is checked before testing the autonomous scanner. Therefore, when the autonomous scanner last test is run, the circuitry under suspicion does not include the last-look memory. When a test fails, the diagnostic program will generate a TTY printout containing a trouble number. A trouble locating manual cross references trouble numbers to circuit packs to be replaced.

The general features of the three control programs include removing the connector from service, sequencing through the tests, forming the appropriate printout, and, if possible, restoring the connector to service. The exercise control program tests a network if both connectors are in service. After testing both connectors, a network is initialized with a different clock and scanner configuration. After an automatically requested connector online diagnostic, the control program will restore the connector to service if all tests passed. The offline diagnostic control will restore a connector to its original status when the diagnostic is complete. That is, whether

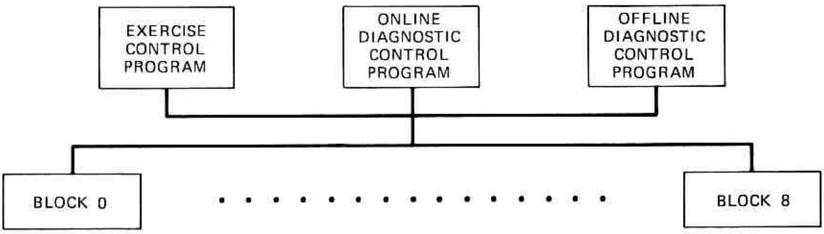


Fig. 17—Connector and scanner exercise and diagnostic.

the diagnostic fails or passes, a connector which was on line will be restored to service, since it operates satisfactorily with the online processor. A manual request can instruct a test control program to run a complete exercise or diagnostic or a small part of a diagnostic. For example, a single test block or a test segment may be run separately.

4.4.1 Connector and scanner removal from service

When a circuit is going to be tested, the control programs and the working mode programs can cause the connector to be removed from service "gracefully" or "hard." A "graceful" removal allows the calls in progress to complete before starting the test. On the other hand, the "hard" removal causes any calls which were in progress to be immediately removed from the time slots and routed to operators via the other connector. The online diagnostic will only "hard" remove when the fault has affected the calls which were in progress (for example, a fault in the connector memory). The exercise and offline diagnostic control programs always "gracefully" remove a connector from service.

4.4.2 Offline diagnostic

If the working mode program encounters a processor-connector interface problem, the processor is removed from service and an offline diagnostic is requested. The connector offline diagnostic is preceded by a diagnostic on the offline processor. If the processor diagnostic passes, the connector offline diagnostic program tests the interface between a processor and a connector. Figure 18 shows the functions performed by the two processors during an offline diagnostic. The control program in the online processor must wait for an interval during which no peripheral orders will be issued from the online processor. The autonomous IO circuit in the online processor sends orders to the peripheral system every 1.25 ms. Also the connector and scanner offline diagnostic control program running at base level in the online processor may be interrupted by IO25 or file interrupts. Programs in these interrupt levels in the online processor may also send orders to the peripheral system. When an interval is found in which there will be no IO circuit activity and no IO25 or file interrupts, the offline processor is started and may send peripheral orders without interfering with the work of the online processor. The tests run in the offline processor are segmented into short 640-μs sections. After completing a section, the connector

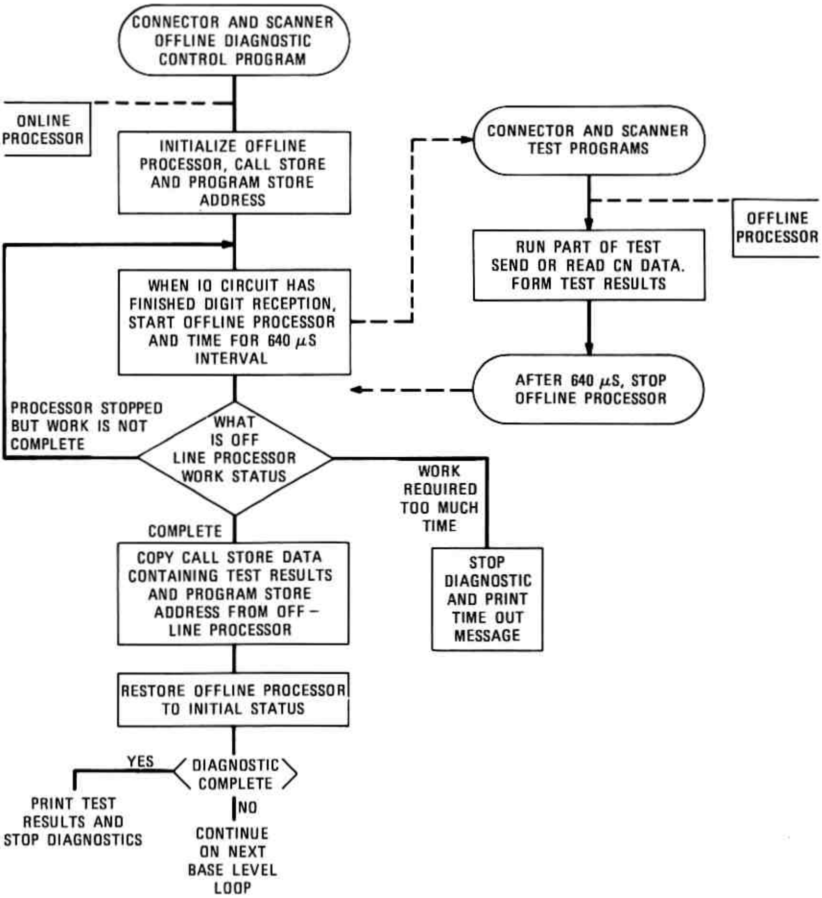


Fig. 18—Connector and scanner offline diagnostic.

and scanner diagnostic program running in the offline processor stops the offline processor.

While the offline processor is executing a test, the control program in the online processor times for 640 μs. At the end of this interval it checks the status of the offline processor. If the offline processor has stopped but has more work to do on this base level loop, the control program will locate another noninterfering time interval and allow the offline processor to execute another section.

When the offline processor has completed the work which is to be done on this base level loop, it signals the control program in the online processor to return to the maintenance monitor. The offline diagnostic control program then copies data from the offline call store and restores the offline processor to its initial status. If the diagnostic is not finished, it will be continued on the next base level loop. If the diagnostic is complete, appropriate TTY messages are printed.

V. CENTRALIZED INTERCEPT BUREAU (CIB)

Most intercept traffic is handled automatically by the AIC without need of human intervention. However, a small number of calls—less than 5 percent—require the assistance of an operator. When a call is routed to an operator, all available data about the call are sent to the position via a data outpulser and displayed in front of her. In some cases, such as the dissolution of partnership where there is more than one new number, the information is not placed on the disc file and the operator may have to refer to a small position record. The operator console and a typical display are shown in Fig. 19.

5.1 CIB position

The operator has the ability to interrogate the disc file to obtain information about a number other than the one displayed. To convey her request to the processor, the operator seizes an inquiry trunk to the appropriate AIC and MF keypulses the necessary digits to an MFR. After keying, her old display is erased and replaced by the new display. She also can elect to hear an announcement regarding the keyed number but this is rarely done because of excessive operator holding time.

Should it become necessary, she may transfer a customer to the service assistant or she may include the service assistant for a conference. The operator is also able to extend the call via a subscriber line circuit should the occasion arise. Tie lines are also provided at the position so the operator may have access to the service order bureau or to the repair service desk.

The position has been designed for ease of installation. Since very few leads connect from one position to another, adding, removing, or changing positions in the field is a very simple operation. The position is designed with plug-in connectors and circuit packs and requires a 24 V power source. Since each position is self-contained, the maintenance program is considerably simplified over that which would be required if positions shared common equipment.

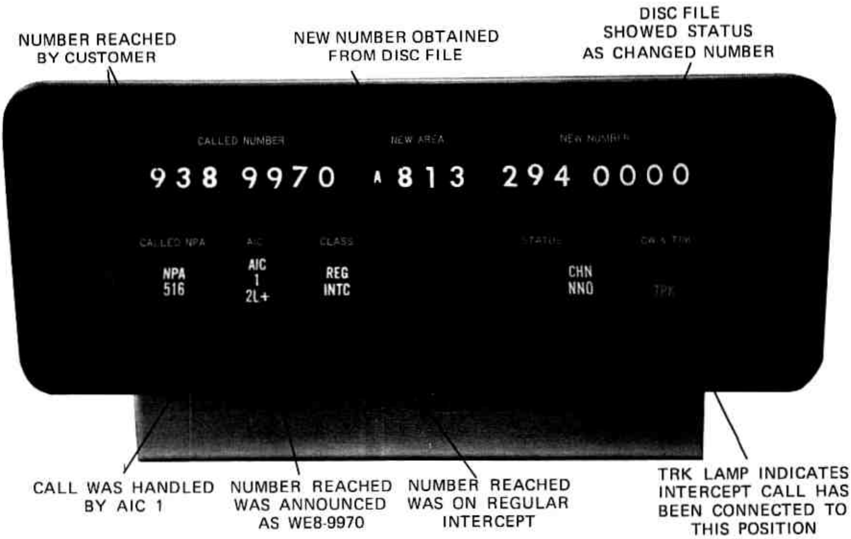


Fig. 19—CIB position and typical display.

5.2 ONI position

Although the CIB may also be used to handle ONI traffic for the home AIC, for those AIC which have a large amount of ONI traffic a special-purpose operator position is used. Since the ONI position is only used for the operator number identification function, it is less expensive than a CIB position which has data receiving and display circuitry. On an ONI call the operator interrogates the customer to obtain the called number and keys this number into the AIC. Since MFR usage is so great in an ONI position, the MFRs are dedicated to each position and not switched through the network. A special MFR test trunk may be automatically connected to any dedicated MFR to provide a network appearance to allow program tests.

The ONI positions are administered as a separate operator group. An ONI operator group is required at a remote AIC if that AIC handles ONI traffic. The home AIC has a night transfer arrangement which permits closing the ONI group of the home AIC and allowing the CIB group to handle all traffic.

5.3 Operator training

A maximum of two CIB positions may be used for operator training. During periods of training, the processor will not allow service traffic to be directed to a training position. A standard 12-button *Touch-Tone* card dialer plugs into the position, or may be remotely located, and terminates in a dedicated incoming trunk at the AIC. The trainer uses the card dialer to outpulse intercepted numbers. This training call will then be connected to the training position with a display. All types of incoming traffic can be simulated in this manner.

5.4 Administrative and maintenance features

Various features are provided to assist traffic and plant personnel with the operation of the system.

5.4.1 Operator group displays

A lamp signal cabinet is furnished at the CIB and ONI position locations. The lamps in this cabinet indicate the following:

(*i*) Position occupied.
(*ii*) Position busy or idle.
(*iii*) Calls are waiting to be served.

(*iv*) An excessive number of calls are waiting to be served.
(*v*) Service affecting trouble exists at the AIC.
(*vi*) Position on training.
(*vii*) Night transfer of ONI to CIB in effect.

5.4.2 Operator-keyed troubles

Instead of filling out paper tickets, operators report troubles by keypulsing a three-digit trouble code into the AIC. When an operator-keyed trouble report is received, the maintenance TTY will print the position number reporting the trouble, the incoming trunk number, and the time slot associated with the call. These trouble reports have been used to locate troubles not only in the AIC and CIB but also in the local offices, concentrators, and other switch trains connecting to the AIC.

5.4.3 Test codes

Craftsmen may request test setups for position trunks by MF keypulsing three-digit test codes from the position. A series of codes are reserved to request various test displays from the AIC. Other codes request connections to transmission test lines such as quiet termination, 1000-Hz milliwatt generator, or a jack-ended test line.

Craftsmen in local offices may also reach these test lines in the AIC by keying similar test codes over incoming intercept trunks. In the case of trunks which are immediately connected to operators (directory assistance, ONI), the craftsman at the local office may seize the trunk and then request the operator to key the test code to transfer his trunk to the test line.

5.4.4 Service observing

Service observing (SO) is provided on intercept calls connected to operator positions. A high-impedance trunk appears on the time division network and is bridged onto the connection when an observation is being made. This trunk is connected to a service observing desk which is also shared with other systems. The AIC transmits information pertaining to the call to the service observing circuit which forwards it to the desk where it is displayed in front of the service observer.

VI. CONCLUSION

The AIS peripheral circuits, consisting of the time division network, announcement machine, and trunk and service circuits, have been described. Maintenance techniques consisting of both hardware and software features have been integrated to achieve a reliable system.

REFERENCES

1. C. J. Byrne, W. A. Winckelmann, and R. M. Wolfe, "Automatic Intercept System: Organization and Objectives," B.S.T.J., this issue, pp. 1–18.
2. J. Digrindakis, L. Freimanis, H. R. Hofmann, and R. G. Taylor, "[No. 2 ESS] Peripheral System," B.S.T.J., *48*, No. 8 (October 1969), pp. 2669–2712.
3. H. J. Beuscher, G. E. Fessler, D. W. Huffman, P. J. Kennedy, and E. Nussbaum, "[No. 2 ESS] Administration and Maintenance Plan," B.S.T.J., *48*, No. 8 October 1969), pp. 2765–2816.
4. H. B. Haard and C. G. Svala, "Means for Detecting and/or Generating Pulses," U. S. Pat. 2718621, issued 1955.
5. W. L. Shafer, Jr., "The No. 23 Auxiliary Operating Room Desk," Bell Laboratories Record, *31*, May 1953, pp. 167–171.
6. H. Cohen, D. E. Confalone, B. D. Wagner, and W. W. Wood, "Automatic Intercept System: Operational Programs," B.S.T.J., this issue, pp. 19–69.
7. T. E. Browne, D. J. Wadsworth, and R. K. York, "New Time Division Switch Units for No. 101 ESS," B.S.T.J., *48*, No. 2 (February 1969), pp. 443–476.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 53, No. 1, January 1974
Printed in U.S.A.

Automatic Intercept System:

File Subsystem

By J. W. HOPKINS, P. D. HUNTER, R. E. MACHOL,
J. J. DiSALVO, and R. J. PIERETH

(Manuscript received June 8, 1973)

The Automatic Intercept System must maintain a large data base from which information can be obtained pertaining to telephone numbers on intercept in the area. The file complex provides the mass storage medium necessary for this system. This article describes this subsystem, its operation, and how it is maintained.

I. INTRODUCTION

The Automatic Intercept System (AIS) must maintain a large data base from which information can be obtained pertaining to the hundreds of thousands of telephone numbers on intercept in an area. The file complex, consisting of a disc file and associated controller, provides the mass storage medium and data access mechanism for the AIS.

In establishing such a data base, two important points were considered. First, the file complex must be highly reliable. Second, the frequency of calls and the changeable nature of the data require that the data base be readily accessible and easily alterable.

In selecting a disc file it was determined that the disc file used by No. 1 ESS ADF would adequately fulfill AIS requirements. In addition, it would be possible to make use of much of the head accessing and motor drive circuitry already developed for No. 1 ESS ADF disc.[1]

High reliability is provided both through a coordinated hardware-software design which includes many hardware and software error detection and location techniques, and by duplicating the entire file complex—each file containing identical data.

Data interchange between the system control units and the two files is accomplished independently over the peripheral bus system under control of signals from the central pulse distributor (CPD).

Since the file subsystem is comprised of two identical units, much of the following hardware and software description considers just a single unit.

II. HARDWARE DESCRIPTION—FILE CONTROL

2.1 General

The file controller is a wired logic machine consisting of several synchronous sequencers and register circuits. The sequencers are individually associated with major file functions. Collectively, they provide the ability to retrieve, store, and check the data on the discs.

The file circuitry is composed of discrete-component, diode-transistor-logic (DTL) circuit packs, originally designed for the No. 1 ESS.[2] The choice of this type of logic provided a wide variety of available circuits, compatible disc-logic speed, and a favorable cost factor.

In the AIS, the file subsystem has duplicated files and each file operates independently of the other, allowing each to handle separate operations simultaneously. The data base on each disc, however, is the same with the exception of the recorded "call-counts" (the number of times the entry has been referenced) on the individual entries. This configuration provides both reliability and higher call throughput. If one file is out-of-service, the remaining file has the ability to handle all call processing requirements with minimum effect on service.

Although a file complex is a synchronous machine (using clocks recorded on the disc), it is independent of the No. 2 ESS Control Unit (CU) and performs most of its actions autonomously. It requires only the initial data and an instruction to perform a particular function such as looking up the status of an intercepted number. This enables the CU to perform other tasks while lookups and other functions are performed by the files.

2.2 Control Unit—file communications

A Control Unit communicates with a file over a 36-bit, ac, Peripheral Unit Address Bus (PUAB) and a 16-bit Scan Answer Bus (SCAB), as shown in Fig. 1. Each CU has its own PUAB and SCAB but each file connects to both sets of busses, allowing either CU to communicate with either file.

A file receives orders and data from the online CU via its PUAB and returns data to both CUs over the SCABs. Instructions either to

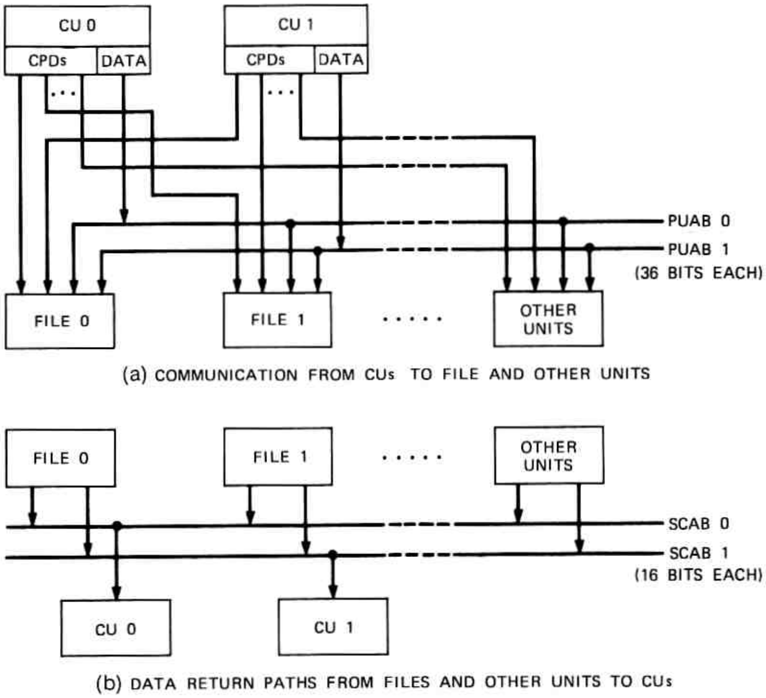


Fig. 1—CU communications paths.

accept data from a PUAB or to gate data onto the SCABs are received over separate leads termed CPD leads. Unlike the PUAB and SCAB busses which multiple to all peripheral units, separate CPD leads are assigned to each file.

2.3 Timing circuit

Timing in the file complex is provided by the combined use of three clock tracks recorded on the discs (see Section 3.3). One track is written as a single pulse, called track index, which is used to define the start of each revolution. The other two clock tracks are written as a pair. The first clock track of this pair defines the beginning of each bit. It is called the bit clock. The second clock track of the pair defines which bit within a word is present at any particular instant. It contains a pseudo-random pattern 46 bits in length which is repeated 1600 times around the track (a track contains sixteen hundred 46-bit words). This pattern is unique in every set of six consecutive bit positions and is, therefore, decodable into a 46-phase clock. The pattern is read from the disc into an open-ended shift register with the clock decoding circuit using the contents of the register to produce the proper bit time indication. Use of seven stages of register instead of six allows instant detection of an error in the reading of the pattern from the disc by providing a simple error detecting circuit. The bit clock provides a shifting signal for the shift register and is used for strobing data read from the disc.

This pair of clocks provides positional information within each word, while the word counter (Section 2.4.1) provides positional information with respect to the beginning of a disc revolution.

2.4 Register circuits

A number of special-purpose registers are provided to perform various file operations. A brief description of some of these is given in order to suggest the types of information needed to handle data on the file.

2.4.1 Word count register

The word count register has two separate sections, a counter and a register. The counter, which counts to 1600, is set to 0 at the beginning of each revolution of the disc and is incremented by one phase of the 46-phase clock. The associated 12-bit word count register can be loaded in parallel from either the word counter or directly by the CU. The contents of the counter and the register can be compared in a match circuit which will indicate when the counts are the same.

During an associative lookup (see Section V), the contents of the word counter section is gated into the register section to store the actual location of the called number word that was found. On the subsequent disc revolution the match circuit locates this word for call count rewrite. For block read or write operations, the CU loads the word count register with a location in the block ahead of the desired block to be processed. This enables the file to interrupt the CU when the disc is in the proper position, thus allowing the CU to be freed for other processing during the waiting period.

2.4.2 Called number register

The called number register is a parallel input, serial output 30-bit flip-flop register. During a lookup sequence it is loaded by the CU over the PUAB with the intercepted number to be looked up. During a search sequence this register is read out serially and matched against data read from the disc file. The serial output of this register is also used in rewriting a called number on the disc during call count rewriting.

2.4.3 Input-output register

The input-output (IO) register acts as a buffer between the CU and the file. When the IO register is used in lookup functions, the status (reason for intercept) of the intercepted number and new number, if one exists, are loaded into it serially from the disc. In block read and write operations, the IO register is used to buffer the word being transferred to or from the disc.

2.4.4 File status register

The 16-bit file status register is used to store certain indications for use by the CU, such as progress of file operations, file availability, and errors that might be encountered during a file operation.

2.5 Correspondence control circuit

This circuit consists of a group of subcircuits which perform the following functions:

(*i*) Control the associative lookup sequence.
(*ii*) Check for a serial match between data in the called number register and data read from the disc file.
(*iii*) Control block read and write functions.
(*iv*) Perform miscellaneous data checking and gating functions.

The combined use of these circuits is described in Section V.

2.6 Maintenance

The major portion of the maintenance circuit consists of a set of "crosspoints" (see Section VII) that can be accessed by the CU to determine the states of circuits internal to the file. The remaining portions of the maintenance subcircuits perform the following functions:

(*i*) Detect when the voltage output of any dc-dc converter in the file complex is incorrect.
(*ii*) Check the integrity of the timing circuits.
(*iii*) Check for proper cycling of the word counter.

III. HARDWARE DESCRIPTION—DISC MEMORY

3.1 Physical description

The disc file used in AIS is a modified Burroughs Corporation model BC475 disc file memory module. The BC475 is a head-per-track file using fixed air-bearing heads. The use of one head per track eliminates the need for head positioning. Two models have been developed for use in AIS, one with a storage capacity of 14.6 megabits, and the other with a capacity of 29.2 megabits. The lower-capacity file is equipped with two discs while the larger has four. Both models are equipped with the outer zone heads only, or one-third the heads normally supplied in the BC475. Pertinent characteristics of these memories are shown in Table I. The two-disc model omits the discs, heads, and other parts within the right-half enclosure.

Recording on the nickel-cobalt-coated discs is accomplished using the nonreturn-to-zero (NRZ) recording method (see Section 4.1). The read/write transducers, or "heads," are contained in assemblies of 13 heads each. When the discs are not rotating, the head assemblies are held retracted from the disc surface by head mounting springs. During operation, pneumatically driven pistons force the assemblies toward the disc surface. Opposing this action are the force of the spring and the force of the moving layer of air developed by the spinning disc surfaces. By applying a calibrated amount of gas* pressure to the pistons, the heads are forced toward the disc surface until they reach the point where all forces balance out. Thus, the heads literally "fly." In the KS-20512 unit, the head-to-disc spacing is approximately 75 micro-inches. Because of this close spacing, the discs are enclosed in an air-tight dust-free cover to prevent head instability problems that can be caused by dirt particles.

3.2 Data track accessing

In the disc file, only one data track can be accessed at a time. The 1-out-of-200/400-track selection is accomplished through the use of head switching diodes contained within the disc file. The information heads are grouped in sets of 100 heads per disc, 50 heads associated with each disc surface. The output leads from each set are connected to a head control circuit. The center taps of the 100 heads in each set connect in parallel to the 100 select lines in each head set. Selecting a head is accomplished by enabling one of the 100 head select lines

---

* AIS provides the option of using either nitrogen gas as in the No. 1 ESS ADF or air from a frame-mounted air compressor.

Table I — Disc memory characteristics

KS-20512	L1	L2
Disc Speed	1500 rpm	1500 rpm
Data Packing Density	1000 bits/inch	1000 bits/inch
Storage Capacity	14.6 megabits	29.2 megabits
Number of Discs	2	4
Number of Data Tracks	200	400
Number of Clock Tracks	8	16
Track for Track Index	1	2 (1-not used)
Cooling Fan	1	1

and one of the 4 head control circuits. With a center tap enabled and a head control enabled, the diodes in the desired head (1 out of 200/400) are forward biased, rendering the head active. For reading, the head control circuit connects the head to the read amplifier. For writing, the head is disconnected from the amplifier and a regulated current of 125 ma is switched to the head by the write switches in the head control circuit.

3.3 Clock head accessing

Each disc surface has a bit clock and a pattern clock recorded on it. These clocks are used to derive timing pulses used throughout the file complex (see Section 2.3). To minimize timing shifts between data and clock, the clock tracks selected are on the same surface as the data track being accessed. There is one center tap line for each pair of bit clock and pattern clock heads. The outputs of the bit and pattern heads on one surface of each disc are connected in parallel to amplifiers, one for pattern clock and one for bit clock. Similarly, the clock heads on the opposite surfaces are connected to two additional amplifiers. Enabling the appropriate center tap line forward biases the clock head switching diodes, connecting the head through to the amplifier. Two center tap lines are always activated, one to select clocks for timing purposes, and the other to provide bit clock for the disc frequency servo circuit.

The pair of timing tracks which are recorded on each surface consists of a bit rate clock and a unique 46-bit pattern. The bit rate clock is a rectangular wave having 73,600 cycles per disc revolution. The disc speed is nominally 1500 rpm, hence the bit rate clock is 1.84 megabits/second. The pattern is a pseudo-random arrangement of ones and zeros which is unique within any group of six adjacent bits (see Fig. 2).

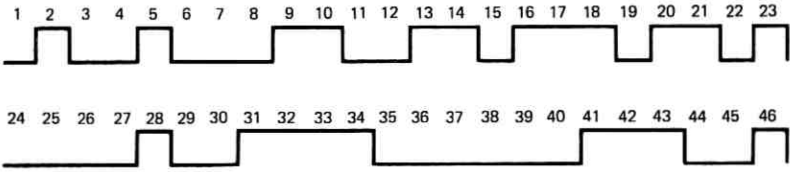


Fig. 2—Pseudo-random pattern for bit times.

Positional information is not contained in either the bit clock or the pattern clock. To provide this information a separate track containing a single "one" bit is used. This bit, called track index, defines the start of a disc revolution, and is used for many purposes in the file circuitry. The index track, bit clock, and pattern clock tracks are all written prior to use of the disc module in the system. Once the disc is in the system, these tracks can only be rewritten by special means.

3.4 Disc drive and servo

The discs are driven by a 2-hp, 208-volt, 3-phase, 60-Hz synchronous motor which is belt-coupled to the disc shaft. To insure the existence of a reliable power source that can be frequency controlled, ac power for the motor is converted from the normal office dc supplies by circuitry contained entirely within the file frames. A 360-Hz voltage-controlled oscillator, the frequency output of which is divided by a counter, supplies the required drive voltage. The voltage is transformer-coupled to the disc motor.

The disc speed is controlled by a frequency servo system to meet tight speed tolerances (the circuitry used is essentially a portion of that used in No. 1 ESS ADF).[1] The frequency servo circuit compares the bit clock frequency read from the disc to the frequency of a crystal-controlled reference oscillator and develops an error voltage proportional to the frequency difference. This voltage controls the frequency of the 360-Hz oscillator which, in turn, controls the speed of the drive motor. As a result, the disc speed is held to within ± 0.1 percent.

3.5 Disc temperature control

As with most telephone switching equipment, the file complex must be capable of operating over a temperature range of 35°F to 120°F. The disc file, however, should not be subjected to temperatures below 60°F mainly because of the shaft bearings. To overcome this limitation, the disc unit is enclosed in a temperature-controlled cabinet equipped with motor-controlled louvers and cooling fans. The operating range within the cabinet of 85°F to 120°F also serves to minimize shifts in data timing due to temperature effects.

3.6 Disc-related maintenance and protection

Numerous safeguards and checking features are built into the circuitry related to disc operation. The magnitude and duration of write current, proper switching of current between the 0 and 1 head windings, and correct head selection are a few of the more important checks performed. Failures detected by these monitoring circuits will result in the disabling of further head selection to prevent destruction of data. In addition to monitoring the data heads for multiple head selects, the clock head center taps are monitored for double enabling. Failures in clock output gating or in clock readout are detected by the bit clock checker which is capable of detecting the dropping of a single bit.

In the disc motor drive and servo areas, extensive checking is done both during motor startup and normal operation. Such factors as motor drive current, correct phase production, and servo operation are continually monitored. Certain failures will cause automatic power shutdown while others will result in the requesting of maintenance diagnostic programs.

Due to the nature of flying head disc memories it is possible under certain conditions for the heads to touch the disc surface causing damage. Very often damage to the surface can be prevented by detecting the presence of conductive dirt particles between the head and disc. This is accomplished by means of "touch" probes imbedded in each head assembly and a common touch detection circuit. The detection of a touch will cause automatic retraction of the heads from the discs.

IV. DATA ORGANIZATION

4.1 General

Intercept information is recorded on the disc serially by bit in binary-coded-decimal form. NRZ (nonreturn-to-zero) recording is used in which a change in magnetic flux occurs only when the logic level changes from one to zero or from zero to one. Data words consist of 42 data bits, 1 parity bit, and 3 guard bits. Three guard bits are used to allow single word alteration without data overlap problems.

Both surfaces of a disc are used for recording, with each surface divided into 50 tracks, all of which are located on the outermost "zone" of the disc surface. Thus, a four-disc unit has 400 data tracks. However, only 48 tracks per surface or a total of 384 tracks are used to store the actual intercept information. Of the remaining tracks, three contain "locator" information (indexing words) which expedite the associative number search; one track contains the nongeneric (installation-dependent) office parameters; and seven tracks are used by file maintenance programs for testing read/write circuitry. The rest are unassigned.

4.2 Word addressing

The disc surface is partitioned into 16 pie-shaped segments termed "sectors." Each sector contains 100 words divided into five groups of 20 words termed "blocks." A block is written in a five-word interlaced

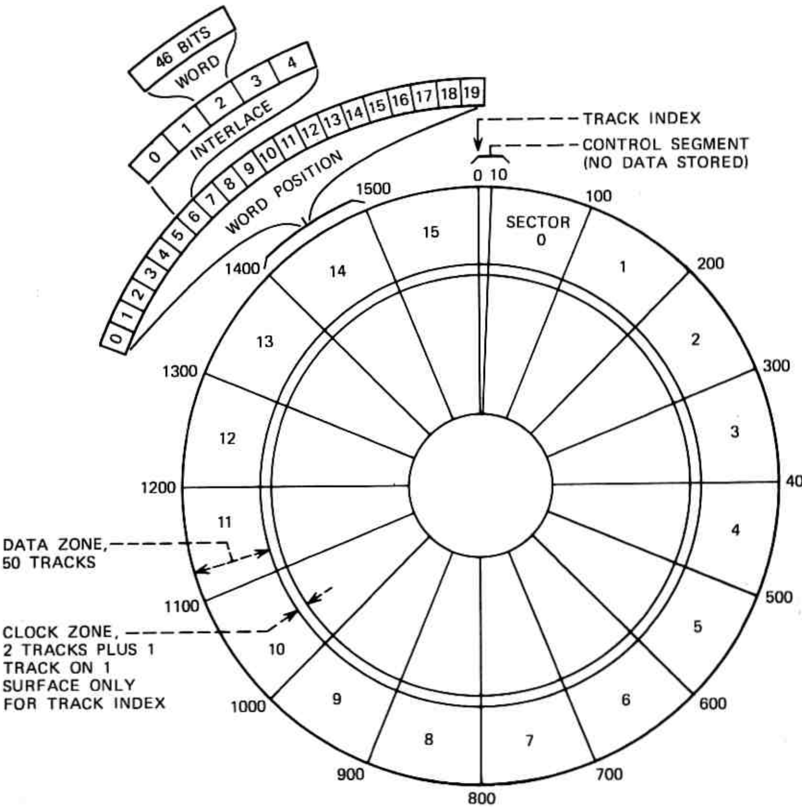


Fig. 3—Organization of data storage facilities on each disc face.

fashion where every fifth word belongs to a given block. However, data blocks are read or written as if the words were consecutive. The data are arranged in this way to match the data handling speeds of the file and control unit. The overall disc layout showing track locations, sectors, and interlaces is illustrated in Fig. 3.

Based on this data organization, a word location or address is defined by its disc (0–3), surface (0–1), track (00–49), interlace (0–4), sector (00–15), and word position (00–19). The transition from sector 15 to sector 00 represents the end of one disc revolution and the beginning of the next. This transition point is defined by the track index pulse, which sets the word count register to all 0's. Since clock and data head switching also take place at this time, the first two word positions of each interlace in sector 00 are not used to allow time for the head switching transients to subside.

4.3 Types of intercept record words

Intercept data words recorded on the disc fall into one of three categories: header or locator words, called or intercepted number words, and new number words. The formats for these three types of words are shown in Fig. 4. The tag on each word allows easy identification of the different types of words. The last bit of each word is odd parity computed over the entire word.

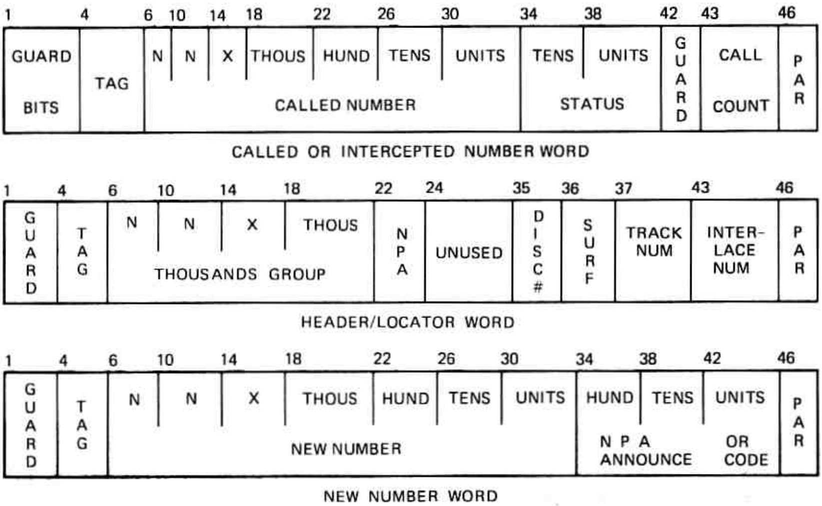


Fig. 4—Word formats.

4.4 Nongeneric data

Nongeneric (office-dependent) data stored in the CU call store memory are also stored on the disc file. This redundant storage serves two purposes. First, it provides a reference for audit programs to verify that the semipermanent call store data have not been changed erroneously. Second, in the event of a high-level system initialization, it is used in restoring the call store nongeneric data areas. These data contain information such as trunk assignments, announcement machine track assignments, lengths of queues, and type of AIS office (two- or four-disc, one- or two-network, etc.).

V. MODES OF OPERATION

5.1 General

The file has two modes of operation. The first is the associative lookup mode which is used for two purposes: (*i*) intercept number lookup by call processing programs; (*ii*) location of disc machine address information for use by file administration programs when inserting or deleting entries. The second is the block transfer mode whereby a block of data can be read or written under control of the CU. These operations are described in greater detail in the following sections.

5.2 Associative lookup mode

The associative lookup is initiated by programs in the CU passing to the File Control the seven-digit intercept number, the Numbering Plan Area (NPA) code of the intercept number, and a request code identifying the operation. The hardware sequencer then proceeds to control the lookup process using the intercept number as an index into the disc data base (see Fig. 5).

The sequencer selects a locator track at the start of the first revolution after the receipt of the command. This allows the reading of locator words, which consist of: (*i*) the first four digits of the intercept number and NPA code, and (*ii*) the track address of the data track where this grouping of numbers is stored. These locator words are compared serially against the corresponding portion of the intercept number being searched for and, when a match occurs, the track address portion of the locator word is loaded into a register.

At the start of the next disc revolution, the register containing the track address information is referenced by the track selection circuitry and this track is selected for reading. The sequencer directs the com-

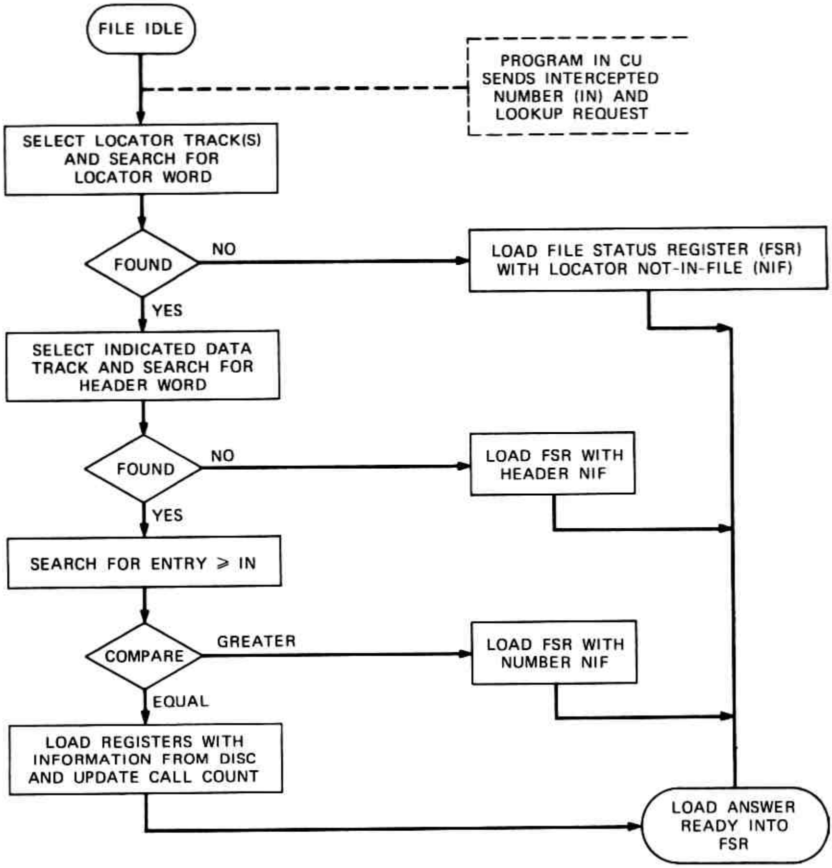


FIG. 5—Flowchart of intercept number lookup.

parison circuit to search for an entry on the data track which is identical to the locator word. This word is called a header word since it is the heading for all intercept numbers with the same first four digits (called thousands groups). When the header word is found, the sequencer begins searching the following data, comparing it with the full intercept number. The intercept number entries are sequentially ordered within each thousands group, allowing the comparator to search for either a match or a "greater-than" entry.

The match of the intercept number results in the following steps: (*i*) the status and machine location of the intercept number are loaded into registers for retrieval by the programs initiating the search; (*ii*) the sequencer waits for one revolution to rewrite the entry with a new call count (the call count is part of the entry indicating the number of times this number was referenced); (*iii*) if the next entry is a referral number (termed new numbers), it is also loaded into a register.

If a "greater-than" entry is found, the sequencer takes a different course of action. The machine location of the larger entry is stored away for use by file administration programs and the file status register is loaded with a "not-in-file" indication for the call processing program.

Upon finding either a match or larger entry, the file status register is loaded with an "answer ready" indication telling the requesting program that the action is complete. This indication is also set should the sequencer fail to find either the locator or header words.

The total associative lookup procedure requires from 80 to 520 milliseconds to be completed, depending upon point of entry, fill of data, etc. This is the most predominately used operation of the file.

5.3 Block transfer mode

The block transfer mode provides direct access to blocks of data on the disc. In this mode a block of disc words is either written or read in conjunction with a program operating in the CU. In order to accomplish this, the program (operating in the CU) and the file control (reading or writing on the disc) must establish synchronism. This is accomplished through the use of an interrupt structure in the CU.

The sequence (Fig. 6) is started when a program requests a block transfer, passing to the file the block address (track, sector, and interlace numbers) and the function request code.

The sequencer waits until the start of the revolution following the request, at which time the proper track is selected. The sequencer then compares the disc position with the address of the desired block. When the disc reaches the address, the file control sends the CU an interrupt signal. This calls in the program to transfer the block. This program reads the position of the disc and establishes synchronism with it. At the same time, the sequencer in the file circuit prepares to transfer the first word at the beginning of the block.

The block is transferred, utilizing the time provided by the interlaced data arrangement on the file for data manipulation. This manipulation includes error checking by hardware circuits in the file, transfer between CU and file, and storage in or retrieval from memory in the CU. The file reads or writes each fifth word on the disc in the selected sector, allowing 125 microseconds between successive words.

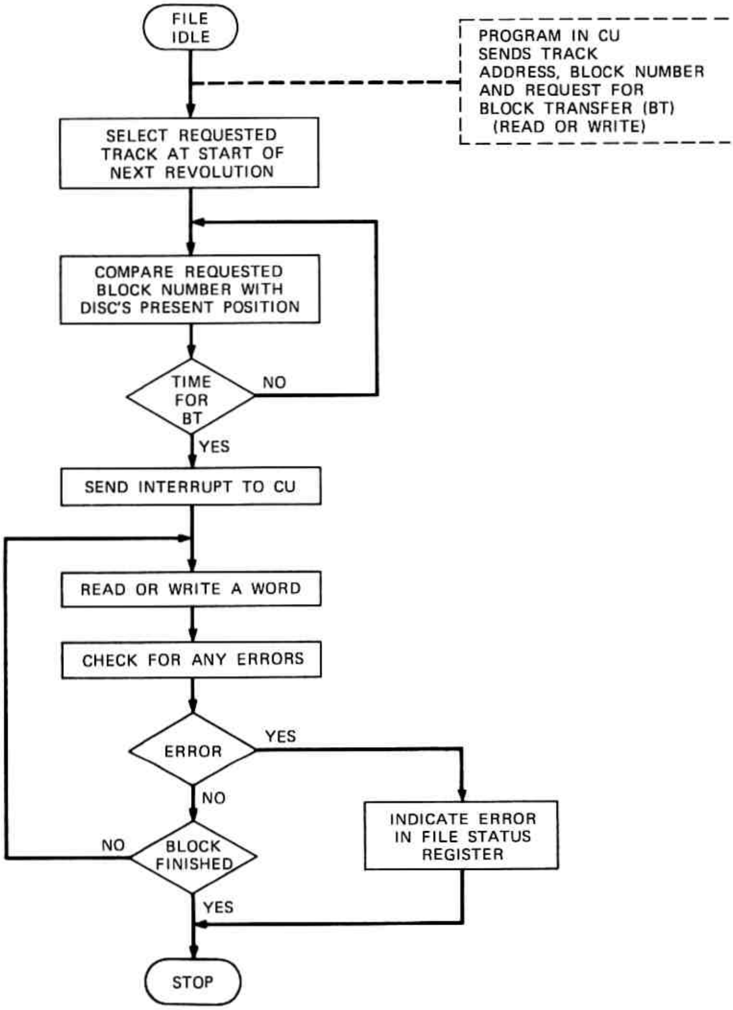


Fig. 6—Flowchart of block transfer.

When the entire block is transferred, the sequencer indicates in the file status register whether or not an error has occurred. The program reads the file status register at a future scanning time and takes action accordingly.

Block transfers, which are used by file administration to restructure data on the disc[3] take from 3 to 80 milliseconds to complete.

Other principal users of this mode are file audit programs, which need to read the disc in order to validate information stored on it, and the call store audit program to validate the nongeneric office parameters in call store.[4]

VI. ERROR DETECTION HARDWARE

Error detection hardware constitutes an integral part of the file design. This circuitry monitors the operation of circuits within the file and sets appropriate indicators in the file status register when an error occurs. Program action as a result of these indicators is discussed in Section VII.

Besides setting error indications in the file status register, certain errors, namely those associated with disc reading or writing, cause write-abort circuitry to operate. Activation of this abort circuit immediately stops disc writing operations. This is done to minimize possible mutilation of recorded data in the event that an error occurs in the read/write chain.

VII. RELIABILITY AND MAINTAINABILITY

Very high reliability is an essential characteristic of any real-time telephone system. This basic reliability objective dictated much of the file subsystem design philosophy. Hence, several important techniques were used to attain high reliability. Specifically, the file subsystem is completely duplicated with each file having sufficient call handling capacity to serve normal traffic loads with acceptable queuing delay. This means that one file subsystem can be down for repair without an appreciable effect on call processing performance. However, to keep the data bases in agreement, updating of the intercept records on disc is normally done only when both files are available.

In order for duplicated units to provide the desired high reliability, the repair time associated with each must be held to a minimum. In other words, the units must have high maintainability. Thus, another important designed-in feature of the file, intended to improve fault location and therefore reduce file down-time, is the incorporation of numerous test points accessible to the control unit. These points are of two types: the first type allows the control unit to examine the states of various internal file registers and flip-flops; the second type allows the control unit to set or reset selected flip-flops within the file complex. This circuitry provides the ability to stop and start clocks, read up to 16 flip-flops or gate outputs with each command, and set and/or clear most flip-flops and register stages. These features allow the programs to check both sequential and combinational circuits as if they were all combinational circuits.

A flip-flop register, called the Memory Register, is provided with multipled inputs, which can be selected by program from the CU. This register can be used to take "snapshots" of circuits while they are in operation. In this way, the occurrence of pulses can be detected, providing a powerful diagnostic tool in analyzing trouble symptoms.

Still other techniques applied to the file system to improve its reliability by minimizing its down-time are automatic error detection, location (also referred to as diagnosis), and system reconfiguration. These are accomplished by a combination of hardware and software methods. Checking circuits have been used in many areas of the design to provide immediate automatic hardware alarms in the event of component failure. Detection programs are provided which run periodically to establish whether the file is functioning properly and to take the proper action if a trouble is detected.

VIII. ERROR DETECTION AND LOCATION SOFTWARE

8.1 General

Diagnostic programs, used to locate faults within a file complex, can be requested either automatically or manually. Requests for automatic diagnostics may be initiated as a result of errors detected by any of the programs normally using the file or by system monitoring programs. Manual diagnostic requests can be made by maintenance personnel from the maintenance center teletypewriter.

8.2 User programs

All programs which use the file subsystem check for error indications in the file status register. These programs include the call processing program, the file administration program, the rapid scan audit program, and the routine exercise program.

Each of these programs is discussed below in somewhat more detail to show how they detect file troubles and request file diagnostic programs to be run automatically.

8.2.1 Call processing program

The predominant user of the disc file is the call processing program which may initiate several thousand intercept number lookups during the course of a busy hour. When a fault is detected by the file error detecting hardware, an error indication is set in the file status register. For example, if a parity error occurs during a normal lookup sequence, an appropriate bit is set. Another type of error that might be uncovered by call processing is a time-out, in which the lookup sequence has not been completed within its maximum cycle time. Either of the errors mentioned above would cause the call processing program to request a diagnostic test to be run on the file automatically.

8.2.2 File administration program

The file administration program which manages the data base performs error checks similar to those of call processing.

8.2.3 Rapid scan program

The function of the rapid scan program is to detect inconsistencies in file status indicators. The status of a file is kept both in file hardware and the control unit memory. When discrepancies are found, the rapid scan program attempts to reconcile the differences. As a result, diagnostic tests on a file may be requested automatically.

8.2.4 Routine exercise program

The routine exercise program tests not only the operational hardware but also the error checking hardware to insure that the latter will indicate a trouble should one occur. For example, one can cause a parity error to verify that the parity checker operates properly. These tests are run every 4 hours on each file, interleaved on a 2-hour basis. If an exercise program fails a test, it requests a file diagnostic to be run automatically.

8.3 Example of a file trouble

The following example illustrates the sequence of events which transpire when a fault occurs in a file. Consider the case where a flip-flop in the bit clock register becomes permanently "stuck-at-one." Any one of several programs could detect this condition, but in this case assume that an intercept number lookup is in progress. The bit clock error checking circuit causes an error bit to be set in the file status register. During each 25-millisecond interrupt, the call processing program reads the file status register to see if the lookup is complete. In the 25-millisecond interrupt following the fault, the call processing program finds the error indication. The sequence of events will then be:

(*i*) The file is marked "Maintenance Busy" so that no other user program attempts to use it.

(*ii*) The call processing program places a request with the system maintenance monitor for the file diagnostic program to be run.

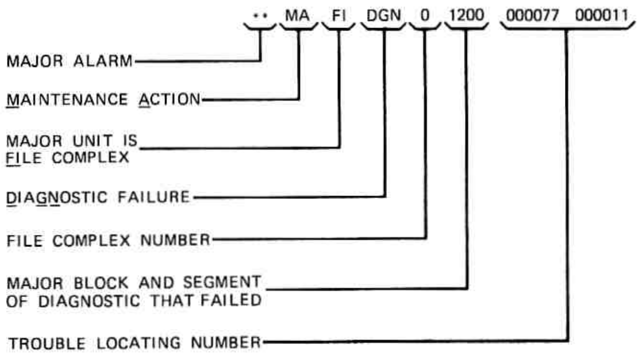


Fig. 7—Teletypewriter message for file trouble.

(*iii*) The file diagnostic is run and locates the faulty circuit.

(*iv*) The file is placed out-of-service and a message is printed on the maintenance teletypewriter (see Fig. 7).

(*v*) The next step requires action by the maintenance craftsman to correct the problem.

8.4 Call store mismatch due to file complex troubles

When the two CUs are running in synchronism, the CSI (*C*all *S*tore *I*nput) registers in the two units are compared at each call store read or write. If they differ, a mismatch interrupt will be generated. This means that if an attempt is made to write different data into the two call stores, a mismatch will occur. Certain file complex troubles can occur which could cause different data to be sent to the two control units. This type of fault will cause a control unit mismatch interrupt if:

(*i*) the received data are written into call store, or

(*ii*) a test and branch sequence is executed on the data that differ in the two control units.

The mismatch strategy used in the Automatic Intercept System is essentially that used in the No. 2 ESS.[5] The control unit mismatch recovery scheme, resulting from a control unit error, is discussed elsewhere. Recovery from a file error is based on this strategy. First the online control unit is tested. If it passes its tests, the communications links between the online control unit and each file are tested. If one of these tests fails, a low-level (Phase A) system initialization occurs. As a result, the "suspect" control unit is switched offline and diagnosed along with the file, since communications between the control unit and the file is in question.

If the online CU-file tests pass, but the tests between the offline control unit and a file fail, then the offline control unit–file communications diagnostics will be requested.

If a communications problem is found, a bit is set in a "bad access word" indicating that a trouble exists between the offline control unit and the specified file. When the problem is cleared and a control unit restoral is requested from the teletypewriter, offline CU-file diagnostic programs will be run automatically to insure that the access problem has indeed been cleared before the control units are put back in synchronism.

IX. FILE MAINTENANCE PROGRAMS

9.1 File maintenance monitor

A file maintenance monitor program controls the sequencing and priorities of all file maintenance programs.

In general, the priority structure in descending order is:

(*i*) Automatic requests based on suspected troubles either through hardware checking circuit activation or routine exercise detection.
(*ii*) Teletype requested diagnostics.
(*iii*) Periodic timed routine exercise programs.

If a diagnostic is being run when a higher-priority diagnostic is requested, the lower-priority diagnostic will be terminated gracefully and may be rerequested under program control once the higher-priority diagnostic is completed.

The function of the monitor is to:

(*i*) Determine if a test can be run.
(*ii*) Control the sequence of tests which are to be run.
(*iii*) Provide a common interface for the numerous file diagnostic blocks.
(*iv*) Screen the file input teletypewriter requests for appropriateness.
(*v*) Provide the appropriate teletype output messages.

9.2 File diagnostic program

The file diagnostic program consists of a sequence of diagnostic segments which start out testing the CU-file communications circuits and progressively test the more complicated internal parts of the file complex. This results in a systematic buildup of tested circuit elements which can be relied upon in subsequent tests. Because of this building-block approach, the tests must be run in a prescribed order. The major advantage of this approach is that the program can stop on first failure. The raw data results obtained, along with the number of the segment being run, are sufficient to locate the trouble to within several replaceable circuit packs. With this building-block type of diagnostic, one can find easily the particular program being executed when the failure was encountered and identify the particular test in the sequence that failed. This information may be useful to maintenance personnel in cases where additional analysis is necessary.

The sequence of tests forms a natural segmenting of the diagnostic programs. This is important since diagnostic programs are executed as part of the base level program and the length of each test is restricted to a maximum of 5 milliseconds in each base loop.

The tests are done in the following order. First, the control unit-file access circuits are tested. Next, the registers most closely associated with the peripheral unit address bus and the scan answer bus are tested. After the remaining static registers are tested, the dynamic registers (counters, etc.) are tested in two ways, first statically and then dynamically. This technique of testing dynamic registers in a static manner by stopping the internal file clock and providing simulated clock pulses under program control has proven to be a very powerful diagnostic method. It permits sequential circuitry to be tested as if it were combinational circuitry. However, dynamic tests are also run to detect marginal or speed-dependent failures which might not be detected in the static tests.

9.3 Automatic reconfiguration of file subsystem–control unit

Each control-unit-initiated order to a file is responded to with an enable verify (EV) pulse indicating that the unit received an order. For scan orders, where a response is expected, an all-seems-well (ASW) signal is sent with the response, again indicating that there are no known communications problems between the online control unit and the file. When the control units are running in synchronism, each unit executes identical code, including external orders, except that only orders from the online control unit are sent over the bus; the standby unit has its outputs inhibited.

The control unit tests for this immediate EV response (and ASW where applicable) and sets an indicator in the event a response is missing. Under software control, a reconfiguration program (also called a working mode program) may be called. In a series of steps, this reconfiguration program attempts to establish good communications between a control unit and the file which did not respond (Fig. 8). First, it retries the order since the failure may be of a transient nature. If the order fails a second time, the program then requests a switch to the other control unit. Once this is done it again retries the order; this time the new online control unit sends the order. Two separate actions are taken at this point depending on the outcome of the retry attempt:

(*i*) Success on retry after CU switch: this implies that either the now offline CU and its associated bus cannot send proper data, or that the related receivers in the file are faulty. In either case, the offline CU cannot communicate with the file and the offline CU is placed out-of-service. A program then tests the communications between the offline CU and the file to locate the trouble.

(*ii*) Failure on retry after CU switch: this means that neither CU can communicate with the specific file. Based on the "single-

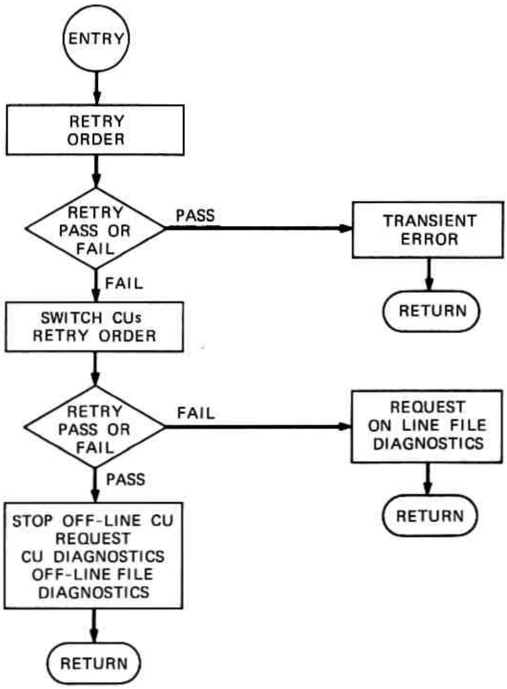


Fig. 8—CU-file reconfiguration procedure.

failure" philosophy, this implies that there is a hardware problem in the file. The file is removed from service and file diagnostic programs are requested to locate the problem.

X. FILE MAINTENANCE USING THE MAINTENANCE CENTER TELETYPEWRITER

10.1 General

A number of special-purpose programs, accessible from the maintenance center teletypewriter (MTC-TTY), provide the central office craftman with a set of extremely flexible tools with which to maintain the file subsystem. From the MTC-TTY, such actions as determining file status, changing file status, and performing detailed or large-scale tests can be accomplished.

10.2 Teletype-requested functions

The following is a partial list of TTY-requested operations that can be performed on the file subsystem:

(*i*) Request a printout of status information—available, out-of-service, etc.
(*ii*) Request running of full or partial diagnostic tests.
(*iii*) Remove a unit from service.
(*iv*) Make a unit maintenance busy.
(*v*) Restore a unit to service.
(*vi*) Generate specific commands or orders to load or read most registers and/or particular flip-flops.
(*vii*) Request a file-look-up of a specified number and a printout of the information found.

10.3 Typical teletypewriter sequence

A typical teletypewriter interchange between a maintenance craftsman at the MTC-TTY and the file software system is illustrated in Fig. 9. In this example, the craftsman wishes to remove file complex 1 from service for the purpose of making a circuit change, performing preventive maintenance, replacing a suspect circuit pack, or some similar function. He proceeds as follows (the following steps refer to the message numbers in Fig. 9):

(*i*) The file status is requested. The system responds with "PF" (printout follows).
(*ii*) The file status is printed as "0 AVL" (complex 0 is available), "1 AVL."

```
M FI : SI ! PF                          (1) - (INPUT)

+tt MR FI SI O AVL 1 AVL                (2) - (OUTPUT)

M FI : RMV : 1 ! OK                     (3) - (INPUT)

M FI : SI ! PF                          (4) - (INPUT)

tt MR FI SI O AVL 1 ØØS                 (5) - (OUTPUT)
.
.
.
.
M FI : RST : 1 ! IP                     (6) - (INPUT)
.
.
.
.
tt MR FI DGN 1 1500 000001 074200       (7) - (OUTPUT)

+ tt   MINUTES AFTER THE HOUR.
```

Fig. 9—Typical file maintenance TTY exchange.

Having ascertained that both files are available for system use, he reasons that file 1 may be removed for maintenance.

(*iii*) The system is requested to "remove file 1 from service." It responds with "OK."

(*iv*) The new file status is requested.

(*v*) The system now shows the status of file 1 as "1 OOS" (out-of-service).

Having isolated the file complex and prevented its active use by the system, the craftsman can now perform the desired maintenance function. Upon completion of the task, he attempts to restore the file to service.

(*vi*) The system is requested to "restore file complex 1." It responds with "IP" (in progress).

Before the system will restore the file, however, a complete diagnostic test will be automatically requested to insure that the file is functioning properly. Should the diagnostic pass, the system will respond with

tt MR FI RST 1 OK

and the file will be made available for normal use.

(*vii*) The file fails the diagnostic tests and the failing diagnostic segment (1500) is printed along with a "trouble number." The file complex is left out-of-service.

10.4 Isolating the trouble

To aid in locating the trouble, a Trouble Locating Manual (TLM) is referenced using the diagnostic segment and trouble numbers. This manual will supply the craftsman with a list of circuit packs to replace. Following the pack replacement, he may elect to restore the file—with resulting diagnostic—or he may simply request a full or partial (segment only) diagnostic directly.

Should the above procedure fail to effect a repair, other facilities are provided to aid in fault correction. One powerful feature, available from the MTC-TTY, is the ability to execute some of the teletypewriter requests on either a repetitive or a "one-shot" (i.e., repeat on request) basis. For example, a subtle or difficult-to-find trouble might be located by repetitively running the failing diagnostic segment, thereby enabling the craftsman to use an oscilloscope to trace the signals through the suspected circuit. Similarly, a repetitive lookup of an intercepted number may be requested. Using this method, the lookup would be repeated over and over, allowing the operation of the sequencers to be analyzed. Or if, as the result of a diagnostic, the trouble is localized to a group of several circuit packs, the diagnostic can be repeated on a "one-shot" basis following the replacement of each pack until the fault is corrected. To facilitate these features, a hand-held key/lamp assembly is provided which connects to the system Maintenance Center via a belt-line that strings through all the major frames in the office. Operating the key will stop and restart a repetitive test or simply trigger a one-shot test. The pass or fail results of the diagnostic are displayed on the lamps.

XI. GENERATION OF THE TROUBLE LOCATING MANUAL

In some systems, the Trouble Locating Manuals have been generated automatically by inserting faults, one at a time, running the diagnostic programs, and operating on the results with a data reduction program.[6] For the AIS file complex, however, the TLM generation was essentially manual, that is, it was formulated by the diagnostic programmer through circuit analysis. Both schemes have advantages and disadvantages. The principal disadvantage of the automatic scheme is that circuit changes require that a large part of the fault insertion procedure be redone. By comparison, the analysis method requires only the minor TLM changes that correspond to the actual circuit changes. The analysis method, on the other hand, requires that the TLM be prepared by experienced circuit technicians. However, these skilled people are usually required to write the detailed hardware diagnostic programs in the first place. Thus, the TLM is obtained with a small additional investment while the programs are being written.

XII. SUMMARY

The file subsystem has been designed to provide both fast and reliable storage and retrieval of intercept records. To achieve fast access, it was designed with two basic modes of operation. One mode, the lookup mode, allows the file to do associative searches of its contents independently of the CU. The second mode, the block transfer mode, permits the high-speed transfer of blocks of data between the control unit and the file.

To provide a reliable and maintainable system, a great deal of additional hardware and software has been incorporated in the design. This includes file duplication and the incorporation of error detection and checking circuits within each file. Furthermore, detection, diagnostic, and system reconfiguration programs are provided to quickly isolate faults in a subsystem to within a few replaceable circuit packs.

REFERENCES

1. E. J. Aitcheson, C. F. Ault, and R. G. Spencer, "No. 1 ESS ADF—Message Store —A Disk Memory System," B.S.T.J., *49*, No. 10 (December 1970), pp. 2887–2914.
2. W. B. Cagle, R. S. Menne, R. S. Skinner, R. E. Staehler, and M. D. Underwood, "No. 1 ESS Logic Circuits and Their Application to the Design of the Central Control," B.S.T.J., *43*, No. 5 (September 1964), pp. 2055–2095.
3. J. H. Carran, K. E. Greisen, W. G. Hall, and D. J. Wells, "Automatic Intercept System: Administering the Intercept Data Base," B.S.T.J., this issue, pp. 133–153.
4. H. Cohen, D. E. Confalone, B. D. Wagner, and W. W. Wood, "Automatic Intercept System: Operational Programs," B.S.T.J., this issue, pp. 19–69.
5. H. J. Beuscher, G. E. Fessler, D. W. Huffman, P. J. Kennedy, and E. Nussbaum, "[No. 2ESS] Administration and Maintenance Plan," B.S.T.J., *48*, No. 8 (October 1969), pp. 2765–2815.
6. R. W. Downing, J. S. Nowak, and L. S. Tuomenoksa, "No. 1 EES Maintenance Plan," B.S.T.J., *43*, No. 5 (September 1964), pp. 1961–2019.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 53, No. 1, January 1974
Printed in U.S.A.

Automatic Intercept System:

Administering the Intercept Data Base

By J. H. CARRAN, K. E. GREISEN, W. G. HALL,
and D. J. WELLS

(Manuscript received October 5, 1971)

An Automatic Intercept System data base of up to a half-million changed or disconnected telephone numbers is updated, corrected, verified, abstracted, restructured, and restored through the actions of a collection of function-oriented subprograms. These subprograms run in the base-level main program loop under their own monitor which also controls interrupt-level accesses to the asychronous disc memory. The monitor together with the set of subprograms provides a file administration capability which responds to both machine stimuli, such as timed entries or trouble indications, and human requests initiated from teletypewriters.

I. INTRODUCTION

The Automatic Intercept System (AIS) assembles machine announcements for calls to telephone numbers which have been changed or disconnected. Such calls are switched to intercept trunks in many local offices connected to one Automatic Intercept Center (AIC). The dialed numbers are transmitted automatically to the AIC by local offices equipped to do so or by operators when local offices are not so equipped. The AIS also provides special handling for calls to numbers which have never been equipped and for calls to lines on which a trouble condition has been marked at the local office.[1]

The principal data base, containing as many as a half-million directory numbers, is stored in duplicated disc memory units.[2] Clerical personnel keep it current with additions, corrections, and deletions of numbers on intercept in all connecting offices.

A distinctive portion of the system program provides for these updating functions and for verifying, abstracting, restructuring, and restoring the data base while the AIC continues to process telephone calls. These actions, all referred to as file administration, respond to internal stimuli such as timed routine entries or trouble indications as well as to external requests. Interactive teletypewriter input/output (I/O) provides access for both clerical and maintenance requests through the No. 2 ESS processor[3] which is part of the AIC.

II. FILE ADMINISTRATION OPERATIONS

The information filed with each intercepted directory number in the data base includes a status code, a count of inquiries, and, if appropriate, a new number to which calls are referred by automatic announcements. The status and the new number are changed only by the administrative programs, which remove and insert the whole entry. The call count is incremented by hardware (to a maximum count of seven on each disc memory) every time an inquiry is made for the number.

Changes to the data base are made only by human intervention. An insertion is made when a number is disconnected, and a deletion when it is reassigned to an active line. Numbers in active service are not kept in the AIS file. Numbers which have never been in service are covered in the file as soon as the connecting central office is equipped to divert calls to the AIC, but may be noted in a single entry for a group of 100 or 1000 until individual assignments to active lines begin breaking up the group.

Most file administration functions are handled on a single-server basis; that is, only one action is undertaken at a time and additional overlapping requests are rejected. External requests are accepted through four different teletypewriter channels, three of which are intended primarily for various plant maintenance purposes. Only one, the file administration teletypewriter, is used for the routine clerical work of updating the data base. (See Fig. 1.)

2.1 Updating

A typical update message, though very brief and stylized, takes three seconds of teletypewriter transmission time for the order and an "OK" response plus a second or two of elapsed time for processing. The system is arranged to control a 10-character-per-second paper tape reader at the teletypewriter for batched clerical operation. An interface is under development which will also provide for an optional 2000-bit-per-second data link.

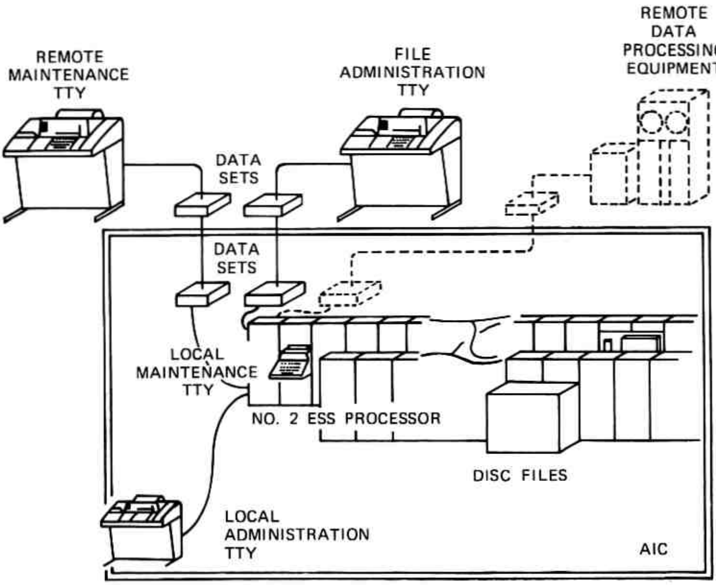


Fig. 1—External I/O channels for intercept data.

Routine updating transcribed by clerks from commercial service orders may average as many as 8000 original transactions per day, depending on the size of the data base and the mobility of line assignments in the telephone population served. Message originations and handling times are inflated by clerical errors and data inconsistencies, which result in retries and sometimes extensive response printing. In the presence of these anomalies, the system can handle 500 original transactions per hour. This requires efficient organization of the work flow feeding messages to the AIC and responses back to the clerks, as well as relegation of other file administration uses to separate hours.*

2.2 Other interactive uses

The other community of file administration users, the plant maintenance people, are called on occasionally to help the clerks correct any machine-related data anomalies. Plant also uses file administration functions to obtain data-related clues to machine troubles, using both interactive and internally stimulated messages. Hardware troubles

* A minicomputer-based File Administration System (FAS) is now under development which will assist the preparation of update messages and speed their flow, using the 2000-bit-per-second data link.

which automatically remove a disc memory and its controller from service preclude further routine updating, which must usually process both files in sequence. Thus, maintenance activity required to clear the trouble is preemptive. Call processing accesses continue in the duplicate file and the file administration monitor accesses the out-of-service file as requested by the diagnostic programs. In the event of a prolonged outage, a special condition can be instituted temporarily to permit updates to be done in one file only.

2.3 Out-of-hours activities

Some file administration functions are scheduled for light-hour operation. Routine tests of the validity of the intercept data are timed to start spontaneously at 1 a.m. each night. It takes from a quarter-hour to a half-hour to audit one-eighth of the file, plus time to print a record of all data anomalies found. When this is finished, the paper tape reader at the file administration teletypewriter is turned on by program. This provides a convenient means to obtain unmanned initiation of other actions in the middle of the night.

One routine job that lends itself to night turn-on is the abstracting of call count data from the file. This is done by printing directory numbers with counts less than a specified threshold for each of a desired list of central office designations, then resetting the counts to zero to start a new statistical period. Schedules for obtaining these statistics are set locally to provide data for reassignment of numbers to active lines. An hour of printing can provide call counts for about 1500 directory numbers.

2.4 Backup actions

Less frequently, a backup copy of the data base is created to guard against the remote possibility of loss of data from both of the duplicate disc memories. Depending on local practice, the backup may be maintained in an offline spare disc memory or in reels of paper tape or both. These are supplemented by paper tapes recording ensuing daily updating inputs. Eventually, other offline media will be accessed via the higher-speed data link. Copying the data base is done by the file administration programs while calling traffic is being handled, but must be scheduled to avoid routine updating work.

The file administration programs also provide for noninterfering system accesses to nongeneric office parameters[4] stored in the disc memory as a backup for call stores. This data base, three orders of magnitude smaller than the file of intercepted directory numbers, is maintained separately. System accesses to it have negligible effect on file availability for intercept record functions.

III. FILE ORGANIZATION

The AIS disc word has 42 usable bits, so that it is capable of storing ten binary-coded-decimal (BCD) digits. The formats used include:

(*i*) a seven-digit called number with two status digits and one call count digit,
(*ii*) a seven-digit referral number with three numbering plan area (NPA) digits, or
(*iii*) a four-digit or header number with associated machine address information.

Figure 2 shows the file organization.

The first two bits of the word form a tag which indicates which format has been used. A special pattern is recorded with a called number tag to mean that the location in which it resides is blank, i.e., available for storage of intercept record data. In a different context, the word can also be used in a free format, for example, for storing call store backup information on the special track reserved for that purpose.

The hardware environment has naturally induced two different means by which these numbers can be accessed and manipulated. On one hand, any number or group of numbers can be referred to on a machine address basis. That is, the user can retrieve a particular word, a block (18 or 20 words), an interlace[2] (16 blocks), or a track of data (5 interlaces totaling 1590 words). In addition, he has available operations which use many tracks up to and including the whole file (384 data tracks).

On the other hand, the implementation of a hardware search capability in the file complex[2] has induced a "thousands group" type of categorization. The clerical personnel can avail themselves of functions which operate on the data associated with one or more locator words. A locator contains four high-order digits of a telephone number, so that there are one thousand numbers which it can refer to. Each such group is called a thousands group.

The locator words on three specially accessed tracks act as a machine-address index to the intercepted directory numbers, so that the thousands groups form relocatable sets of data. Each thousands group is bounded by a header word and an end mark, and within this set of data all non-blank called number words are kept in ascending

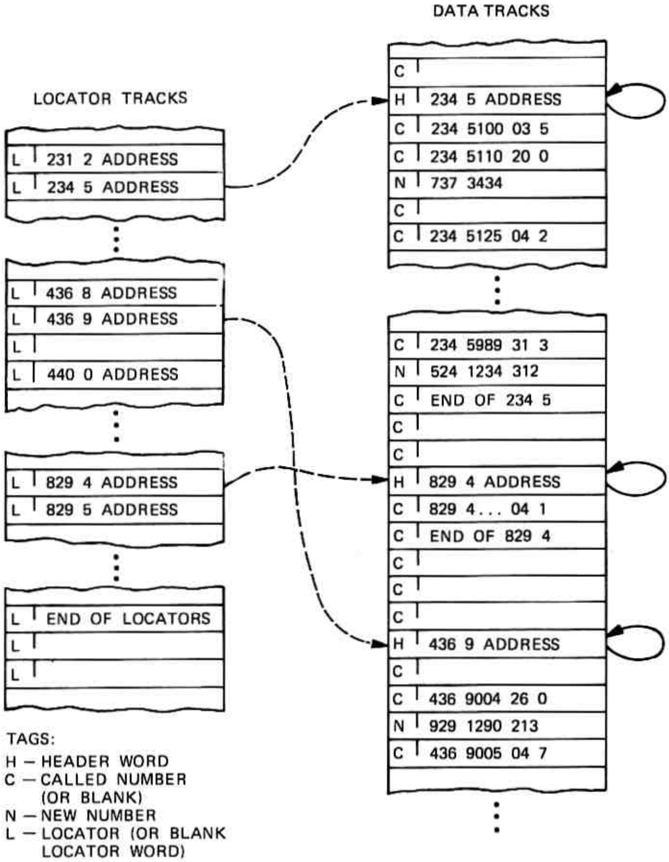


Fig. 2—File organization.

numerical sequence. New number (referral number) words are filed directly after the called numbers they pertain to. Blanks can be almost anywhere, though certain distributions are operationally desirable. Both the file administration programs and the hardware search features rely on these constraints.

IV. CONTROL PROGRAMS

4.1 File monitor

All file data handling except for call processing takes place under the auspices of the file administration monitor. This is basically a table-driven executive which is activated once each base-level loop.[4]

During each execution it updates an internal record of the state of the AIS file subsystems and either invokes an appropriate subprogram or, in case no work is to be done, returns control to the other base-level programs. This base-level execution may vary from a minimum on the order of a few hundred microseconds, when no work is to be done, to a maximum of about 3 milliseconds for the longest subprogram. In addition, the monitor must call for and manage a number of programs which operate in three interrupt levels, all higher than base level.

Every file administration function, including those executed on internal request, is defined by means of a table of addresses contained in program store. Each address is a transfer reference to a subprogram. A file administration subprogram is a program module designed to accomplish a simple operation. It is given control by the monitor, and when it has completed its task it returns control via one of several entry points to the monitor. Typically a subprogram can test or move data, request file I/O operations of either kind, or execute teletypewriter actions.

Under normal circumstances when a file function is in progress, the monitor accesses the function table once each base-level loop, and executes the associated subprogram. These table accesses are accomplished with the use of two call store words, one which indicates which function is in progress, and the other which is used as an index. Proper manipulation of these words achieves in a simple manner the ability to retrieve the subprogram addresses in consecutive sequence, to repeat a subprogram, to "branch" to other tables, or to loop on a sequence. It is also possible to execute a plurality of subprograms in one base-level execution. The choice, however, of performing loops, branches, etc., is made in the subprogram, and implemented through the use of the different entry points in a return to the monitor.

Requests for file administration activity come from several sources. They can be received from teletypewriter channels, from a 24-hour timer, from programs which audit and restore office parameter data, or from file diagnostic programs. The first two sources are referred to as data management sources; the same restrictions and priorities are applied to both. These data management requests are usually for relatively long and involved functions, while the call store audit and file diagnostic requests are always for highly specialized "single-shot" data transfers.

Nominally the monitor realizes three priority divisions established by the order it uses to examine request indicators. In order from first to last, they are requests from call store audit programs, from data management sources, and from file diagnostics.

Call store audit programs request only a transfer of a single block of office parameter data at one time. If such a request is received while another function of lower priority is in progress, the monitor suspends the function, processes the new request, and returns the previous function to an in-progress state. Neither data management nor file diagnostic requests can interrupt each other, e.g., if a file diagnostic request is in progress, it cannot be interrupted by a request from file administration.

4.2 File I/O

Two types of file I/O functions are available: lookups, which typically hold the file complex busy for 160 milliseconds, and block data transfers, which average 40 milliseconds. In file administration use, lookups are called on for the purpose of obtaining machine addresses, rather than referral information as in the call processing usage. The lookup I/O routines are capable of procuring the addresses of data words or of locator words on their respective tracks. The

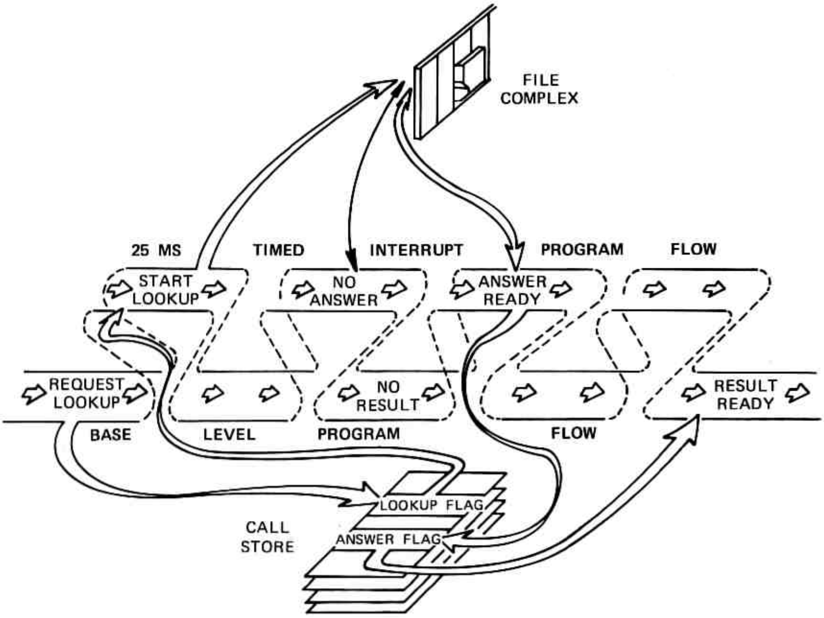


Fig. 3—Data lookups for file administration.

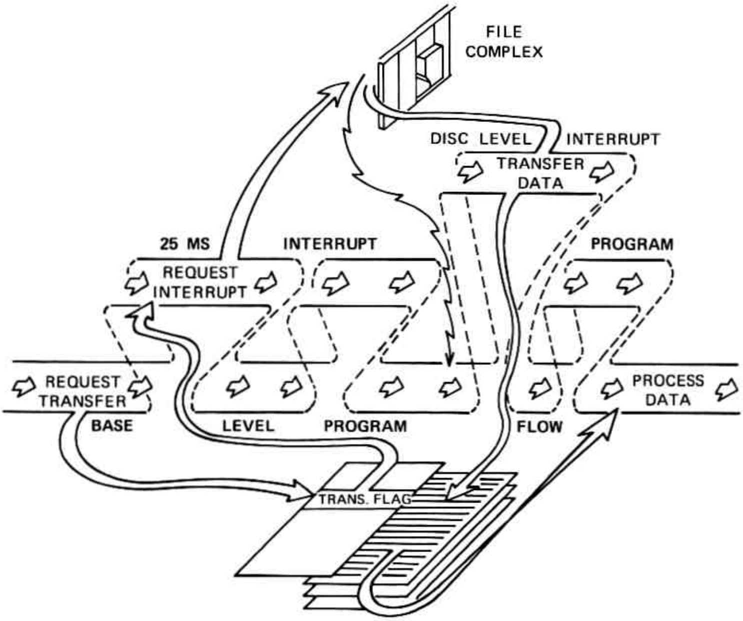


Fig. 4—Block transfers.

block transfer operation is used either to read a block of information from disc and load it in a call store buffer, or to do the reverse, i.e., write the file from call store. File usage for the most commonly requested functions entails a lookup followed by one or more block transfers.

Requests for I/O actions are executed in the same manner as data managing subprograms, in base level. All requests are recognized by a routine in the 25-millisecond timed interrupt level, and the order for action is then sent to the selected file control unit. (See Fig. 3.) If the request is for a lookup, another routine in the timed interrupt level scans the file status register in the file control[2] for an "answer ready" indication. When the indication is received, the address sought is retrieved from the track address register in the file control.

In the case of a block transfer, however, the file control generates an interrupt request in the processing unit when the proper disc memory address has been reached.[2] There are two disc interrupt levels which can be requested. Each is associated with only one file control, and both are of a higher level than that of the timed interrupt. (See Fig. 4.) The program affiliated with each disc interrupt level checks the validity of the interrupt request. If it is acceptable it transfers to a main I/O program which then scans registers in the file control until it is in close synchronism with the disc memory, and at this time it transfers the data in the desired direction.

The fact that the processor and the disc must be brought into synchronism for the duration of a data transfer introduces a restriction on file administration activity, i.e., only one file complex at a time can be used by the program in progress. Designing with this restriction (and applying it also to lookups) guarantees that a file will always be available to the call processing routines.

The high reliability of the AIS file subsystem[2] has warranted the use of an uncomplicated error-handling algorithm. If an I/O program detects a trouble indication and passes this result to the monitor, the I/O request is simply repeated. If no subsequent indication is detected on the second attempt, the function continues to progress normally. However, when the indication is repeated, the function in progress is aborted and teletypewriter output detailing the circumstances is generated. Basically, then, error handling consists of a single retry for all I/O functions.

4.3 File use under adverse conditions

The philosophy of changing data on the disc memories in normal circumstances is to modify one copy of the data base at a time. For example, when a number is added, a sequence of subprograms enters the number in one file complex, then the same insertion algorithm is repeated in entirety using the other file. This guarantees that if a file complex fails during either sequence, then at least one disc memory will contain completely valid data with or without the new entry, depending on how far execution proceeded.

If a function is aborted due to a persistent error, it will possibly produce a mismatch between disc memories: a number residing on one file complex and not on the other. In a case of this sort, manual intervention is required via teletypewriter-requested data maintenance functions (Section 5.1) to correct the condition. During such a mismatch condition, both call processing and file administration continue to function successfully. However, they will print messages calling attention to the anomaly whenever it is encountered. In addition, if neither the original abort message nor any subsequent mismatch messages gain attention to the anomaly, the nightly routine validity tests will also find and record it.

Consistent with the philosophy of redundant updating, an input message normally will be rejected if a file complex is unavailable to call processing at the time of request. There are, however, some special conditions which can be effected by input messages in order to continue processing intercept record changes in the face of file trouble.

One of these conditions directs the file administration executive to select the available file complex, use it, and not repeat the sequence on the unavailable file. This enables data management to proceed in the event of long-term outage of a file.

The other condition enables the use of an out-of-service file. A benefit of this feature is that a data base can be completely restructured on an out-of-service file. Operating in this manner protects call processing from incurring data errors and protects the restructuring from disturbance by maintenance activity.

V. FUNCTIONS

5.1 Interactive accesses

The most common file administration actions consist of deleting and inserting file entries. Both functions involve use of lookup I/O routines to locate the data block affected, then block transfer to obtain an image in call store, program manipulation of the image, and block transfer to write the revised block at its proper machine address in the disc memory. For a deletion, the revision consists of replacing the entry with blanks—one in place of the called number word, and one in place of an immediately following new number word if present. An insertion must be placed in numerical order. If one or two blanks are not present just before the next greater called number, they must be found elsewhere in the file. The preferred source is within the block already imaged in call store, but if necessary the data are rippled through successive sectors by transfer after transfer until blanks are found. The action is completed in both files before another request is accepted.

In case the action requested is inconsistent with data in file, it is rejected. Entries to be deleted must really be there, and numbers to be inserted must not already be in file. Other checks are made on existing file structure and hardware integrity with every action.

The call-count abstracting function uses a lookup to locate the start of a thousands group. Then the function obtains call store images of block after block of data which it scans for counts below the specified threshold. Other printing functions are also provided for administrative and maintenance purposes, some using lookups on data, some starting from given machine addresses. These print as little as one file entry or as much as all the data in one track of the disc memory. An ancillary function is available to print the machine address of a particular file entry.

Data maintenance functions use only block transfers to operate on one or both file complexes. Match and transfer actions, respectively, verify and produce the correspondence of blocks of data in the duplicated disc memories. Machine-addressed deletion and data writing functions provide for writing blanks or other specified bit patterns into any desired locations.

5.2 Intercept data error checks

Auditing functions check the intercept data for inconsistencies or violations of the file structure and coding scheme. Depending on the mode of initiation, either a single thousands group of numbers, a single track, or one-eighth of the intercept data base may be covered, in one file complex or both. The method of testing is to proceed word by word through the specified area in one file. If both files are to be checked, the corresponding area in the other file is tested by means of block comparisons with valid data in the initial file.

Each data word may be distinguished as to type by its two-bit tag and tested accordingly. The validity check function tests called number words for sequential order and verifies that the two-digit status code is one of an allowed set of values. This status code than serves as an indication of whether the called number should have an associated new number. It also indicates the format and type of information that should be contained in the new number NPA digits which are used by call processing programs to form the locality and NPA segments of the new number announcement.

In addition, tests are made that all telephone numbers consist of valid BCD digits, from 0 to 9 or from 2 to 9 as appropriate. Certain special codes are permitted, such as those used to represent groups of 100 or 1000 called numbers. Where group entries occur, checks are made that no individual entries occur within the range covered by the group entry.

Header and locator words for thousands groups should be in one-to-one correspondence and contain identical information. In addition, the machine address contained in each should point correctly to the location of the header word. Locators are tested for ascending sequential order, although the order of thousands groups on the data tracks need not be sequential.

Blank words are permitted to occur anywhere except between a called number and its associated new number. In addition, checks are made that areas of varying size between thousands groups contain only blank words.

An external initiation of validity checks may request that the tests cover a particular thousands group of numbers, or one-eighth of the data base. In the latter case, each subsequent request causes the next eighth to be tested. In both cases, either a particular file or both files may be tested. The automatic initiation of validity checks at one a.m. each day covers one-eighth of the intercept data in both file complexes so that all the data are audited in eight days. Timed and externally initiated eighths are incremented independently.

Internal indications of possible data anomalies, such as call processing detection of a header not-in-file, a mismatch between files, or a new number missing result in requests for file diagnostics. If all the hardware tests pass in these cases, validity checks are initiated in the area of the possible data error. These checks cover a single track on which the error should be located, and test that track in both file complexes if both are in service.

For all types of validity checks, records are printed of each individual error detected. At the end a summary record is printed including a total count of errors and an indication of the range of data tested. The information in the individual records may be used later to locate the errors and correct them by deletion of incorrect data and reinsertion of correct data. No automatic corrections are attempted on the intercept data because it is difficult or impossible to determine by program what the correct data is.

5.3 Recovery and restructuring

Recovery from a loss of intercept data involving one or both disc memories varies depending on the amount of data lost and whether the same data were lost from both file complexes. When the correct data are still present on one file they can be readily transferred internally via the No. 2 ESS processor to the other file. A maximum of about four hours is required if all the data in a four-disc unit must be transferred. In the unlikely event that both online copies are lost, even in part, then data must be entered externally, either individually using routine update messages or by using appropriate sections of the latest backup copy of the complete data base. If the backup copy is maintained on a spare disc memory unit, the backup unit can be temporarily placed on line allowing internal transfer of the desired data. If backup is in the form of paper tape, then recovery proceeds at about 10,000 entries per hour. Any recovery operation utilizing a backup copy of the data base must be supplemented by relevant update tapes generated since the last complete copy was made.

Should neither file be considered suitable for processing calls, all calls requesting file lookups are diverted to a simple "blank number" announcement. Because such a condition could conceivably persist over the potentially long time intervals involved with data base recovery operations, provisions are made to return a partially recovered file to service. When this is done the system is made temporarily insensitive to the expected flurry of not-in-file indications since their normal handling would cause flooding of the centralized intercept bureau operator positions.[1]

It may be found in normal operation of some AICs that occasionally a region of the file becomes so congested that insertion of new entries requires abnormally long times to complete all the data rippling required to maintain ordering within thousands groups. To relieve such congestion it is possible to do some restructuring of the data base. Since thousands groups are relocatable, restructuring can be done by extracting copies of one or more thousands groups from the congested region, removing their old image, and reinserting them into a less-congested region of the file.*

5.3.1 Intercept data compression techniques

Lengthy data transmissions from external sources are usually required for restructuring, recovering, or expanding the intercept data base. To reduce the time required for such extensive information exchanges, the intercept data are compressed so that fewer bits are required to record and subsequently reconstruct them than are stored on disc. Several characteristics of the intercept data format contribute to the applicability of a variety of data compression techniques.

One such characteristic is the existence of a significant amount of redundancy. In the section describing file organization it was noted that all entries belonging to the same thousands group are stored

* A recently completed automatic blank redistribution now accomplishes the desired restructuring. The FAS provides for a magnetic tape backup copy of the data base and an intermediate-speed recovery operation.

sequentially. In addition to such ordering, each entry includes its thousands group designation as part of the stored information. Since the thousands group designation can be inferred from the position of an entry in the data stream, it is not necessary to record it for each entry.

Another basic characteristic of the data base is that intercept number information is largely stored as BCD digits. The use of a BCD format, while convenient for many of the basic AIS functions, is inherently inefficient as a coding scheme. An alternative coding scheme having higher information content is used for compressing the disc data.

Throughout the data base, some blank disc words are usually interspersed for updating convenience. The exact placement of spare words is not critical, making it possible to omit them during data compression. Blank words may be inserted automatically during subsequent reconstruction if desired.

Still another source of data compression results from certain fields of each entry assuming predetermined values when being reconstructed from compressed code. These fields need not be recorded since they can be automatically initialized by a reconstruction program.

The implementation of encoding and redundancy omission for intercept data compression is described in greater detail in the next two sections.

5.3.2 Encoding techniques

The conventional use of ASCII encodes *each* decimal digit in seven information bits and one optional parity bit. These seven or eight bits are then treated as one character. The coding scheme employed by the AIS data compression program converts each *pair* of BCD digits into a single seven-bit code which need not be equivalent to the decimal value of the BCD pair. This seven-bit code, plus an optional parity bit, is then treated as one character for transmission purposes. Compressed coding thus achieves a "two-for-one" reduction of character transmission compared to conventional use of ASCII coding. Figure 5 illustrates the difference in the codes as they would be used to punch a new number with its area code in paper tape.

A table lookup technique is employed both for converting pairs of BCD digits into single seven-bit characters and for expanding single seven-bit characters back into BCD pairs. The principal advantage of a table technique compared to conventional conversion algorithms is the ease of assigning arbitrary correspondences between BCD pairs

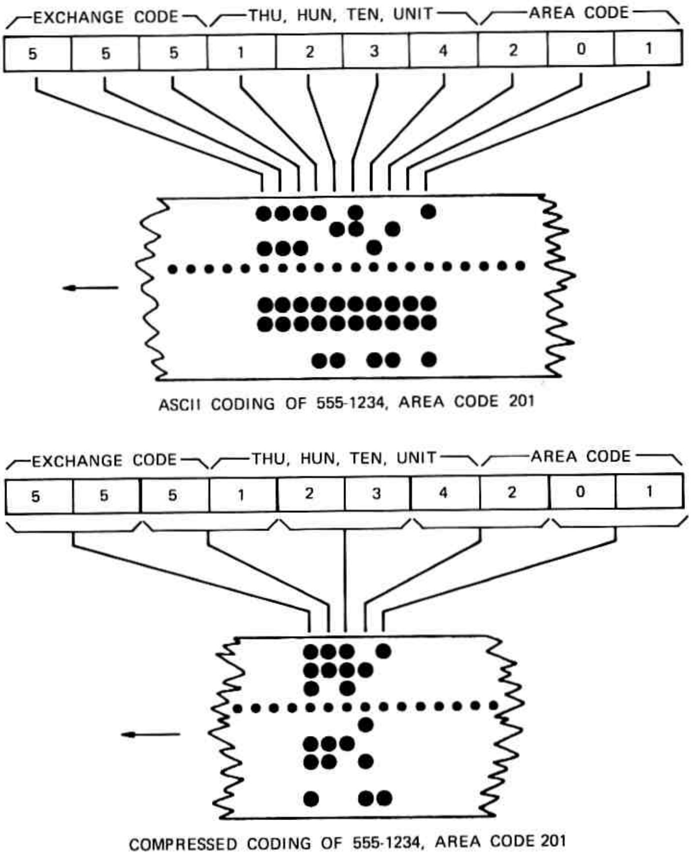


Fig. 5—Compressed code vs ASCII coding.

and seven-bit codes. This feature is particularly useful in avoiding characters which are likely to result in undesired actions by the data processing facilities involved. An occurrence of an EOT character ("end of transmission") in the middle of a compressed code data stream, for example, could result in a premature disconnection since the receipt of an EOT character causes several types of commonly used data terminals to disconnect automatically. This and other troublesome characters indicated by the asterisks in the conversion table in Fig. 6 are avoided by the assignment scheme used.

Directly indexable tables for the expansion and compression functions tend to be relatively large. In AIS some memory saving techniques are employed to incorporate both tables within a single 128-word block of memory. To convert BCD pairs in the range 00–99 a

BCD PAIR	PARITY	BIT 7	BIT 6	BIT 5	BIT 4		BIT 3	BIT 2	BIT 1	SYMBOL	DECIMAL EQUIVALENT
· 00	●	●	●	●	●	•		●		z	122
01	●					•			●	SOH	001
02	●					•		●		STX	002
03						•		●	●	ETX	003
· 04	●	●	●	●		•	●	●		v	118
· 05	●	●	●	●		•	●		●	u	117
06						•	●	●		ACK	006
· 07		●	●	●		•	●			t	116
08	●				●	•				BS	008
09					●	•			●	HT	009
10					●	•		●		LF	010
11	●				●	•		●	●	VT	011
12					●	•	●			FF	012
13	●				●	•	●		●	CR	013
14	●				●	•	●	●		SO	014
· 15		●	●		●	•	●	●	●	o	111
· 16		●	●	●		•	●	●	●	.	119
· 17		●	●	●	●	•		●	●	{	123
· 18	●	●	●	●	●	•			●			121
· 19	●	●	●	●		•		●	●	s	115
· 20		●	●	●		•		●		:	114
21	●			●		•	●		●	MAK	021
22	●			●		•	●	●		SYN	022
23				●		•	●	●	●	ETB	023
24				●	●	•				CAN	024
25	●			●	●	•			●	EM	025
26	●			●	●	•		●		SUB	026
27				●	●	•		●	●	ESC	027
28	●			●	●	•	●			FS	028
29				●	●	•	●		●	GS	029
30				●	●	•	●	●		RS	030
31	●			●	●	•	●	●	●	US	031
32	●		●			•				SP	032
· 33		●	●	●		•			●	q	113
34			●			•		●		"	034
35	●		●			•		●	●	#	035
36			●			•	●			$	036
37	●		●			•	●		●	%	037
· 38	●	●	●	●		•				p	112
39			●			•	●	●	●	'	039
40			●		●	•				(	040
41	●		●		●	•			●	)	041
42	●		●		●	•		●		•	042
43			●		●	•		●	●	+	043
44	●		●		●	•	●			'	044
45			●		●	•	●		●	–	045
46			●		●	•	●	●		.	046
· 47		●	●	●	●	•				x	120
48			●	●		•				0	048
49	●		●	●		•			●	1	049
50	●		●	●		•		●		2	050
51			●	●		•		●	●	3	051
52	●		●	●		•	●			4	052
53			●	●		•	●		●	5	053
54			●	●		•	●	●		6	054
55	●		●	●		•	●	●	●	7	055
56	●		●	●	●	•				8	056
57			●	●	●	•			●	9	057
58			●	●	●	•		●		:	058
59	●		●	●	●	•		●	●	;	059
60			●	●	●	•	●			<	060
61	●		●	●	●	•	●		●	=	061
62	●		●	●	●	•	●	●		>	062
63			●	●	●	•	●	●	●	?	063
64	●	●				•				.	064
65		●				•			●	A	065
66		●				•		●		B	066
67	●	●				•		●	●	C	067
68		●				•	●			D	068
69	●	●				•	●		●	E	069
70	●	●				•	●	●		F	070
71		●				•	●	●	●	G	071
72		●			●	•				H	072
73	●	●			●	•			●	I	073
74	●	●			●	•		●		J	074
75		●			●	•		●	●	K	075
76	●	●			●	•	●			L	076
77		●			●	•	●		●	M	077
78		●			●	•	●	●		N	078
79	●	●			●	•	●	●	●	O	079
80		●		●		•				P	080
81	●	●		●		•			●	Q	081
82	●	●		●		•		●		R	082
83		●		●		•		●	●	S	083
84	●	●		●		•	●			T	084
85		●		●		•	●		●	U	085
86		●		●		•	●	●		V	086
87	●	●		●		•	●	●	●	W	087
88	●	●		●	●	•				X	088
89		●		●	●	•			●	Y	089
90		●		●	●	•		●		Z	090
91	●	●		●	●	•		●	●	[	091
92		●		●	●	•	●			\	092
93	●	●		●	●	•	●		●	]	093
94	●	●		●	●	•	●	●		^	094
95		●		●	●	•	●	●	●	-	095
96		●	●			•					096
97	●	●	●			•			●	a	097
98	●	●	●			•		●		b	098
99		●	●			•		●	●	c	099
A0	●	●	●			•	●			d	100
A1		●	●			•	●		●	e	101
A2		●	●			•	●	●		f	102
A3	●	●	●			•	●	●	●	g	103
A4	●	●	●		●	•				h	104
A5		●	●		●	•			●	i	105
A6		●	●		●	•		●		j	106
A7	●	●	●		●	•		●	●	k	107
A8		●	●		●	•	●			l	108
A9		●	●		●	•	●		●	m	109

(TAPE CODE: PARITY through BIT 1; A0–A9: DELIMITER SERIES)

· INDICATES DEVIATIONS IN CONVERSION OF BCD PAIRS TO DECIMAL EQUIVALENTS.

NOTE: A = DECIMAL 10

Fig. 6—Compressed code conversion table.

directly indexable table of 154 words (nine blocks of 16 words each plus one block of 10 words) would be required. In AIS a delimiter digit equal to decimal ten is used which increases a directly indexable table size by one 16-word block, thus requiring 170 entries. The table size for expanding any seven-bit binary character to some BCD pair need only be 2^7 or 128 entries at most. In AIS the memory for storing tables consists of 22-bit words. This permits both the expansion and compression tables to share the same words by being assigned different bytes in each word. The combined table has the expansion table occupying the low-order eight bits and the compression table occupying the next higher seven bits. Just using shared memory alone would allow this combined table to fit into a single block of 170 words of memory. A further significant reduction in memory space for such a table is possible by using an "overlay" technique. The specific "overlay" structure for AIS is described below.

Whenever a BCD pair to be compressed is equal to or greater than BCD "80" (corresponding to the 128th entry), BCD "30" is subtracted and the compression table is indexed with the remainder. If the seven-bit binary number obtained from the table is added to the binary equivalent of decimal 30 the result will be a seven-bit binary equivalent for the original BCD pair. This kind of "overlay" is only possible if that portion of the table to be "overlaid" contains binary equivalents for the BCD ranges involved that differ only by a fixed constant. This happens to be the case for the conversion table in Fig. 6 since the portion of the table shared is BCD "50" to BCD "79" and all exceptions to straightforward decimal conversion lie below BCD "50." The fixed constant difference as noted earlier is the binary equivalent of 30 decimal. With the "overlay" technique the "two-for-one" conversion table can now be accommodated by a block of 128 words of memory.

5.3.3 Omitting redundancy

In addition to using the "two-for-one" coding scheme, the AIS achieves significant data reduction by omitting from compressed data certain types of the redundancy provided in the data base for ease of access. As noted previously, of the ten possible BCD digit fields for each disc word only seven digits are used to store the intercepted directory number. Of these seven BCD digits, four digits are used to designate the thousands group and the other three give the hundreds, tens, and units digits. The major source of redundancy is the repetition of the four digits designating the thousands group for each called number entry to be intercepted. Since all entries for a particular thousands group are placed sequentially on the file, the thousands group information need only be recorded once for each group.

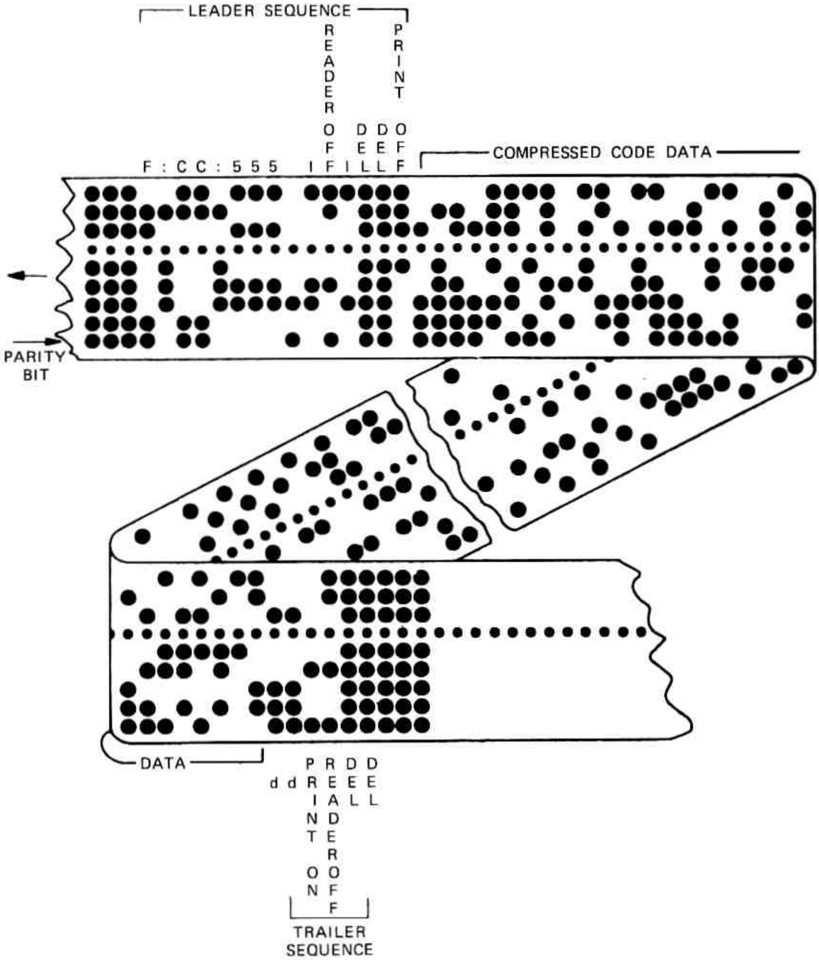


Fig. 7—Typical compressed code tape.

The call count and tag bits which account for six of the remaining useful bits need not be explicitly transmitted since the tag bits can be implied from entry type and the call count bits are initialized to a fixed value at time of reconstruction. It is thus found that, at most, only five BCD digits are required to record or reconstruct each called number entry. These are the hundreds, tens, and units, and two status digits. Sometimes an intercepted number has an associated new number reference. In this case the new number entry will always be found in the next consecutive disc word. The new number entry is treated as ten BCD digits consisting of a seven-digit directory number plus possibly either an area code or some special announcement code. Currently all ten digits for a new number entry are transmitted and recorded since a high degree of randomness is found for this type entry. The tag bits, however, need not be transmitted since they are implied by the entry type.

When full advantage is taken of all of the potential savings it is possible to reduce transmission times by almost an order of magnitude compared to methods using routine update facilities. In addition to reducing transmission time, the amount of storage media required to record all or part of the data base externally is also reduced. These savings benefit not only AIS, but also telephone company computer centers whenever such centers become involved in data base generation, backup storage, or routine transmissions to and from associated AICs.

At present for the AIS the storage medium for compressed data is paper tape. A typical compressed code paper tape consisting of a leader, data, and trailer is illustrated in Fig. 7. Teletypewriter page copy normally used to monitor paper tape transmissions would not be available, since to a teletypewriter compressed code appears as a random stream of both printing characters and nonprinting control characters. Except for the leader at the beginning of each thousands group and a summary message giving called and new number totals at the end of each thousands group, a teletypewriter page printer would normally be suppressed to avoid erratic printer operation. For AIS the loss of page copy is considered an acceptable trade-off in exchange for the overall time and storage media savings, particularly since an end result check of the reconstructed data can be made with audit programs and selective printouts of individual or group entries.

VI. SUMMARY

The AIS file administration programs provide both the external interactive human interface and the internal machine interface for managing the intercept information stored on duplicate disc files. Routine file administration activities have minimal effect on call processing as long as one good copy of the intercept data is available.

To aid in maintaining valid matched data on duplicate files, both automatic and externally initiated auditing functions are provided. Portions of the data base may be displayed in a variety of formats under the control of human requests.

Some special-purpose functions have been included for infrequent recovery and restructuring actions. The speed of both the special-purpose functions and the routine update functions is expected to improve when a data link interface to remote data processing equipment becomes available in the near future.

VII. ACKNOWLEDGMENTS

The authors wish to acknowledge contributions from many others associated with the AIS project, especially Mrs. T. C. Means for file administration teletypewriter programs and the late R. A. Sussman for initial development of the compressed code conversion tables.

REFERENCES

1. C. J. Byrne, W. A. Winckelmann, and R. M. Wolfe, "Automatic Intercept System: Organization and Objectives," B.S.T.J., this issue, pp. 1–18.
2. J. W. Hopkins, P. D. Hunter, R. E. Machol, J. J. DiSalvo, and R. J. Piereth, "Automatic Intercept System: File Subsystem," B.S.T.J., this issue, pp. 107–132.
3. T. E. Browne, T. M. Quinn, W. N. Toy, and J. E. Yates, "[No. 2 ESS] Control Unit System," B.S.T.J., *48*, No. 8 (October 1969), pp. 2619–2668.
4. H. Cohen, D. E. Confalone, B. D. Wagner, and W. W. Wood, "Automatic Intercept System: Operational Programs," B.S.T.J., this issue, pp. 19–69.

Copyright © 1974 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 53, No. 1, January 1974
Printed in U.S.A.

Automatic Intercept System:

Development Support Systems

By I. D. BUCK, G. D. CRUDUP, C. W. KEMP, and G. C. VOGEL

(Manuscript received September 29, 1971)

A series of hardware-software packages has been developed to aid the Automatic Intercept System (AIS) development. These packages involve a PDP-9 support computer connected to the AIS control unit through specially designed hardware. Described in this paper are a real-time debugging system, a series of three support computer programs which aid in the administration of AIS program development, and two utility programs used in connection with the AIS file subsystem.*

I. INTRODUCTION

Development of a real-time system such as the Automatic Intercept System[1] requires many support programs. The assembler, loader, and simulator used by AIS were developed by the No. 2 ESS Switching System development.[2] In addition, a real-time debugging system called "ORACLE" and a series of utility and program administration programs have been developed and are the subject of this paper.

1.1 Real-time debugging

A debugging aid for real-time program development should provide all the features of nonreal-time debugging aids but in addition be noninterfering with the system operation. The objectives of giving all the features of normal debugging aids and being noninterfering are difficult to meet since most debugging routines suspend program execution to obtain data concerning the state of the program.

* PDP is a trademark of Digital Equipment Corp., Maynard, Mass.

ORACLE, the real-time debugging system, uses a combination of software, special interface hardware, and a PDP-9 computer to provide a noninterfering debugging package. ORACLE uses the PDP-9 [3] support computer to monitor, collect data, and print the results of AIS operations. Basically, ORACLE provides three things:

(*i*) Records of the "from" and "to" addresses of branch (transfer) instructions executed by the AIS control unit.
(*ii*) Dumps of the contents of AIS temporary memory (call store).
(*iii*) Means by which the printing of the above data can be activated.

1.2 Program administration

Program administration is the process of controlling, documenting, and verifying the correctness of a total system program. In the AIS development, programmers produce separate programs which are assembled on an IBM 360. The results of many such assemblies are linked together by a loader program to form an image of the AIS program store on magnetic tape. This image must be transferred onto "magnet cards" which form the AIS memory.[4] AIS uses the PMT (permanent magnet twistor) memory. In this memory, information is stored in small bar magnets attached to an aluminum card. It is the use of this type of memory in AIS, and the lack of high-speed input such as magnetic tape or high-speed paper tape, that has prompted the development of the three program administration programs (MAGNUS, PSUTY, and OVRWRT).

MAGNUS reads AIS program store data from tape and controls an MCW (memory card writer)[5] which records the data on PMT memory cards. PSUTY (*P*rogram *S*tore *UT*ilit*Y*) is used to compare magnetic tape images of the AIS program store with the actual contents of the store. It can also generate tape images from the actual program store contents. OVRWRT (*OV*e*RWR*i*T*e) is an assembler which produces program patches to the AIS program. It allows the AIS programmers to rapidly produce changes for fixing program bugs. OVRWRT translates assembly language statements into machine language, overlays the assembled code on top of what already is in the AIS program store, and produces new PMT cards via the MCW. Each of these programs will be described in detail in later sections.

1.3 Disc backup and system parameter initialization

Two programs, DISKUS and PDATA, have been developed to back up the AIS file data base and to change the parameter data stored on that file.[6,7]

1.4 AIS—PDP-9 hardware interfaces

Figure 1 shows the three hardware interface circuits used by the PDP-9 development support programs. The real-time debugging interface provides the means by which AIS program execution is monitored. It provides a path for passing the contents of AIS call store into the PDP-9's memory. The asynchronous communication register, connected between the PDP-9 I/O bus and the AIS scan answer bus, provides a path from the PDP-9 to the AIS. Over this path both commands and data are passed to an AIS program called CPTSTA. The memory card writer interface connects the PDP-9 to the memory card writer and is used by MAGNUS and OVRWRT.

II. REAL-TIME DEBUGGING

Debugging of a system such as AIS is more difficult than that involved with nonreal-time systems for several reasons:

- (*i*) Real-time operation.
- (*ii*) Random input traffic.
- (*iii*) Added reliability requirements that make system malfunctions an added input rather than a reason for stopping the system.
- (*iv*) Multiprogramming and multiprocessing.
- (*v*) Large, complex programs.

These reasons indicate a need for a debugging aid which monitors the program execution in the AIS without any disturbance of the AIS. ORACLE is such a debugging aid. Within the support processor's memory a complete image of the AIS call store and a trace of transfers that occur in AIS are recorded. The transfer trace contains up to 128 "from-to" address pairs. Match circuits monitor the program address, call store address, and call store information to control the collecting and printing of call store images and transfer trace data.

2.1 Hardware description

The real-time debugging system uses a wired logic interface controlled by support computer programs to make logical decisions

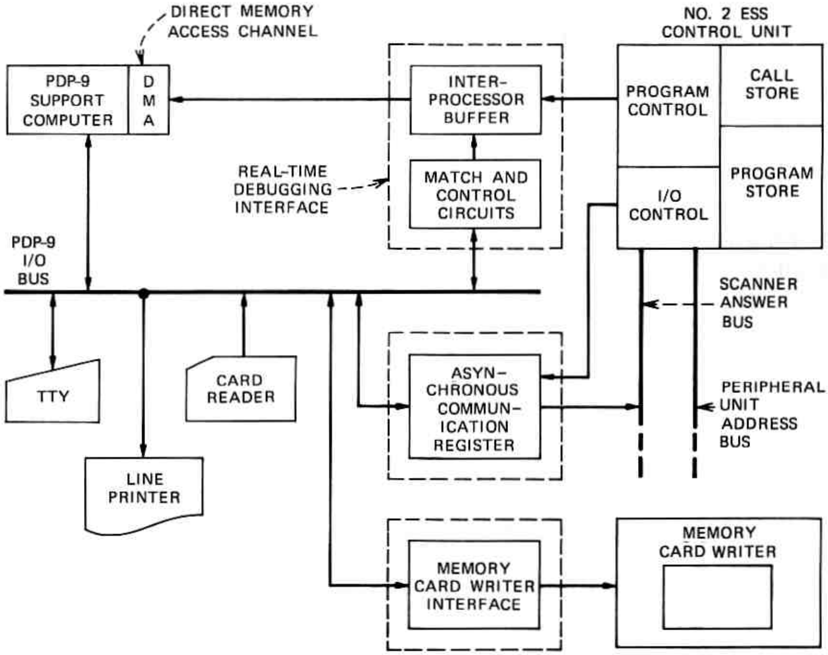


Fig. 1—Block diagram of the AIS development support system.

based upon data inputs from AIS and to control the flow of data from the AIS to PDP-9 memory. Results of the wired logic decisions initiate program action to tabulate and print the desired output data from the PDP-9's core memory.

A block diagram of the real-time debugging hardware is shown on Fig. 2. Three sets of data leads are utilized from the AIS processor. They are the PA (program store address), CSA (call store address), and CSI (call store input) data. The PA leads address the PMT memory and are used for records of transfer instructions executed by the AIS control unit. The CSA and CSI leads provide an image of the AIS call store in PDP-9 core for call store dumps. Eight matcher circuits (M0–M7) and three registers, CR (control register), FR (function register), and SR (status register), form the logic which controls data flow from the AIS to the support computer through a DMA (direct memory access) channel.

2.1.1 Matcher setup

A user inputs requests to the utility system via the support processor TTY or card reader. ORACLE sets up these requests via IOT

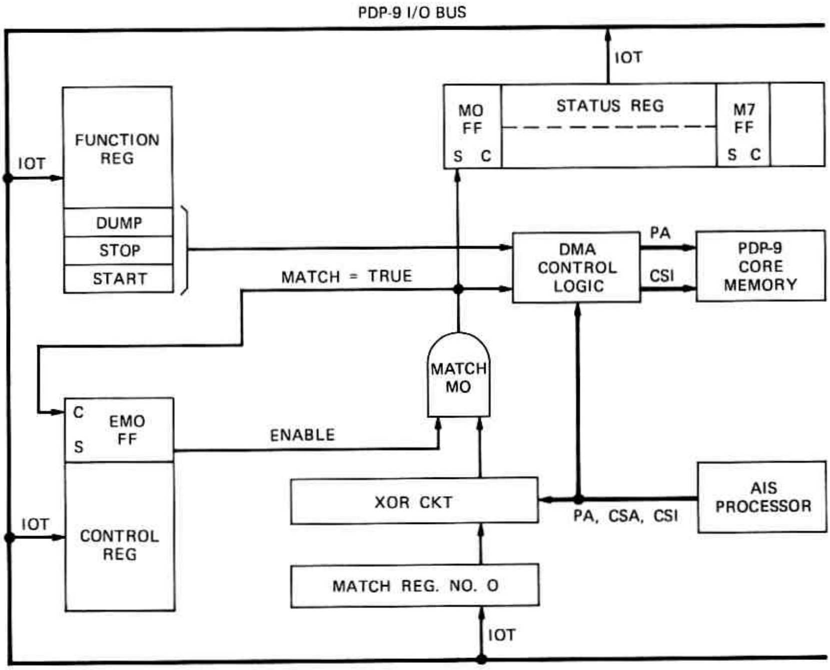


Fig. 2—Real-time debugging system: hardware interface between AIS and the PDP-9.

(input/output transfer) instructions to the match register and control registers in the utility hardware. When a match register is enabled, its contents are continuously matched against the PA leads by an exclusive "OR" circuit. If a match occurs, a bit corresponding to the matcher number is set in the SR, the match circuit is disabled by clearing its associated enable bit in the CR, and action is taken by the DMA control logic to start or stop the data flow to the support computer's memory.

2.1.2 Registers

The CR contains two bits per match circuit to enable each matcher circuit either conditionally or unconditionally. When a match circuit fires, its enable bit is cleared to prevent any further action by that matcher. The FR contains three bits per match circuit. These three bits determine what action will be taken by the DMA control logic, on the data flow, when a matcher fires. The "start" bit starts the flow of transfer addresses to be stored in the PDP-9. The "stop" bit stops the storing of transfer addresses. "Dump" stops the data flow on the CSI leads which provides a call store image. The SR contains one bit for each matcher. These bits are set when the corresponding matcher "fires." Matcher firing also causes a program interrupt in the support computer to signal ORACLE that some action should be taken. The status register may be read by the program via an IOT instruction.

2.1.3 Direct memory access control

The DMA control logic derives all the control signals for strobing matchers and controls the storing of transfer trace and call store data in memory. Decision logic is located here to start and stop data flow dependent upon a matcher firing and the contents of the FR bits for that matcher.

2.1.4 Special match functions

Matchers 0–6 compare the data loaded in their match register against the PA leads from the AIS control unit. Match circuit 0 (M0) is unique in that it may be set up by program to conditionally enable one or more of the remaining match circuits. Until M0 fires, the conditionally enabled matchers are disabled. Matcher 7 compares its data against the CSA leads. This matcher provides all the functions of matchers 0–6, and in addition performs a bit matching function. Matcher 7 is divided into three major hardware sections:

(*i*) CSA match register,
(*ii*) CSI match register, and
(*iii*) CSI mask register.

Two modes of operation may be used with this configuration. If the CSA match register is loaded with an address and the mask register is zero, then the matcher acts as a CSA matcher. If the CSA and CSI are loaded with information and the mask register is set to all ones, the match looks for an exact match on the CSA and all the bits of the CSI. A zero mask bit is a "don't care." With this circuitry, the reading or writing of specified data at a given call store address can initiate utility functions.

2.2 Control program

2.2.1 General organization

Figure 3 is a functional diagram of the real-time debugging software. Primary control is from the PDP-9 console TTY with data coming from the card reader. An input deck, shown in the upper left of Fig. 3, contains a description of what data are to be collected and when they are to be printed out. The input program has responsibility for checking the syntax of the UCL (utility control language) statements on each input card and translating them into a sequence of numerical codes in the table LCTAB. At the end of each card the input program passes control to the matcher setup program which checks for consistency between all cards processed so far, builds an image of the hardware registers, and stores data concerning each matcher in the MCB (matcher control block) table. An end of file in the input deck causes the matcher setup program to load the hardware registers and turn the hardware on with a final "GO" command. The "firing" of one or more of the matchers causes a PDP-9 program interrupt, passing control to the utility function execution program which collects hardware status information from the interface circuits and causes data to be printed out based on the number of the matcher that fired and the contents of the corresponding MCB. The overall operation is explained by the following example.

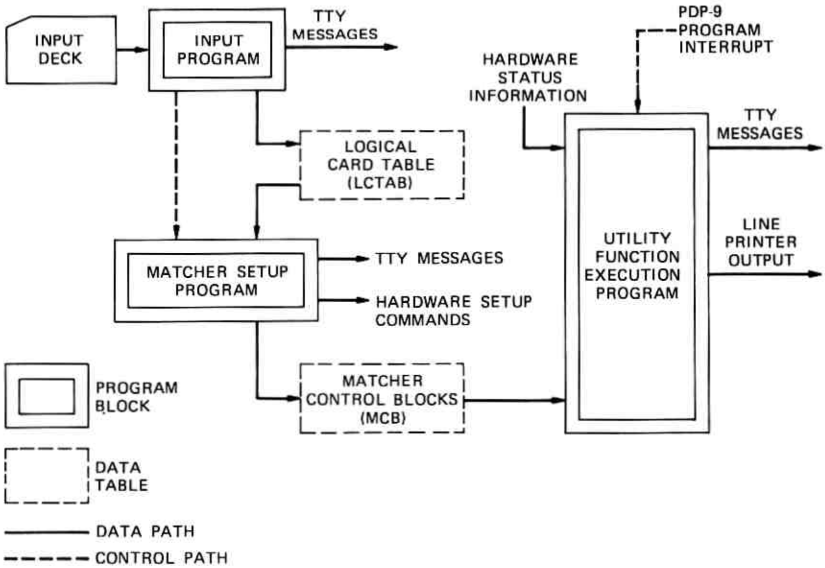


Fig. 3—Real-time debugging system: functional software diagram.

2.2.2 Example

Consider the input UCL statement

```
//START MATCH M0, PA=30000, STARTT, ENAB(M1, M7)
//SYMB. A MATCH M1,PA=17,DUMP (250, 500)
//SYMB. B MATCH M7,BIT(360, 40),ENDTT
/*
```

The first statement says "set matcher M0 to 30000 and, when a match occurs, start the transfer trace hardware and enable matcher M1 and M7." Because of the ENAB(M1, M7) function on M0, matchers M1 and M7 will not be "turned on" until after matcher M0 has "fired." Statement number 2 specifies that matcher M1 be set up for a program address match at 17 and when the match occurs, a dump of the AIS call store image between 250 and 500 will be printed. Matcher 7 specifies a match when 40 is written into or read from location 360. When M7 "fires" the transfer trace data are printed due to the ENDTT (*END T*ransfer *T*race) function. The /* terminates the matcher setup phase of the debugging system.

Figure 4 shows the output that results from the sample input statements. Lines 1–4 are the input statements. Line 5 indicates that the matcher with the symbol START has fired and the SEQ# = 01 says that it was the first matcher to fire. At line 6 we see that the matcher labeled SYMB. B has fired and that a transfer trace will follow. The PA field indicates the program address which caused the bit match to occur and the sequence number indicates that this was the second matcher to fire. Lines 7–24 are the transfer trace data. Line 25 states that the matcher labeled SYMB. A has fired and that a call store dump will follow. Line 37 shows that groups of all zero words are compressed into a single line.

III. PROGRAM ADMINISTRATION AND FILE UTILITY SYSTEM

The program administration and file utility system comprises software and hardware to provide such functions as changing AIS file parameters, comparing AIS PMT store contents with store images on magnetic tape, and the actual writing of PMT cards from magnetic tape via the memory card writer.

3.1 Hardware description

The program administration hardware is shown on Fig. 5. Two registers are necessary for the functions described. The MCW output register passes information from the PDP-9 to the MCW upon request to magnetize AIS PMT cards. The ACR (asynchronous com-

```
LINE 1.... //START MATCH M0,PA=30000,STARTT,ENAB(M1,M7)
           //SYMB.A MATCH M1,PA=17,DUMP(250,500)
           //SYMB.B MATCH M7,BIT(360,40),ENDTT
LINE 4.... /*

LINE 5....    START                  SEQ#= 01

LINE 6....    SYMB.B TRA TRACE  PA= 030240 SEQ#= 02
                  FROM ADDRESS
                        TO ADDRESS
LINE 7.... 030001 033764
           033766 033770
           033772 030002
           030005 030027
           030031 030033
           030035 030101
           030101 033653
           033666 030102
           030103 030117
           030121 030160
           030160 030161
           030172 030202
           030214 032556
           032566 032561
           032565 032576
           032577 030215
           030215 030217
LINE 24... 030240 032421

LINE 25...    SYMB.A CS DUMP    SEQ#= 03
                  CALL STORE ADDRESS X
                                          DATA FOR CALL STORE
                                          ADDRESS X THROUGH X + 7
LINE 26... 000250 045636 000400 000000 000000 000000 000000 000000 000000
           000260 045636 000400 000000 010710 000000 000000 000000 000000
           000270 136573 000000 001200 010710 000323 005102 000000 005317
           000300 077731 700130 001200 011262 000323 005116 000000 000717
           000310 045636 000400 001200 000000 000323 005116 000002 005317
           000320 045636 000000 000002 010710 070005 001414 070005 001622
           000330 045636 000400 001230 010515 000323 005102 000000 005317
           000340 136773 000000 001200 010370 000100 000400 000002 005317
           000350 064363 000100 014675 010370 000001 000400 000000 000007
           000360 104430 000030 014536 010373 000000 007500 000221 000520
           000370 000000 000000 006345 004070 000020 000400 000000 000460
LINE 37...   000400  -  000467  =  0
           000470 000000 000000 000000 000000 000000 000000 000000 177777
LINE 39...   000500  -  000507  =  0
```

Fig. 4—Sample output of the real-time debugging system.

munication register) provides a data link from the support computer to the AIS processor.

3.1.1 Asynchronous communication register

The ACR is connected to the AIS scan answer bus and is loaded from the PDP-9 I/O bus by an IOT instruction. A program (CPTSTA) running in the AIS processor scans the ACR for both instructions and data. If control bits ENV and ASW both equal 1, then the 16 data bits are passed over the scan answer bus to the control unit. CPTSTA issues a data acknowledge to clear the ENV and ASW

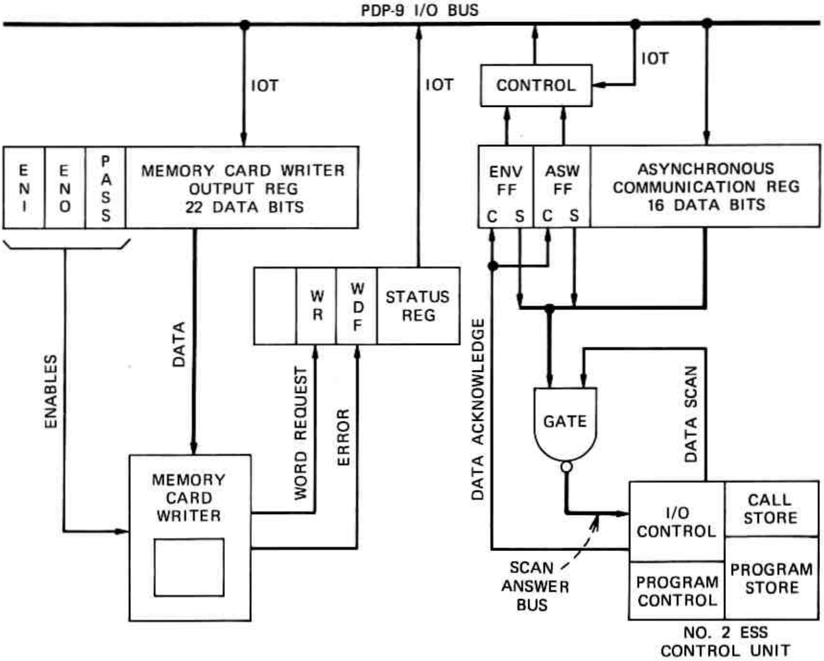


Fig. 5—Asynchronous communication register and memory card writer interfaces.

flip-flops. This indicates to the PDP-9 program that the transfer is complete and new data may be sent.

3.1.2 *Memory card writer*

The MCW is an electromechanical device capable of magnetizing PMT cards for AIS memories. One PMT card consists of 64 rows of small bar magnet with 44 magnets per row. A write head is driven over the PMT card, writing one row at a time. An internal 44-bit register stores the information for each row. As the write head approaches each row of bar magnets, a word request is sent to the SR which in turn creates an interrupt to the control program (MAGNUS). MAGNUS responds with a 44-bit data message via the MCW output register. The process continues until all 64 rows are written on the PMT card.

3.1.3 *MCW output register*

Upon receiving a word request from the SR, the MCW output register is loaded with its data and enables. The 44-bit data register in the MCW is loaded by two paralled 22-bit transfers from the MCW output register. Two enables, EN0 and EN1, are used to steer each 22-bit word transfer into the proper position in the MCW input register.

If either a mechanical or electrical malfunction occurs during the writing of a PMT card, a WDF (word delivery failure) is sent to the SR. A program interrupt signals the support computer program that an error has occurred.

3.2 Program store utility program

PSUTY (*P*rogram *S*tore *UT*ilit*Y*) program uses the AIS—PDP-9 interface to match AIS program store contents with program store images on magnetic tape and to create program store images on tape. PSUTY determines which function to perform from the user's TTY messages. On match functions between tape and the AIS program store, mismatches are listed on the line printer. The tape creation and tape copy functions indicate their completion via a TTY message.

PSUTY communicates with the AIS via the ACR and the DMA channel. The AIS program CPTSTA contains the AIS software necessary to interface with PSUTY. Requests for the contents of a program store card are sent to the AIS via the ACR. CPTSTA reads the contents of the specified plane into the AIS call store. As these data are written in call store, they are also written into the PDP-9's memory via the DMA channel. When this operation is complete, CPTSTA clears the ENV and ASW bits in the ACR which tells PSUTY to begin processing the requested program store data. This process is repeated until all AIS program store data have been processed.

3.3 PMT card magnetization

MAGNUS magnetizes AIS PMT cards from program store images on magnetic tape. MAGNUS is a conversational mode program in which the user can request specific store modules or single planes for magnetization. When the specified data have been found on the magnetic tape input file, the program arranges the data for output and types a message on the TTY. The user may then magnetize the PMT card(s). Error routines are included in MAGNUS which allow automatic restart in the case of a word delivery failure.

MAGNUS communicates with the MCW via the memory card writer interface. MAGNUS reads enough data from magnetic tape to magnetize one PMT card. When the MCW is ready to start magnetizing the card it sends a WR (word request) signal through the interface circuit to the status register. MAGNUS answers the WR by sending 44 bits of data through the MCW output register. The MCW uses the 44 data bits to magnetize one row of magnets on the PMT card. A series of 64 such signals and answers occurs to magnetize one plane.

3.4 Laboratory change assembler

The laboratory change assembler, OVRWRT, is used to assemble and insert program changes into the AIS program store. OVRWRT (Fig. 6) is a conventional two-pass assembler with a magnetizing routine added. Eventually laboratory program changes must be incorporated into the program source files using the EDITOR program and reassembled by the SWAP assembler.[2] Therefore, OVRWRT has been designed to accept both assembly language statements which are compatible with SWAP and control statements required by the EDITOR program.

3.4.1 Special pseudo-operations

Since a program change in one part of a program may use information such as symbol definitions derived from other parts of the program, it is necessary to provide special pseudo-operations for supplying this information. Special pseudo-operations begin with a percent sign (%). The most basic of the special pseudo-operations is %SET. It is generally used to define symbols for use in a program

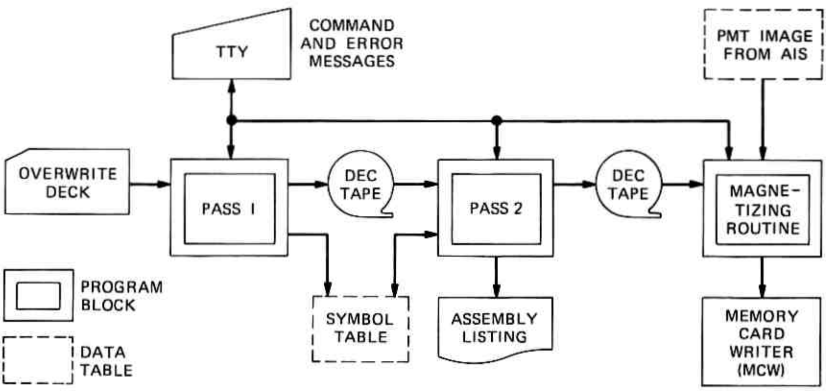


Fig. 6—Laboratory change assembler: functional software diagram.

patch which otherwise would be undefined. For example, the following statement would assign the value 12345 to the symbol TAG.

```
TAG          %SET          12345
```

Symbols may be redefined by %SET. The corresponding regular pseudo-operations (no percent sign) may also be used. However, special pseudo-operations are ignored by the EDITOR program while the regular ones are not. This scheme automatically eliminates inserting unnecessary statements in the user's source program.

3.4.2 System macros for program patching

Program changes requiring the addition or removal of instructions rather than substitution are generally made using system macros. The use of system macros facilitates the subsequent job of editing the program source files. Three system macros, PATCH, TAKEOUT, and WIPEOUT, are used for program patching. The PATCH macro is used for program changes requiring additional instructions to be added in the middle of the program. This is accomplished by overwriting a full word in the program with a transfer (branch instruction) to an unused area in program store. The overwritten instruction as well as the instructions to be added may then be written in the unused area followed by a transfer back to the main body of the program. The following lines show an example of the patch macro:

```
NAME        PCHORG        5000
NAME        PATCH 1
KH          ZGL
            GRXLR
            HLR
            GGR
NAME        PCHEND
```

Expansion of this sample can be found in the assembly listing of Fig. 7. TAKEOUT and WIPEOUT are degenerate forms of the PATCH macro and are used to remove blocks of instruction. TAKEOUT overlays the unwanted instructions with NOP (no operation) instructions while WIPEOUT inserts code which will cause an error condition if an attempt is made to execute it.

3.4.3 The assembly listing

Figure 7 shows an assembly listing produced by OVRWRT. Since input statements are free-field format, the fields are aligned in the

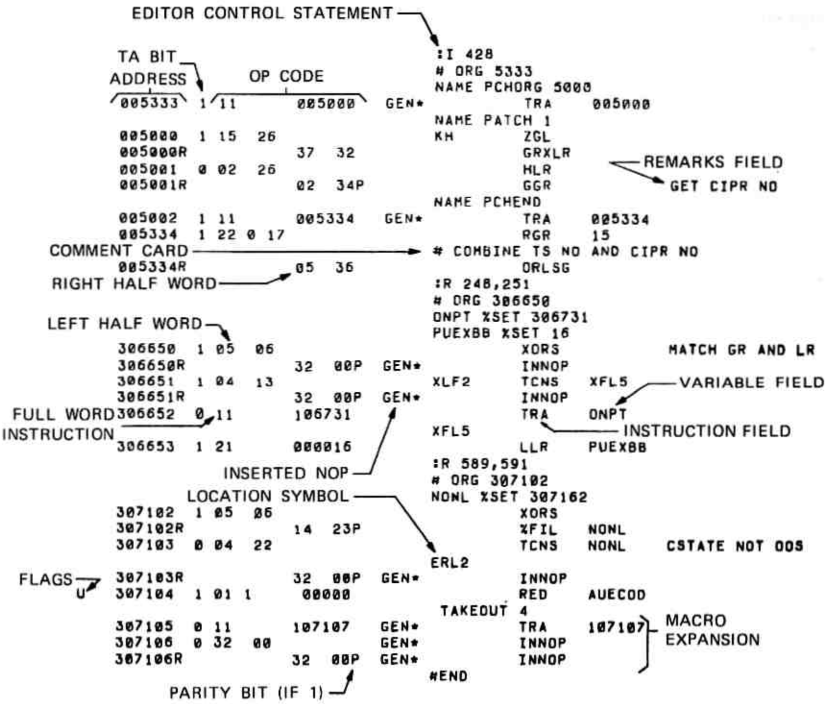


Fig. 7—Example program listing produced by OVRWRT.

listing for greater readability. Assembly errors are indicated by flagging the appropriate statement. Machine language code is presented in one of several field octal formats depending on the instruction classification. Instructions generated by OVRWRT are labeled as such and are listed with both the machine language and assembly language representations. Comments are printed as they appear on the input statements. When a program change has been inserted, the assembly listing is used for program documentation.

3.5 Disc utility programs

The AIS disc utility programs (DISKUS and PDATA) use the AIS—PDP-9 interface circuits to record, restore, and match the contents of the AIS disc file to disc file images on magnetic tapes.[6] These two programs are development tools. DISKUS is used to create backup copies of the AIS intercept number data base. PDATA was developed to record and change the AIS parameter data which are recorded on the AIS disc. In the testing of AIS programs it is necessary to change the configuration of the AIS laboratory model and to change the parameter data accordingly.

3.5.1 Reading and writing AIS discs

Communication between the PDP-9 and the AIS disc proceeds via the DMA channel and the ACR. To initiate a transfer of a block of AIS disc data to the support computer, the ACR is loaded with a function code. When the flag bits ENV and ASW are zeroed, the ACR is loaded with the desired disc address. The AIS program (CPTSTA) locates the disc data and transfers them through the DMA channel into the PDP-9's memory. Communication from the PDP-9 to the AIS disc is established by loading the ACR with another function code. When the code is accepted, the ACR is loaded with the disc address, and then, sequentially, with the 80 data words to be written in the specified disc block.

3.5.2 Disc backup and loading

DISKUS provides AIS disc file backup and restoration of the file data. A complete disc image is stored on a set of 16 low-density magnetic tapes. The user has a choice of three operations: copying the disc contents onto the tapes, restoring the disc from the tapes, or matching the contents of the disc against the tapes. Each of the 16 tapes is assigned a specific portion of the disc. The user has the option of using any one or all of the tapes for each of the three operations.

3.5.3 System parameter initialization

System parameter initialization on the AIS disc is accomplished by PDATA. A maximum of 16 different sets of AIS system parameters may be recorded on one DECtape. PDATA users have the option of: copying current system parameters on tape, changing AIS configuration via a previously recorded set of system parameters, or matching current system parameters to a previously recorded set. Each set of system parameters recorded is accompanied by a description record and a unique name. System parameters occupy a fixed location on the AIS disc.

IV. SUMMARY

Each of the systems described has had a profound impact on the AIS development project. The real-time debugging system and the laboratory change assembler were at first met with skepticism as to the need for such systems, or with at least a wait-and-see attitude. After the systems were introduced, the AIS programmers became heavily dependent on ORACLE and OVRWRT. If the support computer is down because of a hardware failure, the programmers would rather wait for it to be fixed than return to the debugging and program patching methods previously used.

PSUTY and MAGNUS have greatly helped the program administration of AIS. PSUTY reduced the time required to compare AIS program load images from the IBM 360 to the actual program in the AIS machine from several hours to a matter of minutes. The authors strongly feel that every large real-time system should have a series of utility systems such as the ones described and that no proposal for a real-time system development is complete without an outline of what development support systems will be needed.

REFERENCES

1. C. J. Byrne, W. A. Winckelmann, and R. M. Wolfe, "Automatic Intercept System: Organization and Objectives," B.S.T.J., this issue, pp. 1–18.
2. M. E. Barton, N. M. Haller, and G. W. Ricker, "[No. 2 ESS] Service Programs," B.S.T.J., *48*, No. 8 (October 1969), pp. 2865–2896.
3. *PDP-9 User Handbook*, Maynard, Mass., Digital Equipment Corp.
4. T. E. Browne, T. M. Quinn, W. N. Toy, and J. E. Yates, "[No. 2 ESS] Control Unit System," B.S.T.J., *48*, No. 8 (October 1969), pp. 2619–2668.
5. C. F. Ault, L. E. Gallaher, T. S. Greenwood, and D. C. Koehler, "No. 1 ESS Program Store," B.S.T.J., *43*, No. 5 (September 1964), pp. 2097–2146.
6. J. W. Hopkins, P. D. Hunter, R. E. Machol, J. J. DiSalvo, and R. J. Piereth, "Automatic Intercept System: File Subsystem," B.S.T.J., this issue, pp. 107–132.
7. H. Cohen, D. E. Confalone, B. D. Wagner, and W. W. Wood, "Automatic Intercept System: Operational Programs," B.S.T.J., this issue, pp. 19–69.

Contributors to This Issue

Philip J. Brendel, B.E.E., 1961, Manhattan College; S.M.(E.E.) and E.E., 1963, Massachusetts Institute of Technology; Project Officer, U. S. Air Force, Rome Air Development Center, 1963–1966; Bell Laboratories, 1966—. Mr. Brendel initially worked on the Traffic Service Position System where he developed maintenance programs for peripheral units. While he was working on AIS, he was involved in the development of trunk and service circuit maintenance programs. He then worked on the Switching Control Center project, in particular on the centralization of the maintenance of electromechanical switching systems. Mr. Brendel is currently in the Switching Administration and Maintenance Systems Center and is involved in the applications of minicomputers in central office maintenance functions.

Irwin D. Buck, A.A.S., 1958, Hudson Valley Technical Institute, Troy, New York; Bell Laboratories, 1958—. Mr. Buck has worked on the development of the No. 101 Electronic Switching System, No. 2 Electronic Switching System, and Automatic Intercept System in the logical design group. He assisted in the installation of the first No. 101 Electronic Switching System in Cocoa Beach, Florida; No. 2 Electronic Switching System in Iron Mountain, Michigan; and Automatic Intercept System in Hempstead, Long Island. He is presently involved in the design and development of the EADAS system.

Charles J. Byrne, B.S.E.E., 1957, Rensselaer Polytechnic Institute; M.S.E.E., 1958, California Institute of Technology; Bell Laboratories, 1958–1963; Bellcomm, 1963–1968; Bell Laboratories, 1968—. In his earlier association with Bell Laboratories, Mr. Byrne was engaged in research in digital communication systems, particularly in the synchronization of digital networks. At Bellcomm, he was responsible for system studies of lunar reconnaissance spacecraft in support of the Apollo Program. Currently, he supervises a group developing a minicomputer system which collects billing data from local switching offices.

John H. Carran, B.S.E.E., 1959, Ohio University; M.E.E., 1961, New York University; Ph.D., 1969, Stevens Institute of Technology;

Bell Laboratories, 1959—. Mr. Carran has done development work on various electronic switching systems including No. 1 ESS and the Automatic Intercept System. Currently he is a supervisor engaged in system development of a Centralized Maintenance Facility for both electromechanical and electronic switching systems. Member, IEEE, American Physical Society, Tau Beta Pi, Phi Kappa Phi.

Harvey Cohen, B.A. (Mathematics), 1967, Northeastern University; M.S. (Applied Mathematics), 1970, New York University; Bell Laboratories, 1968—. Mr. Cohen worked initially on developing the AIS installation testing programs and on the development of the AIS audit programs. He is currently working on a modern traffic data collection and centralized real-time network management system. Member, Phi Kappa Phi.

William K. Comella, B.S.E.E., 1963, University of Colorado; M.S.E.E., 1965, Columbia University; Bell Laboratories, 1963—. Mr. Comella worked on the design of circuits for PBX–Automatic Identified Outward Dialing and then maintenance programming for the Traffic Service Position System. He is supervisor of the group responsible for network and trunk circuits and maintenance programs for the Automatic Intercept System.

Daniel E. Confalone, B.S. (Mathematics), 1959, University of Rhode Island; M.S.E.E., 1961, New York University; Bell Laboratories, 1959—. Mr. Confalone worked initially on machine aids to design. He later worked on the design of switchboards and testboards. Since 1963 he has worked on the development of the AIS and has been involved with call processing programs since 1967. Member, Phi Kappa Phi, Tau Beta Pi.

Greta D. Crudup, Bell Laboratories, 1969—. Ms. Crudup has worked on the Traffic Service Position System and the Automatic Intercept System. With the Automatic Intercept System, her work is primarily with development utility programs.

J. J. DiSalvo, Electronic Technology, 1950, RCA Institutes; Bell Laboratories, 1951—. Since joining Bell Laboratories, Mr. DiSalvo has been involved in development work on No. 5 Crossbar, Step-by-Step, Panel, and Automatic Intercept Systems. He is currently working in the Traffic Measurements Development group.

Kathryn E. Greisen, Sc.B. (Applied Mathematics), 1968, Brown University; M.S. (Mathematics), 1971, Stevens Institute of Technology; Bell Laboratories, 1968—. Miss Greisen has worked on various aspects of file administration programming for the Automatic Intercept System, primarily including development of the intercept data audit program. Currently she is engaged in development work on a minicomputer-based system for centralized real-time management of the telephone switching network.

William G. Hall, M.S. (E.E.), 1956, Case Institute of Technology; Bell Laboratories, 1956—. Mr. Hall was engaged initially in research in telephone switching systems, centered for several years on time-division switching and PCM coding, especially in the context of the ESSEX experiment. He helped initiate exploratory studies of computer aids to telephone directory assistance service, and continued that work for a time in the Switching Development Division. He holds several patents in telephone switching and logic, PCM coding, and information retrieval. From 1967 to 1972 he was a supervisor in the Automatic Intercept System Department. Currently he is supervising continuing development of programs for the Plug-in Inventory Control System, a BISP product. Member, IEEE, Sigma Xi.

J. W. Hopkins, Bell Laboratories, 1953—. Mr. Hopkins was initially associated with the research area, working first on the Mark 65 digital computer project and later becoming involved in mobile radio propagation studies. During this latter period, he worked on the design of narrowband IF amplifiers and special instrumentation for propagation measurements. Upon joining the development area, he became engaged in transmission studies on the No. 101 ESS, 1A switch unit, particularly with respect to the crosstalk problems of time-division switching. Since joining the AIS development group, he has been primarily concerned with the magnetic disc memory portion of the system.

P. D. Hunter, B.E.S., 1962, M.E. (Electrical Engineering), 1963, University of Florida; Bell Laboratories, 1964—. Mr. Hunter initially did design of the File Subsystem hardware on the AIS; he then did programming of maintenance programs related to the File Subsystem; presently he is working on the File Access Subsystem, which is a minicomputer configuration providing data for the AIS files. Member, Sigma Tau, IEEE.

Charles W. Kemp, B.S., 1961, M.S., 1962, University of Utah; Bell Laboratories, 1962—. Mr. Kemp is working in the telephone switching system development area. His interests are primarily in the diagnostic area, both hardware and software. He is currently a student at the University of Pennsylvania studying for a Ph.D. in electrical engineering. Member, IEEE, ACM, Tau Beta Pi, Eta Kappa Nu.

R. E. Machol, B.S., 1960, B.S.E.E., 1961, Trinity College; M.E.E., 1964, New York University; Bell Laboratories, 1962—. Mr. Machol initially worked on the Traffic Data Recording System; later he was engaged in system studies related to automating Directory Assistance. He also worked on the design of the Automatic Intercept System. Presently he is supervisor of the group responsible for developing a new traffic data collection and network management system. Member Sigma Pi Sigma.

Robert N. Markson, B.E.E., 1961, Union College; Master of Engineering, 1963, Yale University; Bell Laboratories, 1963—. Mr. Markson worked on computer simulation of logic control circuitry for PBX–Automatic Identified Outward Dialing and then maintenance programming for the position subsystem of the Traffic Service Position System. He is presently working on the design and development of trunk and service circuit maintenance programs for the Automatic Intercept System. Member, IEEE.

Philip J. Moylan, B.S.M.E., 1961, University of Maryland; M.S.E.E., 1963, New York University; Bell Laboratories, 1961—. Mr. Moylan was first engaged in system design and programming for the Electronic Translator System for the No. 4 Crossbar Toll Switching System. More recently, he has worked on the development of maintenance programs for the Automatic Intercept System time-division switching network. Member, Tau Beta Pi, Pi Tau Sigma, IEEE.

Joseph Orost, RCA Institutes, 1956; B.S.E.E., 1962, Polytechnic Institute of Brooklyn; Bell Laboratories, 1941—. Mr. Orost was responsible for the design of service observing circuits for the military. Later he developed the 8A and 9A announcement systems used for weather and news announcements. He also designed the Group Alerting System used for alerting volunteer firemen and other groups requiring an emergency alerting system. He did system planning, design, and programming work on the Traffic Service Position System mechanized training system, and system and circuit design work on the Automatic Intercept System.

R. J. Piereth, B.S.E.E., 1967, Newark College of Engineering; M.S.E.E., 1969, Rutgers–The State University; Bell Laboratories, 1961–1971; AT&TCo, 1971—. Mr. Piereth worked in No. 101 ESS, No. 2 ESS, Automatic Intercept, and Traffic Measurements at Bell Laboratories before going to AT&TCo in 1971. Presently he is Assistant Engineering Manager–Local Switching, where his responsibilities include Traffic Measurement Systems and Force Adjustment Systems. Member, Eta Kappa Nu, Tau Beta Pi, IEEE.

Gerald C. Vogel, B.S.E.E., 1969, Clarkson College of Technology; M.S.E.E., 1971, Columbia University; Bell Laboratories, 1969—. Mr. Vogel has worked on the development support system for the Automatic Intercept System since joining Bell Laboratories. Presently, he is working on the design of the File Access Subsystem for AIS. Member, IEEE, Tau Beta Pi, Eta Kappa Nu.

Bruce D. Wagner, B.E.E., 1965, and M.E.E., 1966, Cornell University; Bell Laboratories, 1966—. Mr. Wagner worked initially on exploratory development of a system for aiding directory assistance operators. He later worked on development of executive control, audit, and initialization programs for AIS. He presently supervises a group responsible for developing operational programs for new features in the Traffic Service Position System No. 1. Member, IEEE.

Douglas J. Wells, B.A. (Mathematics), 1967, Washington State University; M.S. (Mathematics), 1970, Stevens Institute of Technology; Bell Laboratories, 1967–1973; Wisconsin Telephone Company, 1973—. Mr. Wells was engaged in the Automatic Intercept Center file administration program development, and continued designing new features for the system. Currently he is on assignment as central office foreman for two No. 1 ESS offices.

William A. Winckelmann, New Jersey Bell Telephone Company, 1933–1961; Bell Laboratories, 1961—. Mr. Winckelmann was engaged in various management activities in the New Jersey Bell Telephone Company related to the maintenance of local and toll central office equipment and later in Traffic Engineering. Since joining Bell Laboratories, he has been involved in the development of group alerting systems, toll testboards, pressure range detectors for exchange cable, and the Automatic Intercept System.

Robert M. Wolfe, B.S.E.E., 1952, University of Louisville; M.S.E.E., 1957, Columbia University; Bell Laboratories, 1952—. Mr. Wolfe's early work included research in ferro-electric and magnetic devices. Since 1963, he has been Head of the Automatic Intercept System Department. Member, IEEE, ACM.

Walter W. Wood, B.S. (Mathematics), 1964, St. Francis College; M.A. (Mathematics), 1966, St. John's University; M.S. (Computer Science), 1972, Stevens Institute of Technology; Bell Laboratories, 1966—. Mr. Wood worked initially on developing call processing programs for AIS. He later was responsible for the development of the plant change program for AIS and worked on the design of additional operator services for AIS. His latest work is on a real-time data collection system. Member, Pi Mu Epsilon.