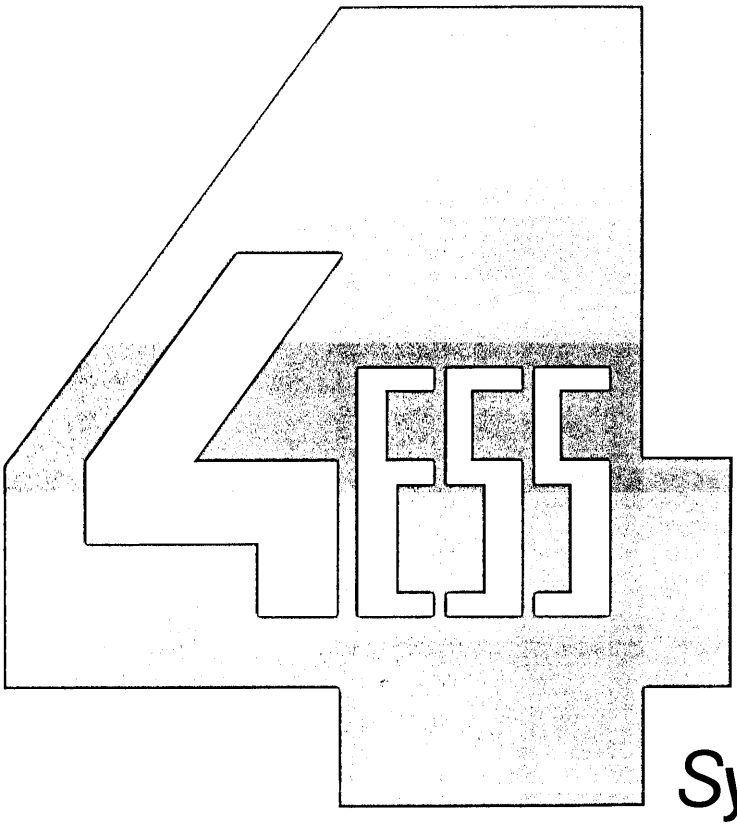
THE JULY–AUGUST 1981
VOL. 60, NO. 6, PART 2

THE BELL SYSTEM TECHNICAL JOURNAL



System Evolution

THE BELL SYSTEM TECHNICAL JOURNAL

THE BELL SYSTEM TECHNICAL JOURNAL is published monthly, except for the May–June and July–August combined issues, by the American Telephone and Telegraph Company, C. L. Brown, Chairman and Chief Executive Officer; W. M. Ellinghaus, President; V. A. Dwyer, Vice President and Treasurer; F. A. Hutson, Jr., Secretary. Editorial inquiries should be addressed to the Editor, The Bell System Technical Journal, Bell Laboratories, Room WB 1L-331, Crawfords Corner Road, Holmdel, N.J. 07733. Checks for subscriptions should be made payable to The Bell System Technical Journal and should be addressed to Bell Laboratories, Circulation Group, Whippany Road, Whippany, N.J. 07981. Subscriptions $20.00 per year; single copies $2.00 each. Foreign postage $1.00 per year; 15 cents per copy. Printed in U.S.A. Second-class postage paid at New Providence, New Jersey 07974 and additional mailing offices.

ISSN0005-8580

Comments on the technical content of any article or brief are welcome. These and other editorial inquiries should be addressed to the Editor, The Bell System Technical Journal, Bell Laboratories, Room WB 1L-331, Crawfords Corner Road, Holmdel, N.J. 07733. Comments and inquiries, whether or not published, shall not be regarded as confidential or otherwise restricted in use and will become the property of the American Telephone and Telegraph Company. Comments selected for publication may be edited for brevity, subject to author approval.

THE BELL SYSTEM TECHNICAL JOURNAL

DEVOTED TO THE SCIENTIFIC AND ENGINEERING ASPECTS OF ELECTRICAL COMMUNICATION

Volume 60 | July–August 1981 | Number 6, Part 2

No. 4 Electronic Switching System

THE BELL SYSTEM TECHNICAL JOURNAL

DEVOTED TO THE SCIENTIFIC AND ENGINEERING ASPECTS OF ELECTRICAL COMMUNICATION

Volume 60 July-August 1981 Number 6

No. 4 ESS:

Prologue

By K. E. MARTERSTECK

(Manuscript received July 30, 1980)

Since the cutover of the first No. 4 Electronic Switching System (ESS) office in January 1976, a program of system evolution has been carried out. As a result significant cost reduction was achieved, major new features have been added to the No. 4 ESS, and the system's performance has been enhanced. This paper gives an overview of the continuing development of the No. 4 ESS carried out during the period 1976 to 1980 as a prologue to a series of papers in this volume which describe some of the specifics of this activity.

I. BACKGROUND

In January, 1976, the first No. 4 Electronic Switching System (ESS) was placed in service in Chicago, Illinois. This culminated the largest single system development ever undertaken in the Bell System, a development which produced a high-capacity toll and tandem digital switching system. The No. 4 ESS, with its powerful central processor and time division network, brought to the Bell System telecommunications network significant improvements in flexibility, reliability, and economy compared with its electromechanical predecessors.

However, except for the large increase in capacity, the basic toll-switching features of the early No. 4 ESS machines were essentially a very modern version of the No. 4A crossbar features. Therefore, the development of the No. 4 ESS did not end with the cutover of the Chicago 7 office. Instead, a program of evolution of the system planned

to achieve significant cost reduction, as well as feature additions and performance improvements, was vigorously pursued. Stimulated by dramatic advances in integrated circuit technology and improved software design and circuit interconnection techniques, the No. 4 ESS was virtually completely redesigned in the period from 1976 to 1980. Five major new versions of the software package, called generics, along with new hardware designs, were introduced on roughly yearly intervals during this period. These generics added some additional toll functions and major new network revenue-producing features; produced reductions in cost, power, and space; and improved reliability and maintainability.

II. INITIAL SYSTEM IMPLEMENTATION

The 1976 architecture of No. 4 ESS is shown in simplified form in Fig. 1.[1] In this original design, analog signals were converted to voiceband by transmission terminal equipment and then converted to pulse-code modulated (PCM) signals and multiplexed into DS-120 streams in the Voiceband Interface Frame (VIF). A Signal Processor Type 1 (SP1) was connected to the E and M leads from/to the transmission terminal equipment. The SP1 detected and interpreted state changes on the E lead and generated appropriate state changes on the M lead. Digital carrier (T1) signals entered the system at the digroup terminal (DT), where these signals were multiplexed into the DS-120 format. The signal processor type 2 (SP2) derived supervisory states from the incoming PCM stream and generated supervisory states in the transmit direction for inclusion in the outgoing PCM stream.

The No. 4 ESS time-division switching network contains six stages of time-shared switching: time-space-space-space-space-time. The first and last pairs of switching stages are implemented in the time-slot-interchange frame (TSI). The TSI also performs a decorrelating function by which the PCM signals in successive time slots in each incoming DS-120 digital stream are spread in both time and space as a result of the switching action. This ensures spreading of traffic over the network and eliminates the need for load balancing. The middle two stages of time-shared space switching are provided by the time-multiplexed switch (TMS). The pattern of network connections in this 1024-by-1024 switch is changed 1.024×10^6 times every second. The No. 4 ESS is internally synchronized by an extremely precise and reliable clocking system consisting of four crystal-controlled oscillators operating at 16.384 MHz. In normal operation, one oscillator is designated the master, supplying the timing pulses for the network, while the remaining three oscillators are in standby, phased-locked to the master.

The entire No. 4 ESS is under the control of the powerful 1A Processor, which has a central control and three memory types:

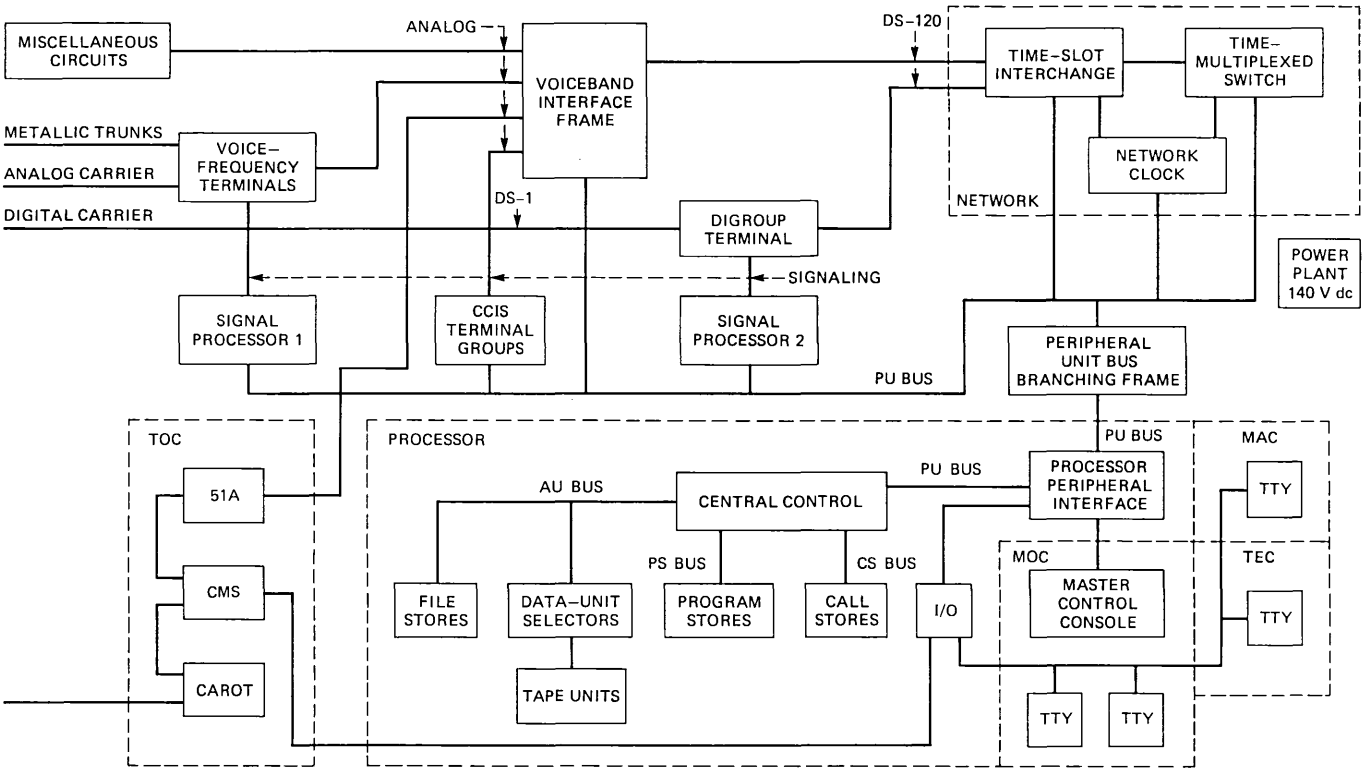


Fig. 1—No. 4 ESS: 1976.

program store, call store, and file store. The program store contains the fixed set of software instructions. The call store contains the time-variant data associated with setting up calls and handling other office activities. The call store also contains translation data describing the office configuration and prescribing the office-specific call-handling parameters. The file store on disk is used primarily as a backup for the program store and the fixed data in call store. The file store also contains less-frequently used programs, such as diagnostic routines. The original program and call stores were coincident-current ferrite-core arrays with fetch cycles of 1400 ns, twice the 700-ns cycle time of the central control.

III. SYSTEM EVOLUTION

The continuing development of the No. 4 ESS affected both the software and hardware architecture of the system. When the No. 4 ESS was first placed into service, its complement of software consisted of approximately 1.4×10^6 stored words in over 800 identifiable functional units called PIDENTS, written primarily in a language called EPL (ESS programming language). EPL is an "intermediate level" language, more powerful than assembly language, but not as powerful as a "high level" language. EPL, together with a set of macros, offers designers a degree of mechanization while maintaining tight control over the use of real time—a matter of constant concern because of the very high switching capacity requirements. However, to increase the productivity of the software development staff by taking advantage of more modern software technology, a high-level language for switching, called EPLX, (ESS programming language extra) was developed for use in the No. 4 ESS. EPLX has modern control constructs and high-level data description and data structure reference capabilities. Since the EPLX compiler produces somewhat less efficient code in terms of memory and real-time consumption, when these factors are paramount the designer can still mix EPL with the EPLX on a module-by-module basis.

An extensive amount of new software has been written for No. 4 ESS to introduce new call-processing features, expanded administrative capabilities, and improved maintenance and fault-recovery operations. As part of the development of the new international switching functions using CCITT No. 5 and No. 6 signaling, the call-processing software was restructured to improve its flexibility and maintainability. Over the last several generics the maintenance and fault-recovery software was modularized and restructured under a special operating system to facilitate the inclusion of new peripheral frames into the system. This new software was written in EPLX. Also, with each new peripheral frame type introduced into No. 4 ESS, a considerable quantity of new fault-recovery, maintenance, and diagnostic software was written. Virtually all areas of administrative function, including recent change-

and-verify, trunk-maintenance, network-management, report-generation, and operations-support-system interfaces, have been augmented and expanded. In addition to the international call-switching functions, major new functions such as Common Channel Interoffice Signaling Inward Wide Area Telephone Service, and the Mass Announcement System have been added. As a result of all this software development activity, the 1980 No. 4 ESS generic contains in excess of 2.1×10^6 words of software program.

Between 1976 and 1980, the evolution of No. 4 ESS hardware has been comparably extensive (Fig. 2). In the 1A Processor, the memory was upgraded from the original ferrite-core memory arrays to semiconductor memories. The transition was accomplished in two stages. In 1977, metal oxide semiconductor (MOS) integrated-circuit memories in 65,536-word modules were introduced into the system. The fetch-cycle time of these memories was 1400 ns, the same as that of the core memories. In 1979, newer semiconductor memories with 262,144-word modules were added. These memories have a 700-ns fetch-cycle time capability. Thus, while the 1A Processor central control was designed to function with a mixture of all three memory types, the processors equipped entirely with the newer semiconductor stores running in the fast mode (700-ns cycle) have 30 percent more processing capacity.

In addition to the memory upgrade, a new high-speed input/output processor was developed for the 1A Processor to handle all the various I/O functions for the system, including interfaces with operations support systems, such as the Engineering and Administrative Data Acquisition System and the Circuit Maintenance System.

Although the frames in the time-division network perform similar functions to the original No. 4 ESS network frames, major improvements have been made in all the frame types. The availability of medium-scale-integration bipolar memories to replace the original small-scale-integration insulated gate field-effect transistor (IGFET) memories in the random access memories of the original frames led to a redesign of the TSI and TMS frames. Also, bulk dc-to-dc power converters with 100-ampere capacity were developed. Thus, new TSI and TMS frames were produced which offered savings in cost, space, and power consumption.

The network clock was also enhanced to allow automatic synchronization to a master clock source. Even though the original network clock had excellent long-term stability, the individual clocks had to be periodically manually adjusted to ensure the synchronization necessary to prevent data from being lost when transmitted over digital facilities between digital offices. The new automatic synchronization obviates the need for manual adjustments and ensures more reliable operation of the toll network.

The original architecture of No. 4 ESS utilized echo suppressors for

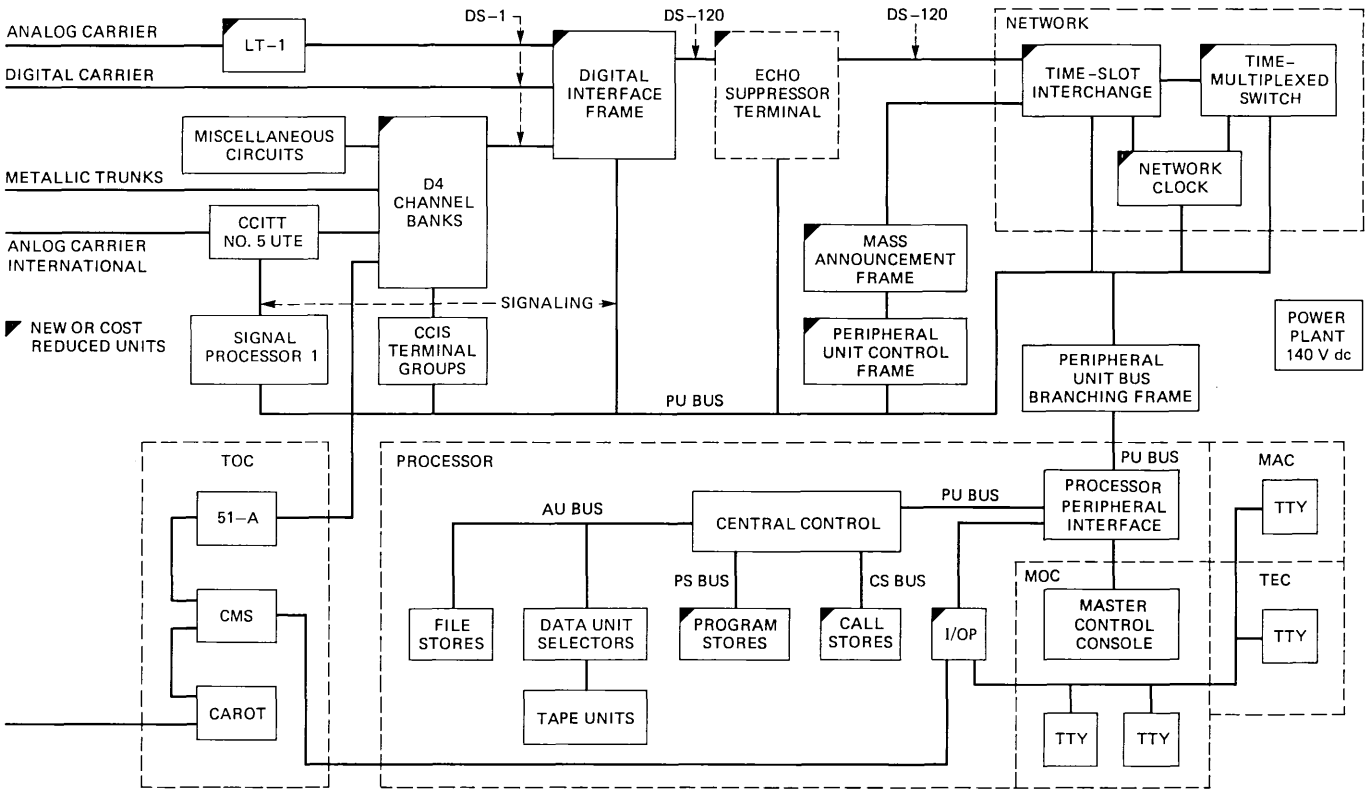


Fig. 2—No. 4 ESS: 1980.

long-transmission circuits. These echo suppressors, which employed standard analog techniques, were mounted in transmission terminal equipment, one per voice channel. Subsequently, a digital echo-suppressor terminal was designed to operate in a time-shared mode on the DS-120 stream. Functionally the digital echo suppressor is equivalent to its analog predecessor, but is significantly more economical. This evolution will continue with the provision of digital echo cancellers, implemented with very-large-scale integrated circuit chips.

There have been significant modifications to the transmission terminals in No. 4 ESS, particularly regarding the handling of the analog carrier interface. Initially, group band carrier was connected to terminal equipment that converted the signals to baseband. Signaling was extracted for processing by the SP1 while the baseband information passed into the VIF where it was converted to PCM and multiplexed into DS-120 streams. With the availability of the DT/SP2 complex, it became economically attractive to use PCM carrier terminals (such as D4) to provide the analog-to-digital conversion. Further economies were realized by combining functions. An LT-1 connector frame was developed to convert 12-channel analog carrier groups to the PCM DS-1 format. Also, a Digital Interface Frame (DIF) was developed to handle the DT/SP2 functions. The DIF is a microprocessor-controlled frame which uses the most current semiconductor device and packaging technologies.

Finally, powerful new functional capabilities have been provided in the Bell System network with the addition in No. 4 ESS of the Mass Announcement System frame and its associated Peripheral Unit Control (PUC) frame. Like the DIF, these frames are microprocessor controlled and use the latest device technology.

Table I gives a chronology of the introduction of the major new No. 4 ESS features described briefly above.

Table I—Major new features

1976	Basic toll features plus Common Channel Interoffice Signaling (CCIS).
1977	Digital echo suppressor, cost-reduced Digroup Terminal (DT), 64K-word 1400-ns semiconductor store.
1978	International-call switching exchange functions using CCITT No. 5 and 6 signaling, cost-reduced Time Slot Interchange (TSI), Input/Output Processor (IOP), and maintenance and administrative enhancements.
1979	CCIS—Inward Wide Area Telecommunications Service (INWATS), 256K-word 700-ns semiconductor stores, switching control center interface, maintenance and administrative enhancements, and reduced system reinitialization time.
1980	Digital Interface Frame (DIF), LT-1 connector, cost-reduced Time Multiplexed Switch (TMS), network clock synchronization, mass announcements, and maintenance and administrative enhancements.

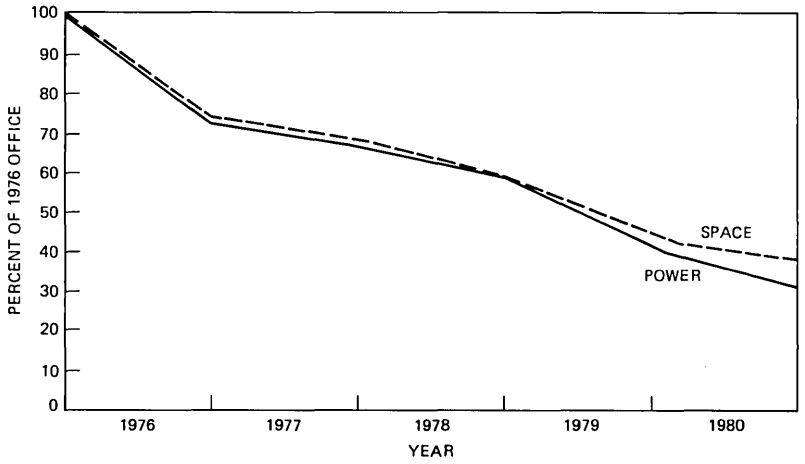


Fig. 3—No. 4 ESS evolution space and power drain (40K trunk office).

IV. SUMMARY

When the No. 4 ESS was initially designed, the best available design technology was utilized. However, as technology advanced, new opportunities were presented to simplify the No. 4 ESS architecture as well as add new features and improve overall system performance. Space and power requirements (see Fig. 3), as well as cost, were reduced. System reliability has continuously improved throughout the period and the system has been kept technologically modern.

This issue of the *Bell System Technical Journal* contains a series of papers which detail the No. 4 ESS hardware and software evolution and discuss some of the significant new capabilities which have been incorporated into the system. The issue concludes with a presentation of the performance achieved by the No. 4 ESS in the field.

V. ACKNOWLEDGMENTS

The continuing development of No. 4 ESS is the result of dedicated effort by people in many organizations throughout Bell Laboratories and Western Electric working in close concert with AT&T General Departments, Long Lines, and the operating telephone companies. The authors of this volume express their gratitude and appreciation to the entire No. 4 ESS team, without whose cooperation and support these objectives could not have been successfully met on schedule.

REFERENCE

1. B.S.T.J., *56,* No. 7 (September 1977).

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 60, No. 6, July-August 1981
Printed in U.S.A.

No. 4 ESS:

Mass Announcement Capability

By R. J. FRANK, R. J. KEEVERS, F. B. STREBENDT, and J. E. WANINSKI

(Manuscript received August 26, 1980)

We describe the mass announcement capability that has been introduced in No. 4 ESS beginning with the 4E5 generic. This capability allows various sponsors to provide services in which a large volume of callers can dial advertised numbers to listen to Public Announcement Service announcements, register their opinions via telephone calls, or participate in call-ins whereby randomly selected callers are connected to a celebrity for a live answer. Although these kinds of services are not new, they have usually been offered on a limited, individually engineered basis at high administrative cost to the telephone network. Mass announcement capability provides a general means to accommodate sponsor-provided announcement-related services which can be offered on a local, regional, or national basis.

I. INTRODUCTION

Public demand exists for expanded new uses of the telephone to provide information and entertainment. Hearing a recorded announcement over a telephone is increasing in popularity. Radio and television stations are encouraging public participation in telethons and call-ins. In the past, the scope of these kinds of services has been limited. More extensive services, such as a presidential call-in, have required special engineering at substantial cost. When the telephone company has not been consulted in advance, peaked traffic caused overloads and widespread congestion in the telephone network. The challenge of the Mass Announcement System (MAS) feature is to provide the mechanism so that sponsors can offer these kinds of announcement-related capabilities on a widespread basis to fulfill existing needs and yet to contribute

significant revenues for the Bell System. High capacity, flexibility, and network protection are key factors because the served area may be large or heavily populated, many different services may be provided simultaneously, and calling may be stimulated by media programming.

The MAS is a major part of the 4E5 generic development on the No. 4 ESS. The No. 4 ESS provides a means of recording announcements and the capability to connect a large volume of callers to these announcements. The high terminating capacity of the No. 4 ESS and its position in the Direct Distance Dialing (DDD) network make it a viable switching machine for providing MAS services. The technical architecture inherent in the No. 4 ESS time division/space division network provides an efficient mechanism for transmitting the announcements in a digital format.

The Mass Announcement System is an optional No. 4 ESS feature. No. 4 ESS offices equipped with MAS will be strategically deployed throughout the country so that sponsors can provide service on a local, regional, or national basis. Each of these offices will be designated as an MAS node and its associated calling region will be designated as an MAS island. National MAS coverage for initial service in 1980 is illustrated in Fig. 1.

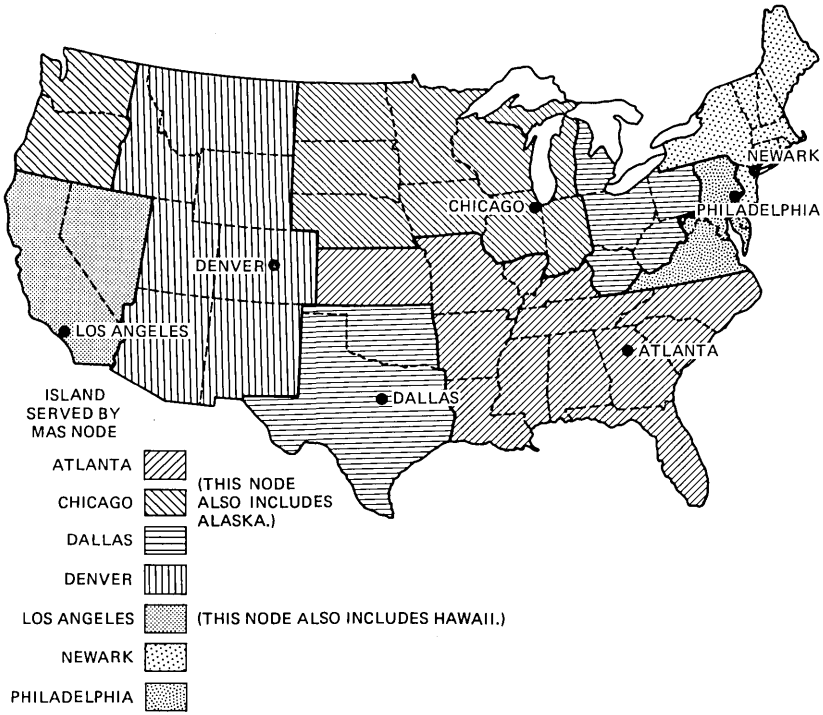


Fig. 1—Seven-node national service for 1980.

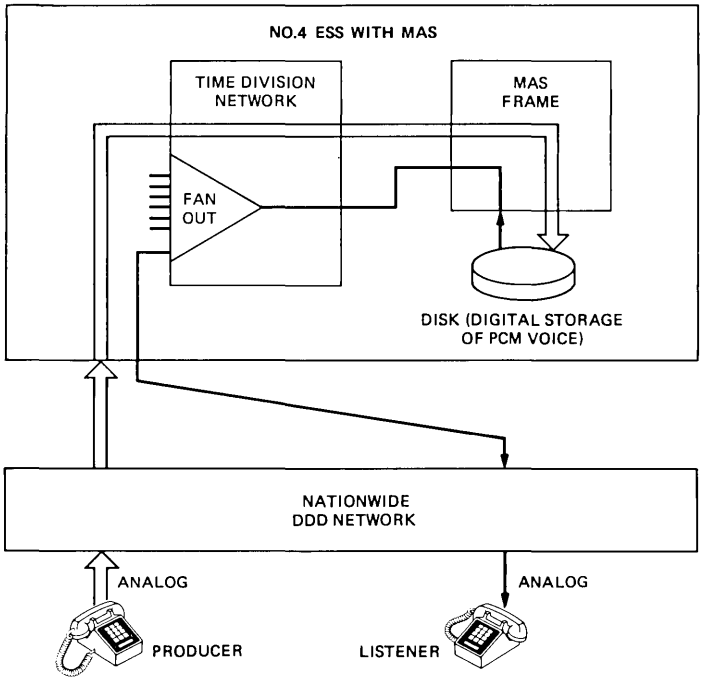


Fig. 2—Mass Announcement System digital recording.

II. MASS ANNOUNCEMENT FEATURE REQUIREMENTS

2.1 Definition of offered features

The MAS feature consists of a variety of announcement-related services in which a large volume of callers dial an advertised number and expect to hear a prerecorded announcement. The main types of service are as follows:

(*i*) Public Announcement Service (PAS)
(*ii*) counting of media stimulated calls
(*iii*) cut through [typically Media Stimulated Calling (MSC)].

2.1.1 Public announcement service

The PAS tends to be stable in nature with reasonably predictable calling patterns. Requests for weather or time are typical applications which have been carried on the network for many years. More recently new forms of PAS have emerged. Some examples are news, horoscope offerings, jokes, sports results, and other entertainment programs.

A No. 4 ESS with the MAS feature provides capability for sponsor-offered announcement recordings, quick updating, and the means for transmitting these announcements, as shown in Fig. 2. An MAS frame can provide a large number of varying length synchronous announcements which start over again every 15 s. A caller to an MAS announce-

ment is generally supplied with audible ringing until the announcement starts.

Public Announcement Service announcements may also be provided by an audio source other than the MAS frame. These are called barge-in announcements since callers are connected to these announcements at any point in the announcement cycle after hearing audible ringing.

2.1.2 Counting of media stimulated calls

Counting of media stimulated calls is a service in which a sponsor can stimulate callers through advertising to register their opinion by telephone on a topic of general interest. If the question posed on television, on radio, or in the print media has a yes or no answer, then two telephone numbers would be assigned to have corresponding significance. If a dozen candidates were nominated for "most valuable player," then it would be necessary to provide 12 telephone numbers. Callers hear an MAS announcement and the No. 4 ESS counts each call to the specified number. More than one No. 4 ESS office can participate in the same MSC counting application. The accumulated counts from all No. 4 ESS offices participating in the same application can be output to a sponsor's location in near real time, thus, allowing these MSC counting applications to be coordinated with radio or television programming.

2.1.3 Cut-through service

Cut-through service is another sponsor-offered MSC service which permits selective access to a telethon or call-in sponsor while the large majority of callers are diverted to a customized MAS announcement. The selectivity relates to one call per unit time which is forwarded to another DDD directory number to be given personal attention, perhaps by a politician or celebrity. Cut-through service can be offered in conjunction with an MSC call counting service as a means of soliciting additional information from a sample of the callers expressing their opinion.

2.2 Specification of No. 4 ESS system requirements

Key attributes of the MAS services are specified in the following sections in terms of minimum and maximum bounds.

2.2.1 Call terminations

The basic capacity of a No. 4 ESS office with a minimum MAS equipment configuration of one MAS frame and two dedicated time-slot interchange switching and permuting circuits (TSI SPCS) for simultaneous call termination is 896 per dedicated TSI SPC or 1792. Using 90 percent occupancy on the dedicated TSI SPCS, 7.5-s average wait time,

and a 30-s holding time per call, there could be approximately 150,000 calls per hour to one announcement or to a mix of MAS announcements available on the two MAS-dedicated TSI SPCs. To provide perspective, the busy hour call capacity of the No. 4 ESS is approximately 500,000. An additional 896 simultaneous call terminations can be handled with each additional MAS-dedicated TSI SPC, of which there is a maximum of 15. However, any one announcement can be available from a maximum of two MAS-dedicated TSI SPCs.

2.2.2 Announcement capacity

There can be from 1 to 8 MAS frames in a No. 4 ESS office. Each MAS frame can provide a maximum of 59 30-s MAS announcements. However, MAS announcement basic building blocks are in terms of 30-s sectors. These 59 sectors can be assigned as desired. If an announcement lasts less than 30 s, one must nevertheless devote a 30-s time sector to that purpose. Announcements can range from 30 s to 5 min in length. Ten 30-s sectors must be allocated to provide the necessary resources for a 5-min announcement.

Each MAS frame has capacity for a total of eighty 30-s sectors of audio. A maximum of 59 of these can contain active audio, that is, audio which is playing back to callers. The balance of these sectors can be used to store standby audio, which is not yet available to callers.

The number of barge-in announcements in a No. 4 ESS can vary from 0 to 24. Barge-in announcements can exist even though an office does not have an MAS frame.

2.2.3 Announcement characteristics

Much flexibility exists in defining each MAS announcement in the No. 4 ESS. The length of an MAS announcement can be from 30 to 300 s but must be a multiple of 30 s. A barge-in announcement can be from 5 s to 300 s long. Multiple plays, if specified, allow three options in which callers can hear the audio two or three times or repeatedly (for about $2\frac{1}{2}$ hours). Callers are automatically disconnected from hearing the announcement after the specified number of plays. The charge option, if specified, results in the No. 4 ESS returning answer supervision, thus, resulting in the caller being charged for calling the announcement. The forced audible ringing option ensures that the caller hears at least one cycle of audible ringing before the announcement audio starts.

Announcement audio can be rapidly updated via any one of several methods, most of which are external to the No. 4 ESS. A maximum of 28 MAS announcements per MAS frame can be simultaneously updated.

Announcements are available to the calling public according to start and stop time parameters for each announcement. These parameters

can be specified so that announcement audio is activated for playback to callers immediately and plays continuously. Or, these start or stop time parameters can specify scheduling up to 23 hr in advance.

An announcement application can have its audio in the standby state or in the active state or in both. However, an announcement application can have at most two audio copies or versions existing on the MAS disks—one in the active state and one in the standby state. Audio in the active state is playing back to callers. Audio in the standby state is scheduled to go active and thereby to replace the active copy, if it exists, at some specified start time.

Mass Announcement System announcements are synchronous and start from the beginning every 15 s. Thus, callers normally wait from 0 to 15 s to be connected to the beginning of audio, with an average waiting time of $7\frac{1}{2}$ s. This wait time may be increased for announcements defined with forced audible ringing.

2.2.4 Capabilities for counting media stimulated calls

Counting media stimulated calls is a service that involves one or more MAS announcements, as well as the pegging, collection, and output of counts of customer calls on a dialed number basis.

In a No. 4 ESS MAS node, there are 128 dialed number counters which can be used for up to 16 different MSC counting applications simultaneously. Up to 64 counts can be collected in connection with one MSC counting program. Number patterns such as 900-234-0001 through 900-234-0064 could be counted separately, yet, cause routing to a single announcement which might simply say, "Thank you for calling. Your opinion has been counted." On the other hand, there could be a separate announcement for each dialed number or any grouping of numbers.

For one MAS node, the MSC counting totals can be transmitted to a sponsor on a minute-by-minute basis via a dial-up connection which consists of a data terminal with a mating unit on the other end of a DDD connection. Alternatively, if multiple No. 4 ESSs are involved in the same MSC counting application, each No. 4 ESS reports results to a designated master No. 4 ESS, which can transmit the results to the service sponsor. Each such No. 4 ESS, which counts customer calls and sends these results to a master, is referred to as a slave. A given No. 4 ESS MAS office can perform slave, master, or both functions. A master function, however, can exist in a No. 4 ESS without MAS. However, that No. 4 ESS must have the 4E5 (or later) generic.

A master can have any number of slaves reporting to it. Figure 3 shows an MSC counting application involving callers in three MAS nodes. Slaves update master counts every 5 min via Common Channel Interoffice Signaling (CCIS) direct-signaling messages. In each No. 4

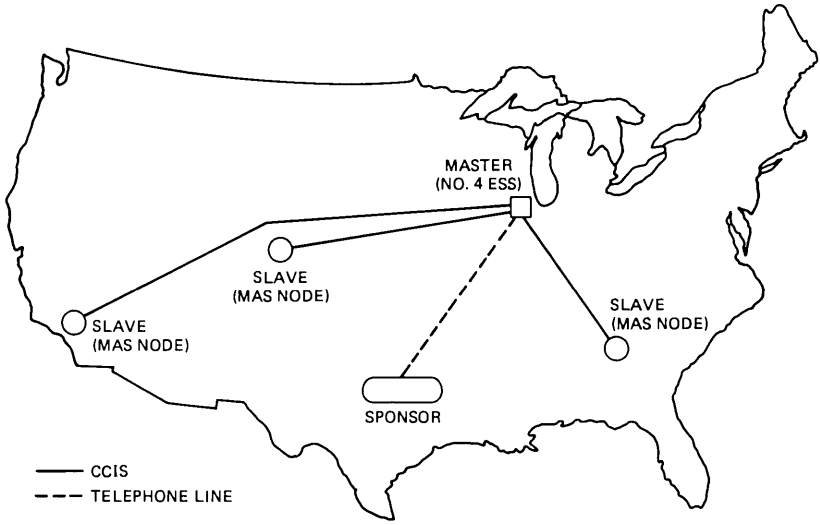


Fig. 3—Counting of media stimulated calls.

ESS office, there are 128 master counters which can be used for up to 64 different MSC counting applications simultaneously. A maximum of 64 master counters can be used for one master application.

A No. 4 ESS office can simultaneously support a maximum of five dial-up connections to sponsors who desire to receive minute-by-minute MSC counting results as counts are being tabulated. This number may be less if the office engineered number of dial-up ports is less. However, any number of master applications not exceeding the 64 limit can transmit to the same sponsor simultaneously over a single dial-up connection.

An MSC counting application can be scheduled according to specified start and stop times, or it can run continuously, or it can cycle on and off on a daily basis. An MSC count scheduling is independent of scheduling of the MAS announcement(s) associated with an MSC counting application.

Flexibility exists to set up a national MSC counting application in which calls are counted at the same hour relative to each time zone. For example, callers may be stimulated to call from 7 to 8 p.m. in their own time zone. The master can serve as a master and slave in its time zone so that calls coming from that time zone are counted only from 7 to 8 p.m., but counts from other nodes are accumulated until stop time has been reached in all slave nodes for this application.

2.2.5 Cut-through capabilities

A cut-through application can be applied to any 10-digit directory

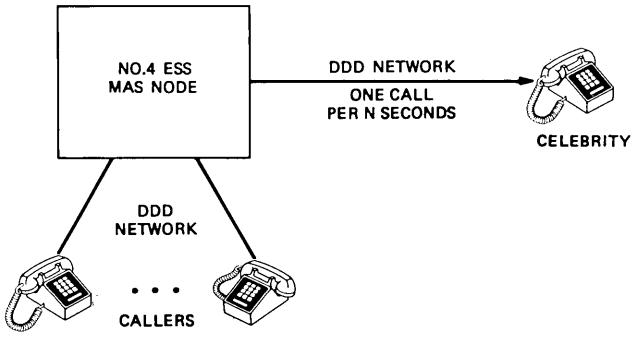


Fig. 4—Cut-through application.

number.* Up to 64 cut-through applications (shared with network management gap controls) can be applied simultaneously in any one No. 4 ESS.

In a cut-through application, one call is cut through to a prespecified DDD directory number roughly every N seconds as shown in Fig. 4. N is referred to as a call-gapping interval. The call-gapping interval for a given cut-through service can be selected from 16 different values ranging from 0 to 360 s.

Each cut-through application must have a different announcement associated with it. Conversely, any MAS announcement can have at most one cut-through application associated with it.

Similarly to counting of media stimulated calls, a cut-through application can be scheduled according to specified start and stop times, or it can run continuously, or it can cycle on and off on a daily basis. Cut-through scheduling is independent of scheduling of the MAS announcement associated with the cut-through application.

2.3 Specification of external interface requirements

Since MAS services may be local, regional, and national in scope, coordination of such things as dialable number, announcement capacity, announcement audio, and service schedules in all the No. 4 ESS MAS offices must be administered by one central source. An organization called the Operations Network Administration Center (ONAC) is responsible for administration of all MAS services. Two main support systems are used by ONAC personnel as shown in Fig. 5. One of these support systems is the MAS Support System (MSS), which is a computerized system minimizing manual administrative functions needed to

* Gap control, however, is a network management control described in Section 4.3.6 which can be applied on a 3-, 6-, 7-, or 10-digit basis.

define, schedule, and monitor MAS services in all the MAS nodes. The other of these support systems is the Announcement Distribution System (ADS) which accepts audio from the producer and then can simultaneously update all the No. 4 ESS MAS nodes which are to receive this audio. Although ONAC and both of these support systems are external to the No. 4 ESS, the No. 4 ESS must interface with them in a compatible manner.

In case a backup method is needed, all functions which ONAC performs must also be capable of being performed in a No. 4 ESS office to administer MAS services in that office. Another external audio update method called the direct producer update method is also needed as a backup in case ADS system failure occurs, or in case local applications may not be using ONAC.

2.4 Transmission plan

Since the overriding purpose of an announcement service is to deliver an audio product of high quality, considerable effort was placed on means to safeguard the fidelity of the ultimate product, voice playback. Three noteworthy opportunities to introduce impairments are readily identified. First a producer may transport an announcement by electronic means to a control location, such as the ADS. If this is a DDD connection, there is exposure to noise, loss, and possible echo in this transaction. The call must then be fed to one or more No. 4 ESS MAS nodes. Although dedicated trunks make possible tighter control of transmission variables, noise, for example, will be an additive

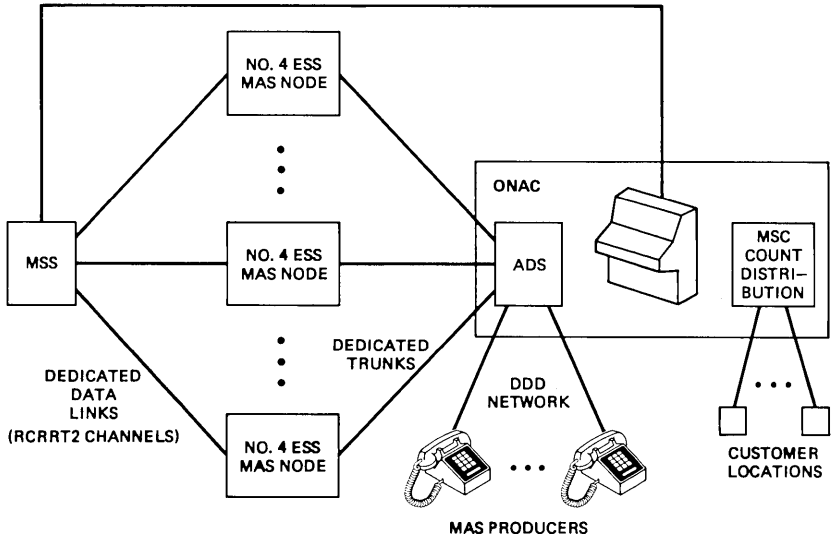


Fig. 5—No. 4 ESS/ONAC interfaces.

impairment. Finally, when digitized recording is played back over DDD, new hazards to fidelity must be anticipated. Since control between the ADS and the MAS node is most practical, special effort was placed on the parameters of this link. Here companding was applied in conjunction with the use of a pilot tone to permit positive checks on transmission level to be made during transmission. Loss of pilot tone would be a positive warning that a gap in the announcement had been experienced. At the MAS node, the pilot tone is filtered out, levels set and feedback provided to the source when repetition of the announcement is called for.

Transmission planning extended to the full network service, just as the signaling and switching planning did. It was necessary to evaluate peak power and average power effects when one popular announcement might be dominant over various facilities. The levels of ringing tone, busy tone, and the playback itself had to be established. Since the network is not homogeneous, simple universal answers were generally not to be found. Tentative answers were established, however, and means of adapting, if necessary, were investigated and identified. Both laboratory and field experiments were conducted. Limits previously found necessary to protect switching and signaling also found use in the transmission world. Thus, a total requirements package was constructed. Within its structure, means are established to allow large numbers of callers to have common access to versatile and customized telephone announcements.

III. NO. 4 ESS SYSTEM ARCHITECTURE FOR MAS

3.1 Mass Announcement System hardware complex

The minimum physical equipment configuration for a No. 4 ESS with MAS is illustrated in Fig. 6 and consists of the following components:

- *An MAS frame and two moving head disk units* record, store, and playback digitized voice announcements. (The maximum number of MAS frames is eight.)
- *A Peripheral Unit Control (PUC) frame* provides common operational and maintenance interfaces between the 1A Processor and the MAS frame. (One PUC frame can handle a maximum of two MAS frames.)
- *Two DS-120 links (two coaxial cable pairs)* connect an MAS frame to the No. 4 ESS network. Record, monitor, and playback channels are identified by their time-slot appearances on these links.
- *Two dedicated TSI SPCs* provide fanout of the announcement phases coming from the MAS frame. Any barge-in announcements and audible ringing are likewise fanned out by these dedicated TSI SPCs. Incoming MAS calls are terminated on these dedicated TSI SPCs. (The maximum number of dedicated TSI SPCs for MAS is 15.)

- *Two auxiliary audible ringing trunks from the ringing and tone plant* provide the ringing that callers hear before start of audio. (One of these is needed per dedicated TSI SPC.)
- *Connections from up to 24 barge-in announcement trunks* may also be optionally provided as shown in Fig. 6.
- *One or more automatic dial-up 1200-baud asynchronous channels* (not shown in Fig. 6) are required on the Input/Output Processor (IOP) frame. These dial-up channels are required for transmitting MSC counting results to a remote data terminal. (The maximum number of dial-up ports in a No. 4 ESS is six, but a maximum of five of these can be used for MSC count reporting to sponsors.)
- *An RCRRT2 (remote recent change) channel* (shown in Fig. 5) on the IOP frame is remoted to ONAC via a dedicated data link and allows ONAC personnel to enter and receive messages from a No. 4 ESS to define, control, and monitor MAS services.
- *A dedicated trunk subgroup* exists (shown in Fig. 5) between ONAC and the No. 4 ESS for recording announcements from ADS. This trunk subgroup must be uniquely identified with a special name in No. 4 ESS.

3.2 Utilization of dedicated TSI SPCs

The concept of a dedicated TSI SPC was introduced in the initial generic of the No. 4 ESS for the fanout of office announcements and tones. A dedicated TSI SPC is physically the same as a regular TSI SPC, except that its transmit and receive ports are looped with coaxial cables. A dedicated TSI SPC has dynamic fanout capability which enables all the callers connected to it to hear one announcement or any mix of announcements available on it according to current demand. This fanout capability of dedicated TSI SPCs avoids having to engineer terminations separately for each announcement service.

The MAS frame continually plays back active announcement phases into the network via the playback channels. Playback channels are "nailed up" from serving TSI SPCs to dedicated TSI SPCs as shown in Fig. 6. For reliability reasons, the two MAS submembers or units from the same MAS frame must be connected to different TSI frames. For further reliability reasons, the serving and dedicated TSI SPCs for each MAS submember or unit should be from the same TSI frame. Each dedicated TSI SPC fans out one audible ringing signal and a maximum of 104 announcement phases.

3.3 Mass Announcement System customer call strategy

Providing MAS announcements is a unique kind of function for a toll switching office. A large volume of customer calls is being terminated in, instead of being switched through, the No. 4 ESS. This large volume

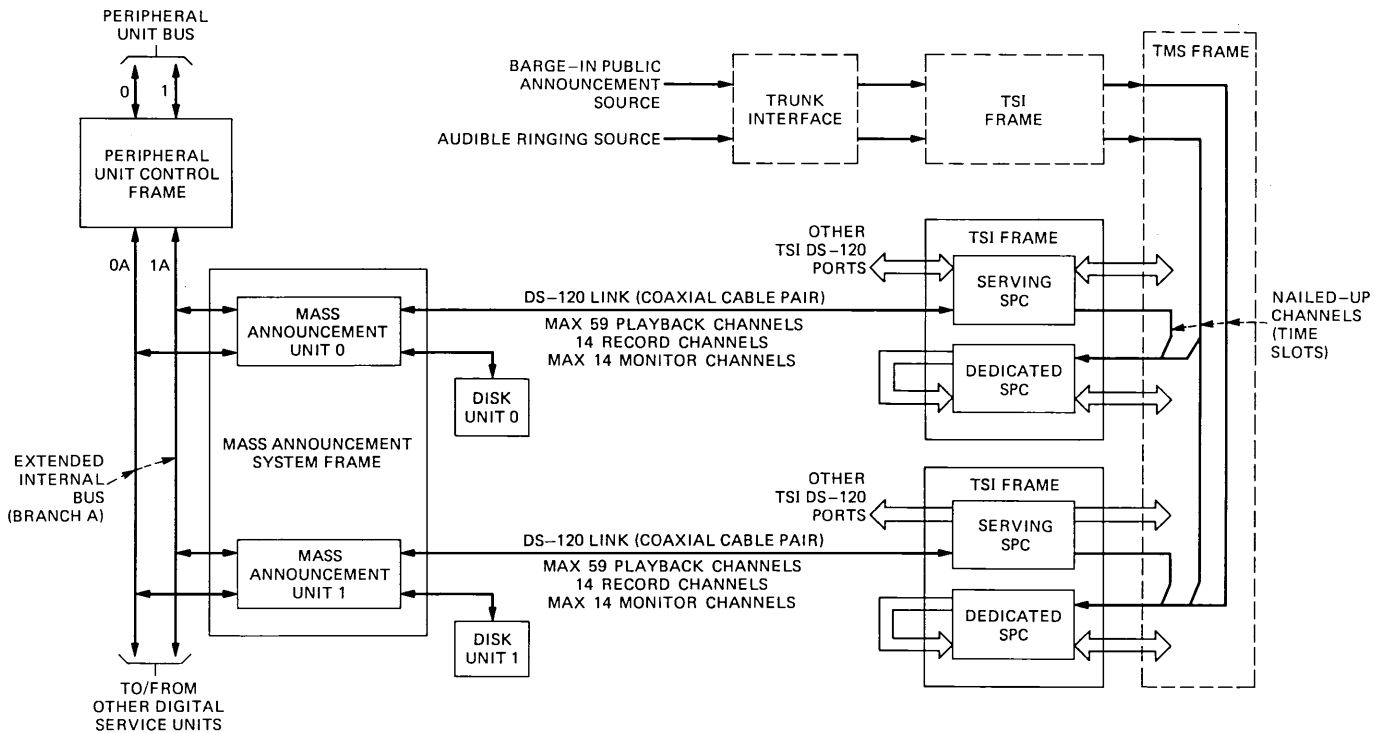


Fig. 6—Mass announcement system configuration.

of callers is being terminated on MAS-dedicated TSI SPCs within No. 4 ESS offices. Since any MAS announcement is available from two dedicated TSI SPCs, customer calls to that announcement are connected to the dedicated TSI SPC on which the next phase of the announcement occurs so that customer delay time is minimized. Customers hear audible ringing and announcement audio from the same dedicated TSI SPC.

An announcement start point in the MAS hardware is randomly selected at the time audio is first recorded. Thus, different announcements should phase at different times. This design was intended to spread simultaneous traffic load and the distribution of answer signals.

3.4 Announcement update strategy

Before any MAS announcement can be recorded initially, service provisioning data must have been input into the No. 4 ESS to define the announcement.

The No. 4 ESS MAS software is designed to accommodate two basic methods of recording MAS announcements from sources outside the No. 4 ESS MAS office. Both methods are software controlled and require no intervention by No. 4 ESS personnel during the recording process. These two methods are designated the ONAC/ADS update method and the direct producer update method. A manual update method also exists for recording MAS announcements from within the No. 4 ESS. A report is output to ONAC after each announcement is recorded in the No. 4 ESS, regardless of the method.

3.4.1 Operations Network Administration Center/Announcement Distribution System update method

The ONAC/ADS typically accepts an announcement directly from the producer and records multiple copies of it for distribution to the appropriate No. 4 ESS MAS offices. Recording calls are then automatically placed via dedicated transmission facilities to these No. 4 ESS MAS offices to transmit the announcement audio. The ONAC/ADS compresses the audio and superimposes a pilot tone to ensure the quality of the announcement audio during the recording process. After recording completes, a report is sent to ONAC to indicate a satisfactory update.

3.4.2 Direct producer update method

The direct producer update method provides a means for a producer to directly update MAS announcements to a No. 4 ESS MAS office. A producer can call a predesignated telephone number and record or update an MAS announcement. After successful completion of the recording process, the No. 4 ESS MAS software originates a callback to

the predesignated producer telephone number. When the producer answers the callback, the No. 4 ESS MAS office plays back the MAS announcement. A producer may accept the audio by listening to the entire callback sequence or the producer may reject the recorded announcement by hanging up any time during the playback.

3.4.3 Manual update method

Personnel within the No. 4 ESS MAS office can make an on-site recording from a 51A or 53A test position [in the Trunk Operations Center (TOC)]. An input message must first specify the announcement identity and several other announcement parameters. There is no automatic callback for update verification after the manual update method completes. The announcement is automatically marked verified.

IV. MASS ANNOUNCEMENT SYSTEM SOFTWARE ORGANIZATION

The MAS software package developed in the 4E5 generic of No. 4 ESS closely interfaces with the new MAS-related hardware to control PAS, counting of media stimulated calls, and cut through. This software package is organized and incorporated into most functional areas of the No. 4 ESS. This section describes the design objectives, characteristics, and main capabilities provided by each of the MAS software functional areas.

4.1 Mass Announcement System software design objectives

The overall design objectives for the MAS software were as follows:

(*i*) Compatibility with the existing No. 4 ESS system and environment was necessary. The development of all these new capabilities involved integrating a large and complex software package into an already large and complex software system where resources are becoming scarce.

(*ii*) Hierarchical, modular, and structured programming design was advocated. This has benefited understandability, development, and maintainability.

(*iii*) Reliability was paramount. Defensive checks abound to ensure integrity of the services being offered.

(*iv*) Audio preservation for an indefinite period of time even through office phases and other unusual system disturbances was required.

4.2 Mass Announcement System software characteristics

Mass Announcement System is the largest feature provided since the initial No. 4 ESS development. It is comprised of approximately 100,000 words of operational software and of approximately 50,000

words of maintenance software. Several new data structures have also been created. All of this software is closely tied together and is closely interfaced with the MAS hardware, thus, forming a complex but unified system.

The MAS software is basically organized into functional areas. It relies heavily on subroutines and is generally separated from other software. For example, new call processing code for new MAS related types of calls is packaged as new call processing programs. Existing call processing code, such as final handling treatment of unsuccessful calls, is in modified existing programs.

Mass Announcement System software has to handle various interesting cases. Most MAS functions take many time segments to complete but occur at infrequent intervals, at least relative to the number of calls that the No. 4 ESS handles. For example, an update call stimulates many bursts of MAS functions which could total several minutes duration, but there should *not* be a large number of such simultaneous update calls. Also, interfaces with the MAS hardware involve delays from the time an order is sent until the time MAS completes the function. In general, MAS functions are deferred when the No. 4 ESS system experiences overload.

Announcement data is distributed over several data bases. Translation data contain permanent service order information. Call store structures contain current status information. File store contains a backup of nontransient current announcement status information.

4.3 Mass Announcement System software functional areas

4.3.1 Announcement Handling

Announcement Handling plays a dominant role in providing MAS services. It is a new software functional area that controls MAS announcements and barge-in announcements from the time they are first defined until they are deleted from the No. 4 ESS. It is the primary operational interface with the MAS hardware (sending almost all the operational orders). Announcement Handling also interfaces with almost all other functional areas involved with the MAS feature.

The main functions of Announcement Handling include the following:

(*i*) administrative processing during recording updates and verification callbacks,

(*ii*) duplication processing for MAS announcements,

(*iii*) scheduling of PAS, MSC counting, and cut-through services,

(*iv*) providing call processing with announcement phasing information,

(*v*) maintaining MAS hardware status as it applies to MAS announcements,

(*vi*) providing manual support capabilities,

(*vii*) providing file store backup for nontransient announcement data.

To control and administer MAS announcements, Announcement Handling maintains a central source of current status and bookkeeping information for every MAS announcement in a No. 4 ESS. The primary data structures, which are all new to the No. 4 ESS, include the following:

(*i*) The MAS Announcement Status Table (MSTAT) is a per announcement call store data table. Each entry contains current announcement data (such as announcement state, duplication status, sector identities, and manual control information), as well as translations-like information which is recent changeable (such as announcement start and stop times and cut-through directory number).

(*ii*) The MAS Announcement Phasing Table (MAPT) is another per announcement call store data table. Each entry contains active announcement information used by Call Processing to handle MAS customer calls quickly.

(*iii*) The MAS sector busy/idle map is a call store structure containing current usage status of each of the 80 sectors available for announcement audio storage on each MAS frame in the office.

(*iv*) The MAS Announcement Register (MAR) is a four-word call store data table seized when needed from a pool of MARS. Announcement Handling functions, such as duplication, callback, and MAS order sending, use these registers in two-way linked lists to queue these internal processes (since they may take minutes of time to complete). This enables easy processing of these per announcement functions on a first-in, first-out basis.

In performing its functions, Announcement Handling controls the announcement audio cycle by processing an announcement through its various announcement states. These Announcement Handling functions are described in more detail in the following sections.

4.3.1.1 Recording interfaces. Checks are made to see if a recording call can be accepted. If so, sectors may need to be allocated in the MAS hardware before recording can begin. Announcement Handling selects which submember the announcement should be recorded on. If transmission check failure reports are received from the MAS hardware during an ONAC/ADS recording, Announcement Handling informs Call Processing as to whether or not the recording call should continue. For a given announcement, the first three such call attempts with transmission problems are aborted. The fourth such attempt is recorded in spite of transmission problems, but a report is issued so that manual actions can subsequently be taken to listen to the audio and either accept it or remove it. At completion of recording, several

Announcement Handling processes are started such as announcement duplication and verification callback for direct producer updates.

4.3.1.2 Direct producer callback interfaces. Direct producer callback requests are queued and, when appropriate, Announcement Handling requests Call Processing to initiate an outgoing call to the direct producer. There is an initial delay of about 10 s to allow the direct producer to hang up after recording completes. Thereafter, reattempts are initiated once a minute until the direct producer answers, for a total of four such attempts. For security reasons, a direct producer recording cannot be activated until verification completes.

4.3.1.3 Announcement duplication. When an MAS frame is duplex, announcement audio is recorded onto one MAS submember. Announcement Handling must send orders to the MAS hardware so that announcement audio can be duplicated within the mate submember. Several orders must be sent for each sector involved in a prescribed time order. Except for maintenance update, the MAS hardware can operationally duplicate only one sector at a time from each of the MAS submembers. An announcement duplication request is queued in Announcement Handling as soon as recording completes. The time needed to duplicate an announcement (not including time on the queue) is equal to the defined length of the announcement plus 15 s. Activation can precede duplication if start time occurs before duplication is initiated.

4.3.1.4 Scheduling of PAS announcements, MSC counting, and cut through. Announcements can be recorded and scheduled to start immediately or up to 23 hr in the future. For external recordings, start time must be previously entered via a recent change. (This recent change could specify that the announcement should start immediately. Start times are normally in terms of hours and minutes.) Manual recordings are initiated with an input message which specifies start time. When a recording begins, a start date is determined based on the current value of start time for that announcement. If the start time is less than an hour past the present time, it is assumed that the announcement should start immediately. When start time occurs, Announcement Handling begins the announcement activation process by sending orders to the MAS hardware. When this is accomplished, the announcement state changes from standby to active. This is a gradual process in the MAS hardware and is not complete until all phases of the announcement are playing back.

If there was a previously active version of the announcement when start time for the standby version occurs, an active/standby switch takes place. The standby version goes into activation and the previously active version goes into deactivation.

Announcement stop time scheduling is similar to start time scheduling. A stop date is determined when activation completes based on the current value of stop time for that announcement. When stop time occurs, Announcement Handling begins the announcement deactivation process by sending orders to the MAS hardware. Deactivation is also a gradual process. When deactivation completes, the audio for this version no longer exists.

Announcement Handling also schedules MSC counting and cut-through applications. Whenever start or stop times occur, the appropriate functional area program is invoked.

4.3.1.5 Announcement phase processing. Active announcements are phased in the MAS hardware so that the beginning of an announcement starts every 15 s. The total number of phases an announcement has is directly proportional to its length. The number of phases equals defined length (which must be a multiple of 30 s) divided by 15. The MAS submembers autonomously issue a playback phasing report when the beginning of a phase occurs. Phases occur on alternate units for a duplex system. Announcement Handling processes these phasing reports by marking a call store table with all the information Call Processing needs in order to determine quickly where to connect callers at any given time (which dedicated TSI SPC, port, channel, time of the phasing report, etc.).

4.3.1.6 Mass Announcement System hardware status change processing. Announcement Handling updates current announcement status to reflect the current state of the MAS hardware. The MAS hardware status changes and the corresponding actions which Announcement Handling takes are as follows:

(*i*) *Restoral with audio lost.* The MAS hardware initially restores the first MAS submember with audio lost. This same type of status change also occurs after a duplex MAS failure due to a fault in the MAS hardware. At this restoral time, Announcement Handling issues reports for any audio lost. Announcement Handling sends orders to the MAS hardware to redefine all announcements defined in the No. 4 ESS translations data base. When order sending completes, the MAS hardware is ready to accept recordings. All audio needs to be rerecorded.

(*ii*) *Restoral with audio saved.* This type of a restoral of a MAS submember takes place after a duplex failure of MAS which was caused by a fault in a connecting unit, such as a TSI or PUC. Announcement Handling determines at this time whether any audio was lost due to the duplex failure. Any audio which was previously simplex on the MAS submember still out-of-service is lost and an audio lost report is issued. Otherwise, all normal announcement activities on the restored submember resume. This announcement audio has survived the duplex

failure condition. Announcement phasing reports resume from the simplex MAS submember. Thus, MAS customer calls can resume. Recording and other actions can also now take place on this simplex MAS submember.

(*iii*) *Restoral to duplex.* The second MAS submember to be restored is brought into service via a process called Maintenance Update. In this process all audio and announcement definitions and assignments from the in-service MAS submember are copied to the out-of-service MAS submember. This process takes approximately 2 min. During this time all Announcement Handling activity (except the standby monitor function) is locked out. After this Maintenance Update process completes, all audio is duplex and normal announcement activities can resume on both MAS submembers.

(*iv*) *Removal causing simplex outage.* Two basic interfaces exist with Peripheral Maintenance. (*i*) A conditional removal is one resulting from a diagnostic or a manual request whereby the simplex removal is delayed until all announcement activity on this MAS submember completes so that audio will not be lost. Announcement Handling ensures that all recordings in progress on this MAS submember complete, that customer calls hear at least one play of the longest active announcement on this MAS submember, and that all simplex audio on this MAS submember is duplicated within the other MAS submember. (*ii*) For a forced removal, all activity on this MAS submember is aborted. This includes recording calls and direct producer callbacks. All simplex audio on this MAS submember is lost and corresponding audio lost reports are issued.

(*v*) *Removal causing duplex outage.* (Both MAS submembers fail simultaneously or the second MAS submember fails.) All announcement activities in progress at the time of the failure are aborted. No processes involving the MAS hardware can take place. Announcement status is left unchanged since audio loss is determined at restoral time.

4.3.1.7 Manual support capabilities. Manual intervention is not required to control MAS services except for defining the services, monitoring their current status, and resolving problem situations. Automatic reports are output regarding announcement situations which administrative personnel must be aware of. (The destination of these per announcement reports is assigned at the time an announcement is defined.) Manual announcement override capabilities are provided which are initiated via input messages. These capabilities can be exercised by either ONAC personnel or personnel in the No. 4 ESS. A list of these per announcement capabilities is as follows:

(*i*) Manual update can be done from a 51A or 53A test position.

(*ii*) Standby audio can be listened to from a 51A or 53A test position or from a dedicated trunk to ONAC.

(*iii*) Announcement audio, either standby or active, can be removed.

(*iv*) External updates can be inhibited (and then subsequently be allowed).

(*v*) Activation can be inhibited even though start time arrives.

(*vi*) Update received reports can be inhibited.

(*vii*) A standby announcement can be manually marked verified (for cases where a direct producer callback failure has occurred and the audio has been manually verified).

(*viii*) A standby announcement which has been recorded using the ONAC/ADS method in which the recording was accepted on the fourth attempt in spite of transmission check failures can have the transmission problem indication removed from its current status so that activation scheduling can take place.

(*ix*) Announcement status can be obtained.

(*x*) The number of busy disk sectors per MAS frame can be obtained.

4.3.1.8 File store backup of nontransient data. A Machine Updatable Data System (MUDS) exists to back-up nontransient MAS announcement data on disk. Every time a significant event happens to an announcement, a disk write request is made. Thus, integrity of important, translations-like information, such as cut-through numbers, is provided. This backup information is retrieved after phases in which call store has been cleared, after Audits have detected call store mutilation, and after recent change rollback situations.

The MUDS system also provides a lockout system so that only one process can be working on a given announcement at the same time. This prevents interfering situations.

4.3.2 Call Processing

Call Processing controls the new types of MAS calls, which are as follows:

(*i*) There are three types of audio recording calls for the various update methods available.

- the ONAC/ADS update method,
- direct producer update method, and
- manual update method using 51A or 53A test position. (Control is shared with trunk maintenance.)

(*ii*) Callback for direct producer recording.

(*iii*) Mass Announcement System customer call.

The traditional call register is the basic data structure used for all these MAS calls. Calls involving connections to a dedicated TSI use new data structures called Dedicated TSI Connection Registers (DTCRs).

Another function which Call Processing provides is PUC report

dispensing. These reports originate from the MAS frame and indicate significant announcement events which either Call Procssing or Announcement Handling must act on. Examples of these reports are recording transmission check failures, recording complete, and announcement phasing reports.

4.3.2.1 Call flow for ONAC/ADS update. Figure 7 shows a simplified event flow for the ONAC/ADS recording method. A more detailed system description for a typical ONAC/ADS update call is as follows:

(*i*) The ADS dials a directory number over the dedicated trunk to the No. 4 ESS. This director number is different for each announcement. At the particular No. 4 ESS, the received directory number is recognized as a request to update a particular MAS announcement. The

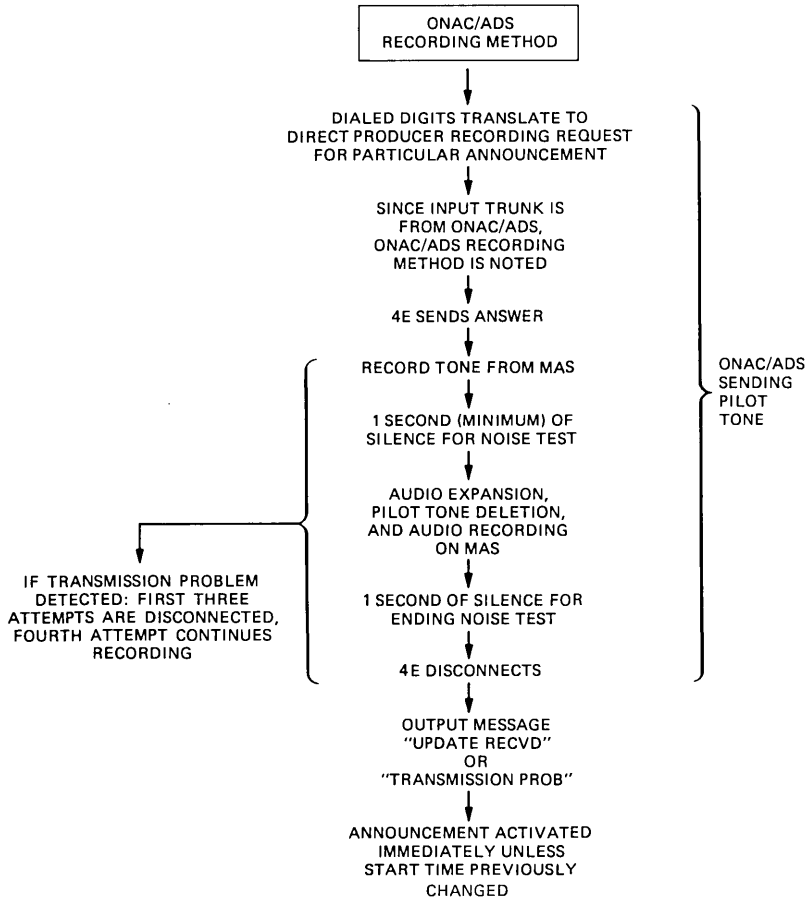


Fig. 7—Simplified event flow for ONAC/ADS recording method.

special identity of the incoming trunk distinguishes this recording call as an ONAC/ADS update call.

(*ii*) ADS starts sending a pilot tone after completing digit transmission. The No. 4 ESS sends an answer signal and hunts for a record port on the appropriate MAS submember. The MAS submember is instructed to begin recording (with transmission checks) after the connection to the record port is established.

(*iii*) The MAS submember autonomously performs the recording function. This includes measuring the pilot tone gain and filtering out the pilot tone, doing an initial noise check, sending the start record tone which alerts ADS to start transmitting announcement audio, expanding the audio material as it is recorded on the disk, monitoring the pilot tone for fading during audio transmission, performing a final noise check when the end of the last allocated disk sector is reached, and generating a recording complete report.

(*iv*) If a transmission check problem is detected at any time during the recording, the MAS submember issues a report. The recording call will be aborted unless this is the fourth consecutive update call attempt encountering a transmission problem for a particular announcement.

(*v*) When the recording completes, the record port is released and the call is disconnected by No. 4 ESS. Announcement audio for the given announcement now exists in the standby state. Announcement duplication and start time scheduling are automatically initiated by the No. 4 ESS.

4.3.2.2 Call flow for direct producer update. Figure 8 shows a simplified event flow for a direct producer update and subseqent callback. A more detailed system description for a typical direct producer update call is as follows:

(*i*) The producer dials a 7- or 10-digit recording update directory number, which is different for each announcement, and is routed to the No. 4 ESS. At the No. 4 ESS, the received directory number is recognized as a request to update a particular MAS announcement.

(*ii*) The call is then connected to a minimum of one cycle of audible ringing and the appropriate MAS submember is selected for the call. At the end of the ringing period there is approximately one-fourth s of silence. At the end of this silent period, answer supervision on non-CAMA (Centralized Automatic Message Accounting) trunks is returned and on CAMA trunks billing is initiated. Call progress tone is connected to the call for a minimum of one-half s while a record port is hunted and reserved.

(*iii*) The MAS submember is instructed to begin recording (without transmission checks). Call progress tone continues until the MAS submember is ready to start recording. The record port is then connected

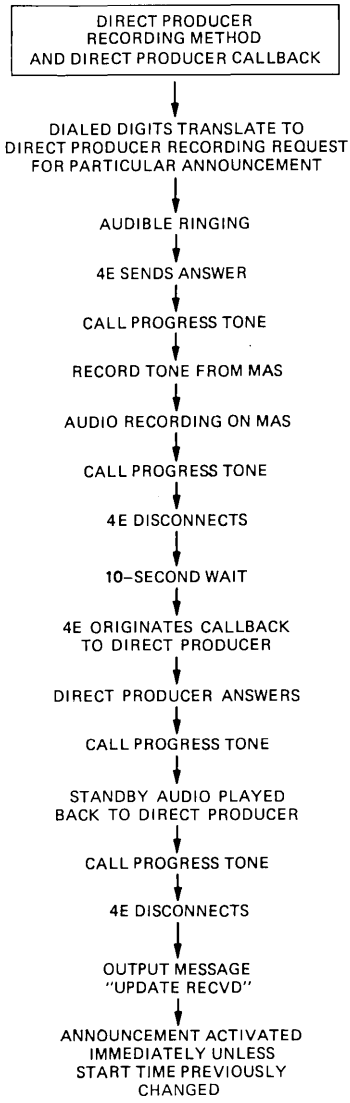


Fig. 8—Simplified event flow for direct producer update and callback.

and the producer hears approximately three-fourths s of record tone and can then begin input of audio material.

(*iv*) The producer must maintain the connection for the allotted announcement length, plus an additional 10 s. When the MAS submember reports that recording is complete (i.e., has reached the end of the last disk sector allocated for recording), the record port is released and

the call is connected to a final 10 s of progress tone, after which the call is disconnected. If the producer does not disconnect first (i.e., before the end of the call progress tone), the recording is successfully completed and announcement audio for the given announcement now exists in the standby state. Announcement duplication and scheduling of the direct producer callback are automatically initiated by No. 4 ESS.

4.3.2.3 Call flow for Manual Update. Figure 9 shows a simplified event flow for a manual recording. A more detailed system description for a typical manual update call is as follows:

(*i*) Within the No. 4 ESS an input message is entered to request a manual update for a particular announcement. The input message also specifies a start time, actual audio length (so that subsequent customer calls to hear the announcement can be removed from the announcement termination promptly), whether or not transmission checks are to be performed, and the identity of the 51*A* or 53*A* test position trunk to be used.

(*ii*) The person making the recording from the test position then hears one of two possible sets of tone sequences. If no transmission checks are specified (which is the normal manual case), the recording scenario follows that described for a direct producer update (after the digit reception process.) If transmission checks are specified, the recording scenario follows the ONAC/ADS update call flow. In this case, pilot tone and compressed audio are expected by the MAS submember.

(*iii*) When recording successfully completes, announcement audio for the specified announcement exists in the standby state. Announcement duplication and start time scheduling are automatically initiated by the No. 4 ESS.

4.3.2.4 Call Flow for direct producer callback. Figure 8 shows a simplified event flow for a direct producer update and callback. A more detailed system description for a typical direct producer callback is as follows:

(*i*) When appropriate, No. 4 ESS originates a call to the producer's callback number and waits for answer supervision to be returned. When steady (greater than 2 s with no switchhook transitions) answer supervision is received, the call is connected to call progress tone. A monitor port on the appropriate MAS submember is hunted and reserved.

(*ii*) The MAS submember is instructed to play back the announcement on the monitor port. At the beginning of the announcement playback, the call is switched from call progress tone to the monitor port to hear the announcement.

(*iii*) After one play of the announcement, the call is connected to a

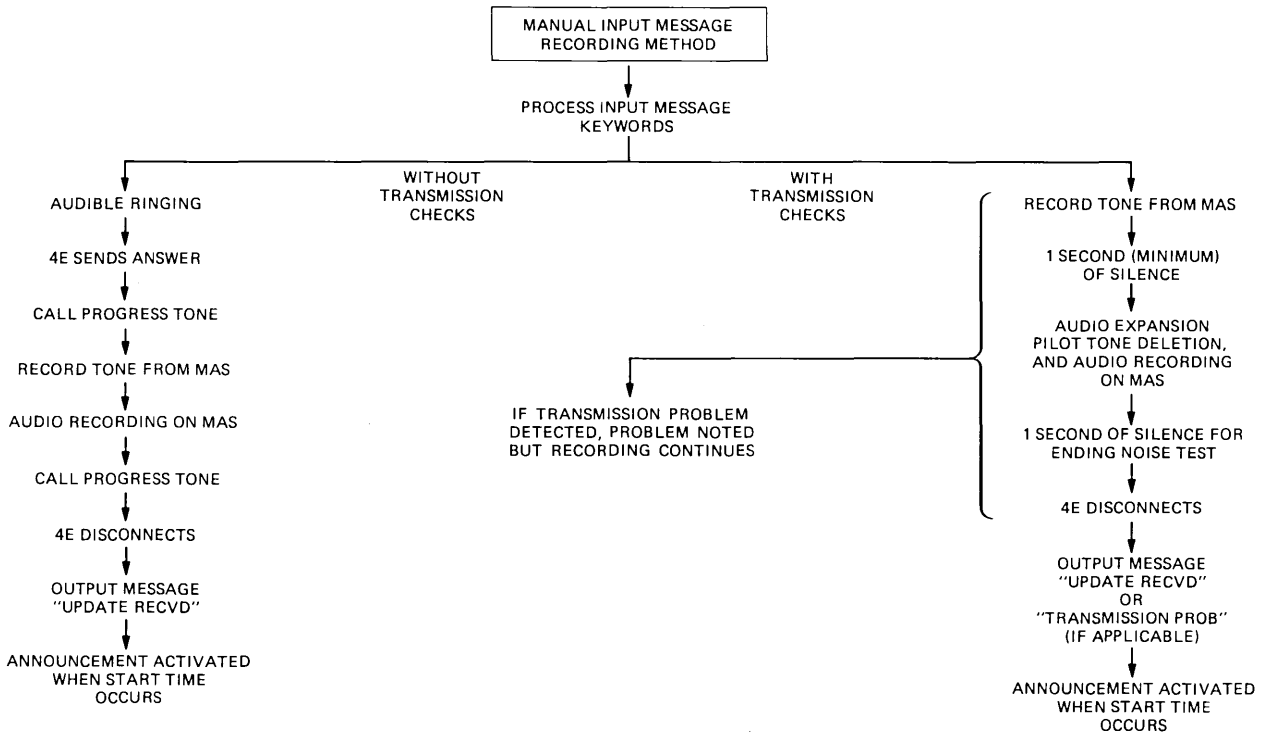


Fig. 9—Simplified event flow for manual recording method.

final 10 s of call progress tone and then disconnected. If the producer does not disconnect first (i.e., before the end of the final call progress tone), the audio update is considered to have been verified and will be scheduled for activation as soon as possible or at the start time given via recent changes.

4.3.2.5 Call Flow for MAS customer call. Figure 10 shows a simplified event flow for a MAS customer call. A more detailed system description of the call flow for a typical customer call to a MAS announcement is as follows:

(*i*) The customer dials a 7- or 10-digit announcement service director number and is routed to a No. 4 ESS MAS office. The No. 4 ESS recognizes the received number as a request to hear a particular MAS announcement. It also recognizes whether MSC counting is associated with the dialed number. Next, it determines whether cut-through service is in effect and, if so, whether the call should be cut through or connected to the announcement. When the dialed number is a MAS vacant code, or the requested announcement service is inactive, the call will be connected to an appropriate special MAS announcement. The preannouncement attempt count is incremented at this point.

(*ii*) The No. 4 ESS determines which dedicated TPI SPC the call should be connected to in order to minimize the time spent listening to audible ringing. When specified, the forced ringing option is taken into account. If all terminations are busy on the first choice dedicated TSI SPC, the call will be connected to the second choice dedicated TSI

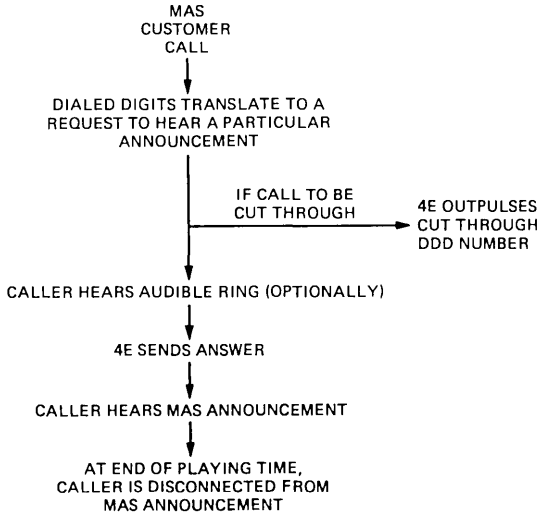


Fig. 10—Simplified event flow for MAS customer call.

SPC (for MAS announcements only) or else to busy tone if no announcement terminations are available. If the MAS equipment should be out-of-service so that no MAS announcement can be given, the call is connected to a no-circuit announcement from the office announcement machine. At the same time, a call is connected to a PAS-dedicated SPC, the appropriate announcement occupancy count is incremented; and the appropriate count in the geographic separations matrix is pegged. This is further explained in Section 4.3.7.

(*iii*) At the end of the audible ringing interval, the call is connected to the announcement via the same dedicated TSI SPC that provided the audible ringing. When the announcement connection is made, the appropriate announcement completion count is incremented. Answer supervision is then initiated (unless specified otherwise) or CAMA billing is initiated (if the incoming trunk is CAMA). If applicable, the MSC dialed number count is incremented after answer supervision is returned.

(*iv*) The customer can abandon at any point during the announcement. In the absence of early abandonment, the call is automatically removed from its MAS termination when the allocated playing time elapses (as determined from the announcement length and specified number of plays). The announcement occupancy is measured up to the point of abandon or forced termination.

4.3.3 Trunk Maintenance

Trunk Maintenance software provides a variety of capabilities. Many of the manual support capabilities were developed in the Trunk Maintenance area to interface with Announcement Handling and will be used by ONAC personnel. These functions are described in the Announcement Handling section and in the Audits and System Integrity section. Other Trunk Maintenance capabilities developed for MAS are as follows:

(*i*) Trunk Maintenance handles the MAS related trunks. It removes and restores the record and monitor trunks corresponding to removals and restorals of the associated MAS submember. Playback trunks are kept nailed up to the dedicated TSI SPC.

(*ii*) The Trunk Operations Center alerting software informs ONAC of duplex MAS failures and restorals.

(*iii*) Trunk Maintenance messages can be requested to include per call information on ineffective attempts for update calls and MAS customer calls. These are useful for resolving problems with unsuccessful update calls and MAS customer calls.

(*iv*) Trunk Maintenance shares control with Call Processing for manual recordings which use the 51A or 53A test position. The trunk maintenance calls for recording and listening to standby audio use the traditional trunk maintenance register (TMR) for per call data.

4.3.4 Peripheral maintenance software

The MAS and PUC frames are two new peripheral units for No. 4 ESS. The peripheral maintenance software functions provided for these new frames are as follows:

(*i*) Bootstrap capability during office phases 2 and higher configures these frames as appropriate to the escalation rate of phases and severity of the problem(s). Duplex configuration of MAS and PUC are preserved if possible.

(*ii*) Input message or Power Control Switch (PCS) functions are manual means which request removal, diagnosis, and restoral of PUC, MAS, and associated hardware units.

(*iii*) Master Control Center (MCC) panel and frame status monitor and display whether or not the PUC, MAS, and associated units are in service.

(*iv*) Diagnostics and routine exercise programs run automatically to ensure the sanity of the MAS and PUC hardware. Trouble location procedures and diagnostics can be requested manually in attempting to resolve problems.

(*v*) Fault recovery routines are executed when faults are detected for any of the PUC, MAS, or associated hardware units. These errors can be reported through *f*-level, interject level, or base level.

Because of the unique design of PUC and MAS frames that use microprocessors, the Peripheral Maintenance software design for these frames reflects significant changes from those used to handle conventional frames in past generics. For example, the new design takes into account the intelligence built in the microprocessors in the implementation of the configuration and recovery actions.

The Peripheral Maintenance software also has special interfaces with Announcement Handling to ensure that audio is not unnecessarily lost. For example, before one of the MAS submembers can be removed from service for diagnostics, sufficient delay time must take place to ensure that all recordings in progress on that MAS submember complete and are operationally duplicated to the other in-service MAS submember and that all other announcement processes complete.

Craft procedures for maintaining the MAS and PUC hardware are different from those for other frames in the office. There may be delays which must be observed when removing, restoring, or diagnosing MAS submembers. Special care must be taken to preserve audio on the MAS disks.

4.3.5 Recent change and verify

Certain MAS entities, such as the basic equipment configuration, must be defined with other office dependent data at generic retrofit time. This includes the assignment of the MAS playback, monitor, and

record channels, auxiliary audible ringing, barge-in playback trunks, and PAS-dedicated TSI SPCs. The nailed-up connections between serving TSI SPCs and dedicated TSI SPCs for MAS frame playback channels and barge-in playback trunks are also exclusively defined at retrofit time. Although individual announcement services can be established initially in the No. 4 ESS, it is expected that the Recent Change system will be the primary means by which this is done.

Recent Change is the system used by the No. 4 ESS to modify the No. 4 ESS translations data base. New capabilities have been added to the Recent Change system to allow the addition, modification, and deletion of PAS announcement and cut-through applications. The MSC counting applications can likewise be added and deleted via recent changes. Modifications have been made to existing recent change system for code grouping to allow directory number translations for MAS announcements and recording updates. Modifications have been made to existing dial-up port recent changes to allow MSC counting results to be output to remote data terminals. Modifications have been made to existing trunk subgroup recent changes to allow geographic separations data to be collected on calls to MAS announcements.

The Verify system is the means to retrieve and display the data residing in the No. 4 ESS translations data base. New capability has been added in conjunction with the new recent changes to retrieve data concerning MAS announcements, MSC counting applications, cut-through applications, and dial-up.

4.3.6 Network management

Network Management software is a direct contributor to cut-through service. The call-gap timer is set to a present time, plus a time interval (gap). The first call to arrive upon expiration of the timer will be forwarded to a prespecified DDD number. The timer is then reset to the present time, plus the gap interval. Mass Announcement System calls which arrive before the timer has expired are connected to a prespecified customized announcement at the No. 4 ESS MAS office. This process is continued until the cut-through service is deactivated. The gap interval also includes an offset interval used to distribute calls over time to prevent bunching of calls from several offices at once. The offset interval also prevents favoring calls from one office.

Other Network Management functions which affect MAS are as follows:

(*i*) Inhibiting reroute control of MAS calls to prevent traffic overflow from one MAS island to another. The data found in the office translations can be used to specify nonreroutable codes.

(*ii*) Gap control is a normal Network Management control available in MAS and non-MAS offices that may be used to meter the amount of

traffic (i.e., selective choke) or to override cut-through values in MAS offices. To provide Network Management capability so as to protect the network during heavy calling periods or during traffic congestion due to facility failures, an override of the service-order specified call gap interval is made available in MAS offices. Intermediate No. 4 ESS offices which are non-MAS offices can apply gap controls also to protect the network. Gap control can be applied on a 3-, 6-, 7- or 10-digit basis. Calls which are not allowed to complete are connected to an emergency or no circuit office announcement as specified by the gap control. Gap control is accomplished via a new control page (CNO8) and the already existing Network Management cathode-ray tube (CRT) display system. The same gap intervals are available as for cut-through applications.

(*iii*) Reports are output to ONAC whenever gap controls are applied, removed, or have their interval changed by the network manager.

(*iv*) In addition to the normal display page capability provided by the Network Management CRT display system, the control page provides inventory displays for MAS cut-through, MAS MSC counting, and MAS announcements. In addition, the capability to select the previous 5, 10, or 15 min worth of data is also available. Mass calling congestion is most likely to accompany the MAS-type of services. Therefore, these displays of MAS data are made available to monitor this type of traffic, determine the source of the problem, and control this traffic if necessary.

4.3.7 Traffic and plant measurements

All MAS-related measurements used to administer and maintain No. 4 ESS MAS offices and the interconnecting network are called traditional measurements. These types of measurements have traditionally provided the data required to measure service volume and quality, detect weak spots in system performance, guide maintenance activities, engineer future equipment additions, and determine the division of revenues. Eighty-one new MAS-related measurements have been provided and are collected on a 15-min basis and stored in the traffic and plant measurement data base.

Geographical survey measurements are contained in a 32 by 32 matrix which provides information about the place of origin of calls to selected MAS announcements. This information may be valuable to the sponsors of some announcements. For example, the information might be used to analyze the impact of advertising in various geographical areas. This geographical survey matrix contains completion peg counts based on incoming trunk subgroup and the announcement called.

4.3.8 Counting of media stimulated calls and dial-up port

The MSC counting service uses two existing capabilities in the No. 4

ESS, namely, CCIS direct signaling and dial-up port. The use of CCIS direct signaling for MSC count transmittal is actually the first application of CCIS direct signaling in the field. The MSC counting service is comprised of two basic functions: slave application processing and master application processing.

4.3.8.1 Slave processing for MSC counting. When start time arrives for an MSC counting slave application and it is activated, all the involved counters are cleared and an indicator is set so that Call Processing pegs counts for each dialed number associated with the slave application.

The counts for a slave application are sent to the master No. 4 ESS machine every 5 min using CCIS direct signaling messages. The specific times for sending these counts are determined by the slave service start time together with the skewing factor. The skewing factor is a number ranging from 0 to 300 s which represents the offset, in seconds, from the start time when the first set of counts is to be sent. Counts are sent every 5 min thereafter. This skewing factor is intended to even the load on the network and on the master No. 4 ESS machine.

Each MSC counting CCIS direct signaling message consists of eight signaling units. Included in this message is the master application number to which the counts are being sent and two four-digit line numbers together with their associated 5-min peg counts. The number of CCIS direct signaling messages required to send all the counts for a given slave application is equal to the number of line numbers associated with the slave application divided by two.

The rate at which these CCIS direct signaling messages are sent is metered so that a maximum of one message per second is sent for a given application. The impact of this number of messages is minimal when compared to the potential CCIS network capacity. This ensures that MSC counting services will not by themselves force a CCIS terminal into congestion.

In cases of CCIS blockage or network overload, returned MSC counting CCIS direct signaling messages are accepted by the slave office. Receipt of such a returned message triggers implementation of a control to meter more stringently the rate at which messages are sent to that destination. When a control is applied, only one message is sent to that destination every 10 s. Such a control applies for 2 min and can be extended if any of these more stringently metered messages are returned. The counts from each of these returned messages are added into the current peg count so that these counts are not lost.

When stop time arrives, counts continue to be sent for five more minutes to ensure that the master No. 4 ESS receives all the counts. If counts could not be sent during this 5-min period after stop time, a report is sent to ONAC listing all the unsent counts.

4.3.8.2 Master processing for MSC counting. Fifteen min before the start time, the master application is initialized. All the associated counters are cleared. If a dial-up connection is not required, a report is output to ONAC stating that the application has started. If a dial-up connection is required for this application and a connection to the same destination is already up, a report is output to ONAC that the application has started and these numbers start to appear on the sponsor's output terminal. If a dial-up connection is required and one is not already up, dial-up port software is used to dial up the ONAC count distributor automatically 15 min prior to the master application start time. Every minute after start time the last four digits of each directory number together with the cumulative count for that directory number are output from the No. 4 ESS. The ONAC count distributor transmits this count information to the sponsor's location(s). This count distributor provides local monitoring for problems and information content and increases reliability.

If a dial-up connection goes down, a report is sent to ONAC and another report to the No. 4 ESS office personnel. Three automatic attempts are made to reestablish the connection, each of these attempts being 2 min apart. Another report is issued as to whether or not the connection could be reestablished. The counts, however, are not cleared in the reestablishment process.

Counts continue to be sent to the sponsor for approximately 8 min. after stop time for the master application. Then a final message is printed to the sponsor stating that the service has stopped.

A number of MSC counting manual capabilities exist as follows:

(*i*) A master application can be initialized manually.

(*ii*) An option exists for manually zeroing counts.

(*iii*) A master application can be stopped if problems arise or if a graceful shutdown is desired for an open ended application. (Once a master application is manually stopped, it will not automatically start again. It must be manually initialized.)

4.3.9 Audits and system integrity

To ensure the integrity of the various data bases involved with MAS, several Audits have been developed. These Audits periodically look for inconsistencies between the contents of MAS

(*i*) administration software structures in call store,

(*ii*) administration software structures backed up in file store,

(*iii*) translations,

(*iv*) firmware data structures.

If an error is found, the corrective action involves reinitializing all data structures pertaining to the involved announcement(s), including those in the MAS hardware. Thus, all audio for the involved announcement(s) is lost.

Special treatment was necessary for System Integrity and especially for Announcement Handling and Peripheral Maintenance to ensure that MAS announcement audio is not lost during office phases up to and including a phase 4. All call store data structures related to MAS announcements are cleared in phases 2, 3, and 4. Certain call store MAS announcement structures are then rebuilt by retrieving backup data from file store. Audio for announcements in transient states is lost. [All MAS audio can be removed during phase 4 by manually depressing a Direct Data Insert (DDI) key on the Master Control Center (MCC) panel and subsequently running a phase with the Modify Recovery Action key set.]

All the new MAS-related calls, described in the call processing section, are taken down in phases 2, 3, and 4. These calls are considered transient because they cannot be rebuilt. (These calls are still connected to a call register (in the case of update calls) or they are connected to a dedicated TSI. In both of these cases, the connection information resides in call store which has been cleared.)

V. SUMMARY

An MAS announcement capability has been developed for No. 4 ESS to take advantage of its position in the DDD network. A large complex software package coupled with two new hardware frames introduced in the 4E5 generic of the No. 4 ESS provides capabilities so that sponsors can offer PAS announcements, counting of media stimulated calls, and cut-through services on a local, regional, or national basis. This development provides high capacity, flexible services that also include network protection aspects. Many of the MAS announcement capabilities are unique capabilities for toll switching offices. The MAS software package, which is comprised of approximately 150,000 words, is organized into functional areas and involves almost all functional areas in the No. 4 ESS.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 60, No. 6, July–August 1981
Printed in U.S.A.

No. 4 ESS:

Mass Announcement Subsystem

By T. W. ANDERSON, J. H. BOBSIN, R. F. COOK, L. GINGERICH, Jr., M. A. MAROUF, and R. J. MILCZAREK

(Manuscript received July 30, 1980)

This paper describes a new hardware subsystem developed to provide mass announcement capabilities for No. 4 ESS. The subsystem records, stores, and plays back recorded announcements for distribution through the No. 4 ESS switching network. Announcements are stored in digital form on a moving-head disk system. Microprocessors are used for control of disks and of interfaces. Duplicated hardware ensures high reliability, and extensive self-testing capability is provided.

I. INTRODUCTION

Provision of the mass announcement capability on the No. 4 Electronic Switching System (ESS) requires the system to record and store voice announcements, and to play them back to large numbers of calling customers. The capability of instantaneously creating multiple copies of an announcement and distributing the copies to many callers is inherent in the design of the No. 4 ESS digital switching network,[1] and is a principal reason for choosing No. 4 ESS as the vehicle for the mass announcement service. Recorded announcement hardware already in No. 4 ESS lacked the capacity and features needed, so a new hardware subsystem was designed.

This paper describes the hardware subsystem portion of the No. 4 ESS mass announcement capability. An overview of the entire capability, including No. 4 ESS processor software and interactions with the telephone network, appears in a companion paper.[2]

Architecture of the new mass announcement subsystem is influenced by the existing No. 4 ESS architecture and interfaces, and the functional

requirements and reliability objectives of the new service. For example, control information between the No. 4 ESS processor and the mass announcement subsystem is carried via the Peripheral Unit Bus (PUB) system which is used as the interface to the switching network and transmission interface equipment. This bus interface requires equipment similar to that used in other No. 4 ESS peripheral hardware frames, such as the Digital Interface (DIF) frame.[3] The simplest interface for voice signals to the all-digital No. 4 ESS switching network is the serial digital pulse code modulation (PCM) encoded format used for internal transmission within the No. 4 ESS network. This format was chosen for the recording and playback interface, and for storing the recorded announcements.

Figure 1 shows the major components of the mass announcement subsystem. Access to the No. 4 ESS processor is via the Peripheral Unit Control (PUC) equipment frame, which provides a standard bus interface plus control circuitry and programs to allow one PUC to serve one or two Mass Announcement System (MAS) frames and, potentially, also to serve additional frames containing features yet to be designed.

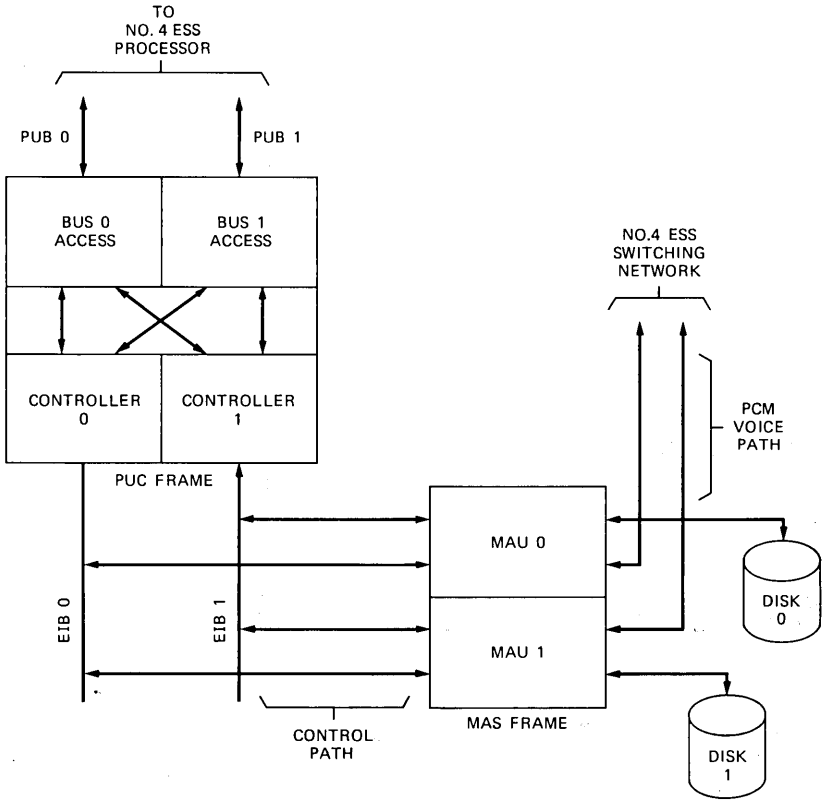


Fig. 1—Mass announcement subsystem bock diagram.

The PUC's full duplex bus interface and duplex controllers allow full service to continue even if an internal PUC unit fails.

Recording and distribution of announcements is performed by the MAS frame (Fig. 1). This frame contains two identical Mass Announcement Units (MAUs). Each has a disk controller and is associated with an 80-Mbyte moving-head disk system for announcement storage. In contrast to the PUC, whose duplex controllers perform identical tasks in synchronism, the two MAS disk controllers operate independently. All announcements are stored on both disks, however, so that if one unit is lost, all announcements remain available. A temporary service requiring longer customer waiting time for an announcement to start is provided in such cases.

In both the PUC and MAS frames, microprocessor systems are used extensively. Both the MAS disk controller, which controls data flow to and from the disks, and the PUC executive controller, which governs PUC internal data movements and the interface to MAS, are high-speed bipolar bit-sliced microprocessors. The PUC also contains a slower *BELLMAC*®-8 microprocessor for background maintenance tasks, and initialization.

II. SYSTEM INTERFACES

A number of external and internal interfaces exist in the mass announcement subsystem (Fig. 1). This results from the structure of No. 4 ESS and from the characteristics of the PUC and MAS frames.

Voice signals for recording and playback of announcements are sent via a coaxial DS-120 high-speed serial PCM data link between each MAS unit and the switching and permuting circuit (SPC), that serves the MAS unit in a Time-Slot Interchange (TSI) frame in the No. 4 ESS network. Each link provides 120 two-way voice and eight maintenance channels. Sixty channels are used for playback, including certain channels reserved as "monitor channels" for verifying announcement integrity after recording. Fourteen channels are used for recording, and certain other channels are used for maintenance purposes. The interface carries no control information other than for timing and synchronizing the link itself.

Control information for the subsystem is carried via the PUB from the No. 4 ESS processor to the PUC frame. This is a 96-bit (total both directions) parallel interface under control of the processor. Frames are addressed via coded enabling bit fields on the bus. A wide variety of operational and maintenance orders destined for the PUC frame and, via the PUC, for the MAS frame are sent over this interface.

Additional external interfaces to the subsystem include dc loop and ac pulse leads from No. 4 ESS signal processor frames. These links are

used to monitor the status of both PUC and MAS frames, and to provide certain subsystem configuration functions. Finally, the PUC receives timing information from TSI frames.

Within the subsystem, the major interface involves the PUC and MAS frames. A bus system used internally in PUC is extended to serve one or two complete MAS frames, and is designed to accommodate additional units in the future. Each PUC controller provides two extended internal buses (EIBS); all MAS units connect to one of the buses from each PUC controller. Each bus contains a 24-bit, parallel two-way data field. Data flow on each bus is under control of its associated PUC controller. Source and destination fields govern the transfer of information in either direction between internal PUC and MAS registers. Additional leads are provided for handshaking and error-control purposes.

III. EQUIPMENT DESIGN

Throughout the mass announcement subsystem, the recently introduced *BELLPAC** packaging system technology is used.[4] The major subsystem components are the PUC frame, the MAS frame, and the disk systems. Figure 2 shows a photograph of the complete subsystem. Both frames use *BELLPAC* packaging system technology, and are 39 inches wide and seven feet high; the PUC frame is 12 inches deep (usual for No. 4 ESS), while the MAS frame depth is 18 inches. The MAS frame depth reflects use of circuit packs also used in the 3B Processor system.

The PUC frame houses one duplicated peripheral controller unit, a duplicated PUB interface, and associated power equipment. A vertical cabling trough divides controllers 0 and 1; the controllers are generally mirror images. The two PUB interfaces are located above the controller units, and principally contain the cable drivers and receivers required to interface to the No. 4 ESS processor. Power supply equipment for each controller unit is located in the lower part of the PUC frame; +140 V input power is converted to +5 and −5.2 V for use within the frame, and +24 V input power is used directly. Power for the PUB interfaces is separate and is derived from converters located in the PUB units. Electronic sequencing, regulation, and overload control is provided on circuit packs located both in controller and PUB units.

The MAS complex contains a single-bay frame and two 80-Mbyte moving-head disk drives (Fig. 2). The two disk drives are located on each side and adjacent to the MAS frame. The frame is equipped with two identical MAUs and associated power equipment. Each MAU is associated with one disk and consists of a controller and circuits interfacing to the PUC, the disks, and the No. 4 ESS switching network.

---

* *BELLPAC* is a trademark of Western Electric.

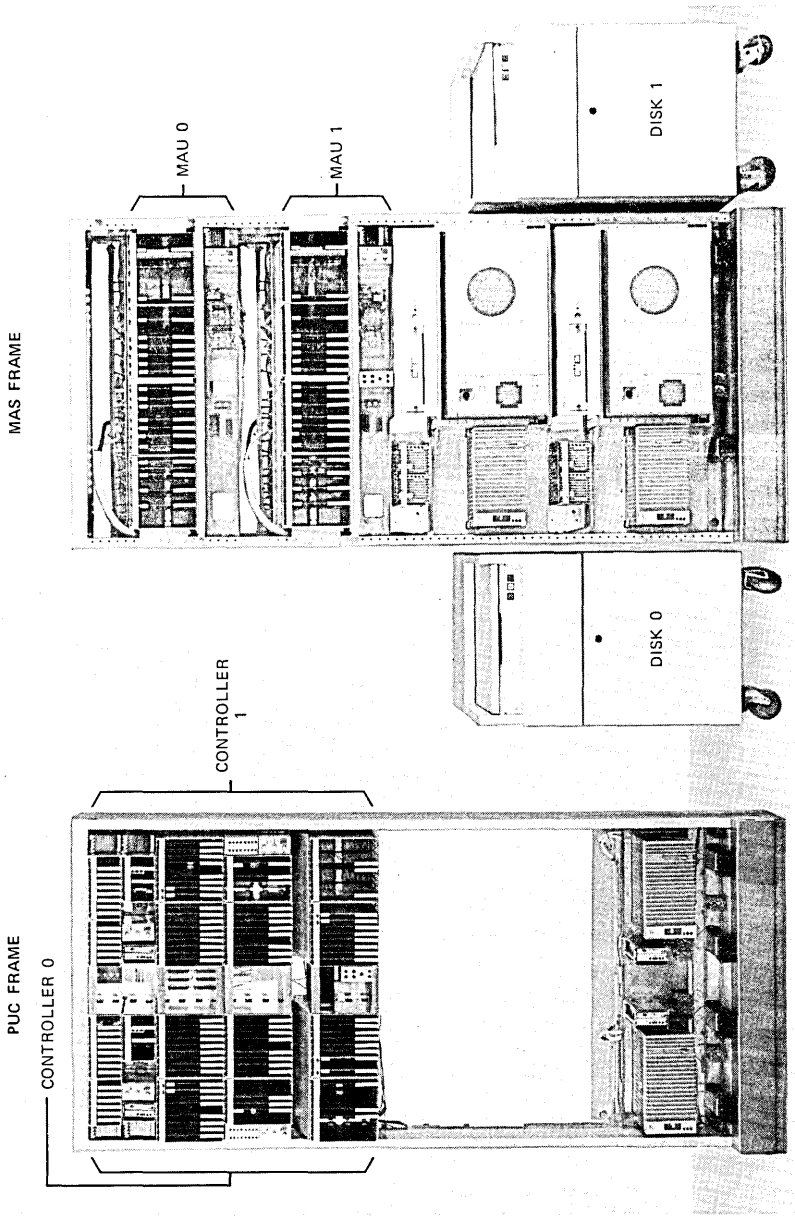


Fig. 2—Mass announcement subsystem frame complex.

The lower half of the MAS frame is used to house the power equipment. To provide autonomous operation of the two MAUs and the associated disk drives, duplicate power feeders of +140 V, +24 V, −48 V dc and 208 V single-phase ac are cabled to the frame from their respective No. 4 ESS office power plants and from the office power

service circuit. The +140 V is converted to +5 V, −5 V, and +12 V dc for use by the MAUs. The −48 V is inverted to 208-V ac for use by the disk drives whenever commercial ac or central office essential ac is interrupted or falls outside the drive's operating limits.

The circuit packs in both the PUC and MAS frames use both Schottky and low-power Schottky transistor-transistor logic, and high-speed emitter-coupled logic contained in dual in-line packages (DIPs). Three different sizes of circuit packs are used; the sizes vary from 4 by 9 in. to 8 by 13 in. Connector pinouts available on these packs number 100 or 200 pins.

The circuit pack technologies used are the double-sided rigid board and the multilayer board with both external and internal power and ground planes. Both types are nominally 0.0625-in. thick.

The double-sided rigid board is an epoxy-glass board with etched copper-printed wiring on both sides. Path widths range from 0.006 in. to 0.050 in., and plated-through holes of 0.020 in. are used. This board is primarily used for low-density circuitry.

The multilayer boards are used in four- and six-layer versions and are used for high DIP packing densities. The four-layer boards use the two external layers for distributing power and ground and for the connector fanout patterns. The two internal layers are assigned to signal routing.

In the six-layer versions, power and ground planes occupy the innermost internal layers, which improves electrical characteristics. Several voltage levels can be provided through segmentation. The two

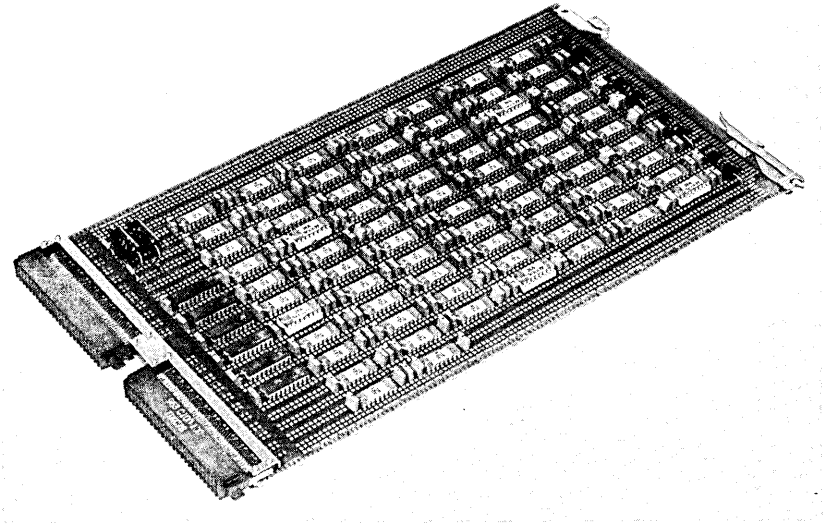


Fig. 3—*BELLPAC*™ packaging system technology TN circuit pack.

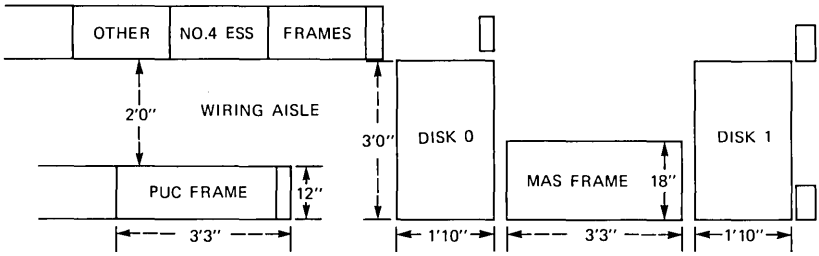


Fig. 4—Mass Announcement System floor plan.

external layers are generally used only for soldering pads and for connector fanout patterns, but can also be used for signal routing, if necessary. The remaining two internal layers are for signal routing. Signal paths on the multilayer boards can be as large as 0.025-in. wide and can be decreased to 0.008 in. where two paths pass between DIP terminals or plated-through holes.

A six-layer 8- by 13-in. circuit pack is shown in Fig. 3. Both power supply filtering and a large number of decoupling capacitors are used on this pack. These capacitors are judiciously placed to minimize noise.

A maximum of eight MAS complexes, each consisting of one MAS frame and two disk drives, may be installed in a No. 4 ESS office. However, to meet reliability objectives, a maximum of only two MAS complexes may be connected to a PUC frame.

A typical central office floor plan for one mass announcement subsystem is shown in Fig. 4. Because of the disk size, the MAS complex consumes the space normally allotted to two frame lineups. The disks also provide a convenient break in the frame lineup so a MAS frame with a depth of 18 in. can be used; most No. 4 ESS frames are 12 in. deep. The PUC and MAS frames are placed in the same lineup to keep the interconnecting cables as short as possible.

IV. PERIPHERAL UNIT CONTROL CIRCUITS

The PUC frame provides a control interface between the No. 4 ESS processor (1A Processor Common Control) and new equipment that must be controlled by the processor. The mass announcement subsystem frame is the first user of the PUC, but the PUC has been designed so that future services can be added easily, with only PUC microprocessor firmware changes required. This eliminates the need to develop a new processor interface for each new hardware system, and saves on the cost and time required to add services to the No. 4 ESS.

Much of the PUC circuitry is similar to the controller of the DIF, which is discussed separately.[3] We review briefly the common portions.

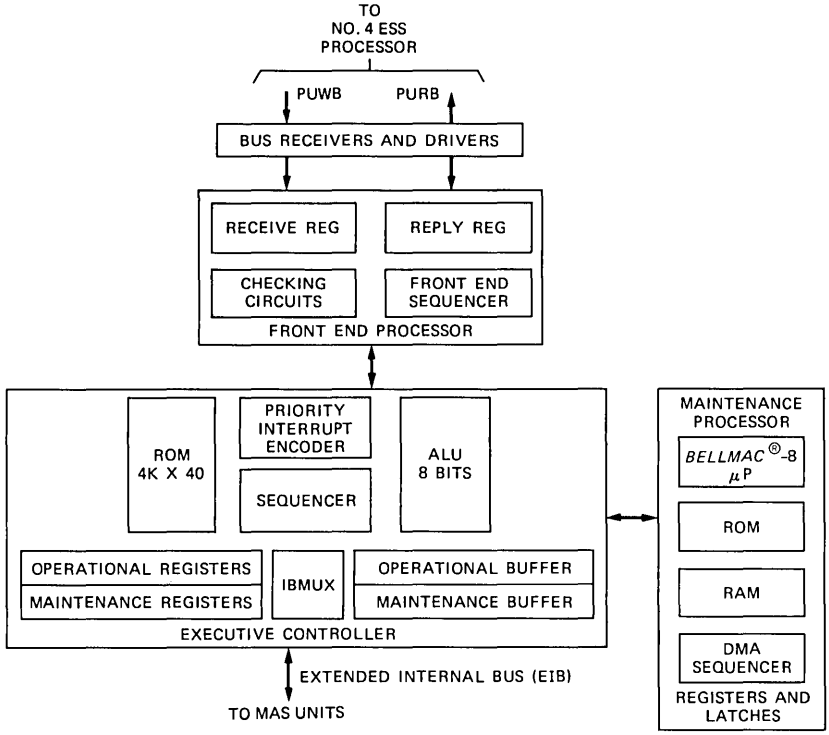


Fig. 5—Peripheral Unit Control frame controller block diagram.

The main unique features of the Peripheral Unit Control circuits include the extended internal bus which connects to the mass announcement frame, and the microprocessor programs or "firmware" that govern the controller's operation.

Each PUC controller centers about an executive controller (Fig. 5), a high-speed bipolar-technology microprocessor which provides processing functions and controls data transfer operations on an internal bus. The executive controller has access to registers and buffer memory, and controls interfaces to the PUB and mass announcement frame. A second maintenance processor can be used for background tasks.

4.1 No. 4 ESS processor interface

The duplicated PUB provides the data and control path between the No. 4 ESS processor and the PUC frame. Figure 1 shows the interface between the No. 4 ESS processor and the PUC frame. The PUB interface provides a fully duplicated communication path between the processor and the PUC frame. Each bus consists of four groups: the enable address bus, the write bus, the reply bus, and the control bus. The enable address and write buses (PUWB) convey instructions from the

processor to the PUC, while data from the PUC is sent to the processor via the reply bus (PURB). Control and maintenance information is transmitted to and from the PUC over the control bus. Each of the duplicated PUC controllers has full two-way access to both buses; bus access routing is controlled by the No. 4 ESS processor through flip flops in the PUC.

Circuits that provide access logic to the bus consist of a receiving register, a reply register, a sequencer, and checking circuits. Orders from the processor to the PUC are sent over the PUWB. The order is latched in a receiving register, and checked for validity and for the correct address code. For valid orders, the hardware generates a high-priority interrupt to the executive controller, which then processes the order. When processing is complete, the results appear in the reply register, usually within 20 μs. The bus access hardware then gates the reply data onto the peripheral unit reply bus. The bus access logic removes significant real-time overhead from the executive controller.

4.2 Executive controller

The executive controller is a microprogrammed bit-sliced processor with a basic cycle rate of 4 MHz. It accepts interrupts from three external sources: from the No. 4 ESS processor, from the MAUs, and from the maintenance processor located within each PUC controller. An interrupt request points to a starting address in the firmware microprogram, a set of routines that route data between different registers and memory locations. The firmware microprogram is contained in 4096 words of read-only memory (ROM); each word is 40 bits wide, of which eight bits are used for parity checking. Processor hardware consists of ROM, sequencer, interrupt control, and arithmetic and logic circuits.

Addressing for the microprogram ROM is by sequencer circuits, which provide for conditional program branching. External interrupts are handled by a 16-level priority interrupt controller, which passes a starting address to the processor when an interrupt is received, and an 8-bit-wide arithmetic and logic unit provides computational power. Certain critical circuits (arithmetic unit, sequencer, and interrupt control) are duplicated within each controller; matchers between the duplicated circuits provide improved fault detection.

4.3 Registers and buffer memory

Each controller in the PUC frame contains a set of internal special-purpose operational and maintenance registers. Operational registers include a status register, reflecting critical configuration and data routing states; receiving and reply registers for incoming and outgoing orders from the No. 4 ESS processor; and a cutoff register used to

isolate MAS units from the PUC. Maintenance registers include error-source registers (ESRs), or error indicators, for hardware failures; an exercise register that creates abnormal conditions to verify the operation of error-detection circuits; and a pest register for disabling individual error indications.

The PUC controller has access to a 24-bit, 256-word random-access memory (RAM). This RAM is logically divided into operational and maintenance buffers, and is principally used as intermediate storage for MAS operational and maintenance reports to be forwarded to the No. 4 ESS processor.

4.4 Internal bus

Data transfer within each PUC controller is via a 24-bit data bus; in addition, 6-bit code fields are provided to select the source and destination of each data transfer. Each possible source register is assigned to one port of a 16-port multiplexer; the source code selects the appropriate register and port. The output of the multiplexer is routed to all possible destination registers; the proper register receives the data in response to the appropriate destination code.

The internal PUC data bus is extended outside the frame to serve MAS and other circuits that may be provided in the future. One bidirectional data bus serves all MAS units connected to each PUC controller. Tristate, dc-coupled cable drivers are provided at each unit; cutoff leads are provided to disable MAS units suspected to be faulty. Additional control leads are provided for handshaking and synchronization of data transfers between PUC and MAS.

4.5 Maintenance processor

A maintenance processor is used for localized diagnostic and fault recovery within the controller. This is a *BELLMAC*-8 single-chip, bus-structured, general-purpose microprocessor, with 60K words of ROM program and 4K of RAM. The maintenance processor is capable of interrupting the executive controller and simulating No. 4 ESS processor orders. Fast, direct access by the No. 4 ESS processor to the maintenance processor RAM is provided. Maintenance processor programs include an operating system, bootstrap routines, maintenance processor diagnostics, and application programs used to initialize the PUC frame.

4.6 Peripheral Unit Control programs

The PUC processor complex is programmed to handle operational and maintenance instructions from the central control and units on the extended bus. The executive controller programs that handle these jobs are organized as a hierarchy of tasks entered from a control

program that services interrupts and controls job scheduling. Since the controller is an interrupt-driven processor, the program services interrupts from the No. 4 ESS processor and MAS interfaces, plus maintenance interrupts, such as errors and real-time clock interrupts, which initiate exercise and audit routines.

4.7 No. 4 ESS processor order-handling programs

Processing No. 4 ESS processor orders that have been detected and validated by bus access circuitry is a principal function of the executive controller. The hardware first generates an interrupt request; if an autonomous or background task is running, the task is interrupted at an appropriate point and the incoming order is processed. The opcode part of the order is used as an index to branch to the program. The remaining parts of the order are used as data or address information to access specific registers or memory, or initiate multiple-operation macro functions. For a simple read register order, the task routine moves data from a register selected by the program into the peripheral bus reply register. Similarly, for a write order, data moves from the peripheral bus receive register to a destination register. When the write order is completed, the reply register is loaded with an order to return an all-seems-well (ASW) acknowledgment to the No. 4 ESS processor. Controller hardware takes over once the reply register is accessed; a hardware sequencer is used to return ASW and data to the No. 4 ESS processor in conformance with bus timing requirements. Execution then returns to the program that had been interrupted, or to an idle routine if no jobs were active when the No. 4 ESS processor order interrupt occurred.

4.8 Mass Announcement System order-processing programs

A unit on the PUC extended bus, such as MAS, initiates communication with the No. 4 ESS processor by loading reports in buffers in the PUC executive controller RAM. Processor orders periodically unload these reports. The reports may be responses to macro tasks previously initiated by the processor, or the unit may initiate reports autonomously due to operational or maintenance conditions. To load a report in the PUC, the unit loads the report type and data in its reply register and signals on a common party-line interrupt request lead. The PUC interrupt-handler routine polls the units on the bus to determine which units require service, and then reads the report data. A task-dispenser routine services each unit that responded by examining the report-type field and branching to a task designed to handle that data. When a task has been handled successfully the controller program resets the unit's reply register and interrupt request. If a report cannot be

handled because a buffer is full, the interrupt request for the unit remains set, and a retry is attempted later.

In the reverse direction, the PUC initiates communication with a peripheral unit such as MAS by checking the receiving register on the communication register pack. If the upper byte of the register is nonzero, an earlier order has not yet been acknowledged by MAS, which services the receiving register every 250 μs. In this case, the order will be reattempted later. If the upper byte is zero, then the PUC sends the order to the receiving register and places a nonzero value in the upper byte of the register. When the MAS controller next checks the upper byte, it unloads the receiving register, and zeros the upper byte.

Two buffers are assigned in PUC for communication from MAS to the No. 4 ESS processor. Operational reports are loaded in a low-priority buffer dedicated to handling single-word reports. (A high-priority buffer, used in certain other No. 4 ESS frames, is not used in PUC/MAS.) A maintenance buffer sends multiword diagnostic raw data and echo reports, which acknowledge all orders sent to MAS. The PUC performs protocol checks on reports received from MAS, and reports irregularities to the No. 4 ESS processor.

4.9 Exercise and sanity programs

To aid in the rapid detection of faults, an exercise program is included in the PUC. This program is entered every 10 ms by an interrupt request generated by the controller clock. The program tests all of the controller logical and arithmetic operations. The ability to access most of the registers is tested by read instructions. The test does not destroy register data and is segmented so it can be interrupted by processor orders within 3 μs. Hardware error detectors are used to verify proper operation. If a failure occurs, ESR bits are set that alert the processor by a peripheral bus maintenance interrupt. The 10-ms interrupt is also used to make a maintenance buffer sanity check; should a multiword report being loaded in the buffer not be completed in a reasonable time, the report is closed so that new reports may be loaded.

4.10 Peripheral Unit Control frame summary

In summary, the PUC contains hardware and microprocessor software to provide an interface between the mass announcement frame and the No. 4 ESS processor. The Peripheral Unit Control frame is designed with flexible interconnections so that equipment to provide new features may be easily added to the No. 4 ESS hardware community in the future.

V. MASS ANNOUNCEMENT SYSTEM FRAME

The MAS frame contains two mass announcement units, each associated with a disk storage system. The units connect to the extended

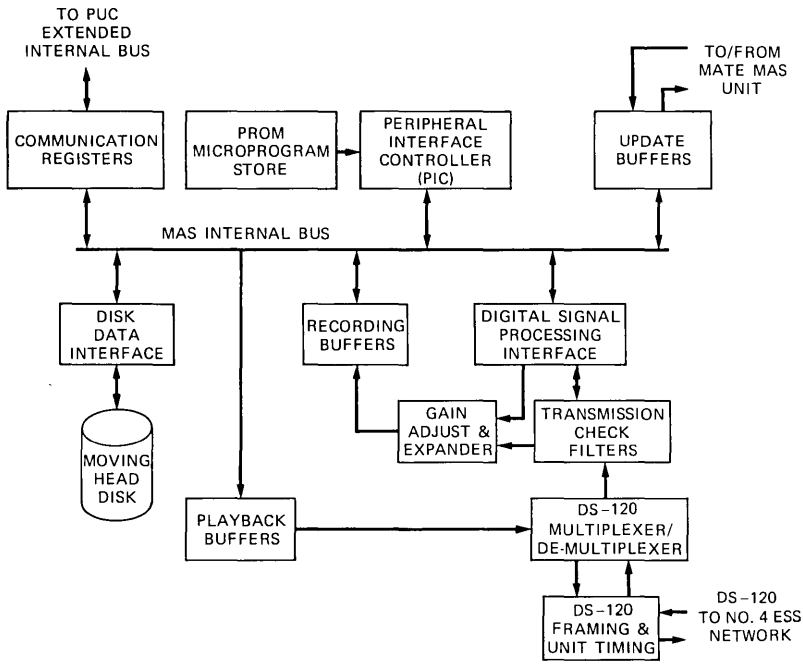


Fig. 6—Mass Announcement System unit block diagram.

internal bus of the PUC for control purposes, and via 120-channel DS-120 links to the switching and permuting circuits of a TSI frame in the No. 4 ESS switching network for recording and playback of recorded announcement signals. Figure 6 contains a block diagram of the major components of a mass announcement unit.

New announcements are placed on the disk by allocating appropriate sectors and designating them as "standby," i.e., not currently playing back. The announcement is then recorded on one disk over a DS-120 link channel via a recording buffer. Coordinating messages are sent from the No. 4 ESS processor to both disks, and the announcement is transferred via an update buffer and a dedicated bus to produce a duplicate copy on the disk of the second announcement unit. Later, under processor control, both MAS units are instructed to change the announcement status from "standby" to "active," and actual playback begins.

5.1 Playback system

Each MAS unit provides announcement playback of 30-s message segments which are read from the disk. Any segment may be assigned to any of the DS-120 link channels dedicated to playback, and segments may be concatenated. Since part of a segment may be silent, the

system has the ability to play messages varying from a few seconds to 5 min in length. Each unit can provide up to 29.5 min total storage of announcements ready for playback.

Although the two units generally store the same announcements, the units are duplicates only in a limited sense. Each MAS unit is essentially a simplex unit playing back announcements independently of the other unit but skewed in time so that when one unit begins the playback of a new cycle of 30-s announcement segments, its mate unit is at the midpoint of its 30-s segment cycle. The No. 4 ESS system connects callers to the announcement unit which first reaches the beginning of the required announcement; the skewing reduces the average waiting time to 7.5 s (Fig. 7).

Each 30-s message segment is stored on the disk units in the 64 kbit/s serial PCM data format used on digital transmission facilities and within the No. 4 ESS network. The storage medium used is an 80-Mbyte moving-head disk system. Each message segment is allocated a three-dimensional portion of the disk storage called a "sector," which

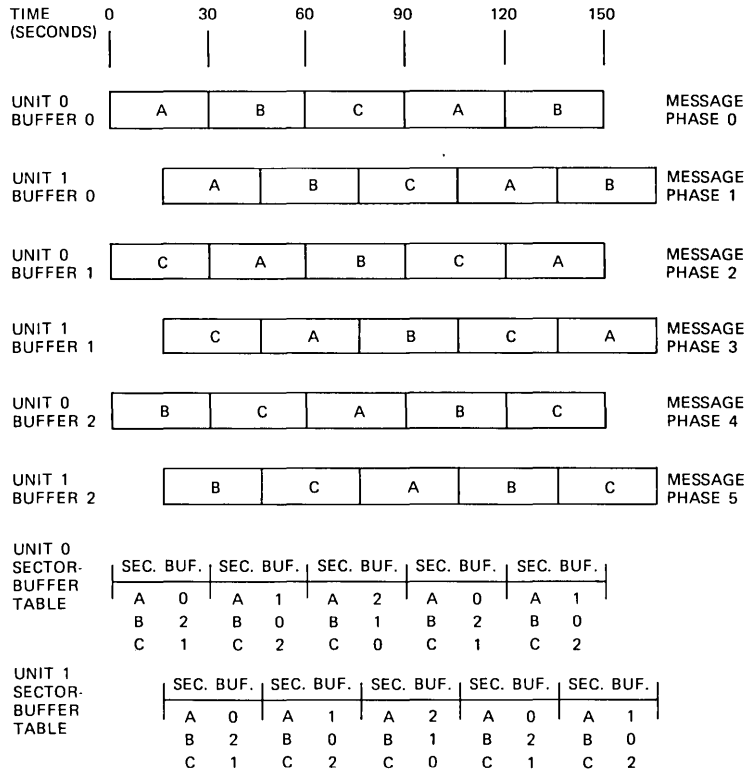


Fig. 7—Message Phase construction. Construction of the six phases of a 90-second message stored on disk sectors A, B, and C.

comprises all the accessible data within an angular portion of the disk. A small section of every active announcement segment is accessed by the disk heads upon each disk revolution.

In operation, the MAS unit reads short pieces of each announcement from the disk into individual playback buffers; all active announcements are served sequentially. The buffers are emptied at a slower rate into the appropriate time slots of the DS-120 link, one 8-bit PCM sample being read out per announcement each 125 μs.

Data read off the disk are converted from serial to parallel form and error correction is performed based on cyclic codes. Parity is generated, and, after an intermediate buffering stage needed because of speed differences, the data are transferred to the playback buffer for delivery to the DS-120 link.

Each of the 64 playback buffers is permanently assigned to one of the 64 even-numbered DS-120 channels 0 to 126. Of these channels/buffers, four are dedicated to maintenance activity and up to 14 others may be optionally designated as monitor channels. Of the remaining 64 channels, 14 are permanently assigned as recording channels, two are maintenance recording channels, two are loop-back channels to the TSI, and the rest are not used.

5.2 Recording and updating system

Announcements may be recorded, activated, and deactivated via dedicated transmission facilities in the telephone network by a centralized administration center (Fig. 8). This function is important in the offering of coordinated, nationwide mass announcement services. The administration center accepts and stores announcement messages from sponsoring telephone companies and commercial advertising sponsors, and handles the distribution of these announcements to MAS frames in No. 4 ESS offices.

Since the recording of announcements on the MAS disk may involve long distances, transmission checks are done at the MAS end. The voice signal is amplitude-compressed at the transmitting end and a pilot tone at 2150 Hz is added so that levels can be monitored. Within the telephone network the signal is converted to PCM digital format, with additional compression according to the $\mu = 255$ law quantization companding standard.

At the MAS end, the PCM data arrive on one of the assigned recording channels of the incoming DS-120 link and are routed through digital signal processing circuitry which first converts both the voice data and the pilot tone back to linear PCM. A noise check on the transmission link is performed before and after the announcement data are received. Digital filtering is used to separate the tone from the voice data, and to adjust the signal level based on the tone level. The signal is also

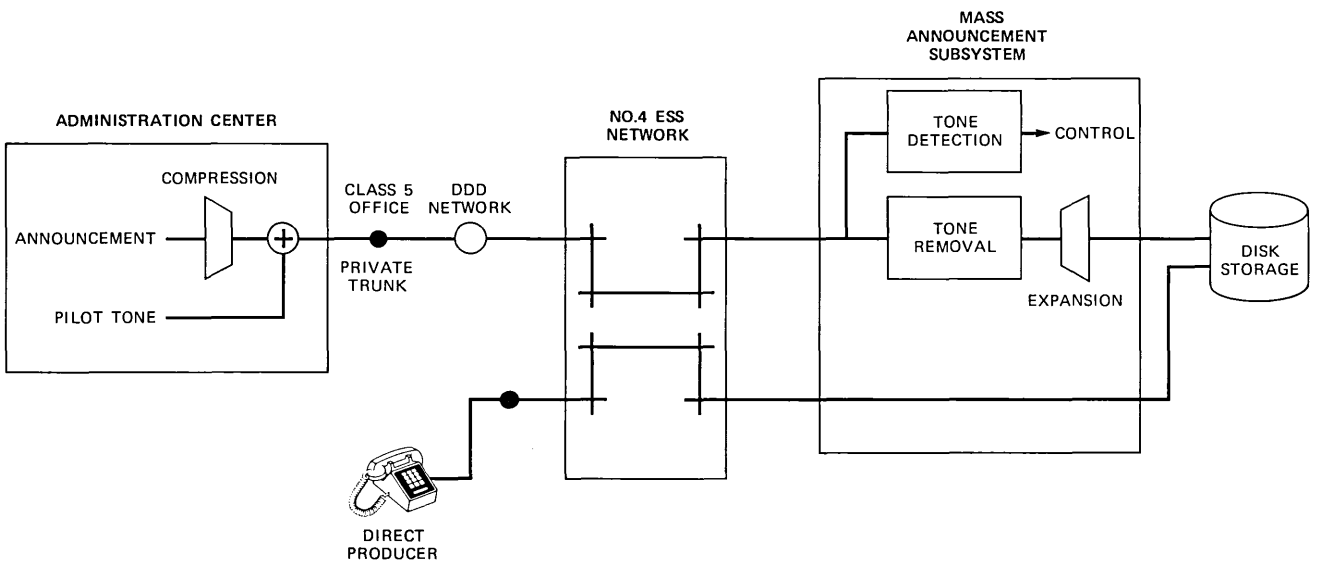


Fig. 8—Direct Distance Dialing network interface.

expanded to remove the compression which was inserted at the administration center. If the tone level is not within specified limits, the network connection is rejected and a retry is requested. The PCM data are finally reconverted to $\mu = 255$ format, changed to parallel form, and delivered to a recording buffer. The data are then written into the assigned disk locations.

Once an announcement has been recorded onto the disk of one MAS unit, a transfer or update of the announcement is scheduled for the other MAS unit. There are dedicated bus paths for updating in either direction between units, with update buffering capable of storing about 16 s of digitized voice at each unit. At the appropriate time, data are transferred in blocks from the originating MAS unit disk through the update buffer to the destination MAS unit disk. This transfer allows the system to provide duplicated announcements with skewed phasing.

Provision has also been made in MAS for local recording of announcements directly, without the assistance of an administrative center. This feature is invoked by placing a telephone call to a special telephone number which results in a connection to an appropriate No. 4 ESS equipped with MAS. A tone is played back to the producer from the recording buffer; this indicates that the desired recorded message should begin. Digital signal processing does not occur in this mode; message quality is confirmed when No. 4 ESS calls back the announcement producer and plays the recorded announcement. The producer decides whether sound quality is acceptable and either approves the recording or repeats the entire process.

5.3 Controller hardware

Central to the operation of MAS is a microprogrammed controller using bit-sliced architecture. This Peripheral Interface Controller (PIC) performs the basic function of data transfer between the moving-head disk and the playback and recording buffers. It also controls execution of orders from the No. 4 ESS processor via the PUC and generates replies. Typical actions include reporting on system and announcement status, updating announcements from one unit to the other, and performing operational and on-request diagnostics.

The PIC is a 16-bit microprocessor-based controller designed using Advanced Micro Devices, Inc., 2900 series bit-sliced integrated circuits. It is capable of a memory-to-memory data move operation in 183 ns. The processor also has 4096 words of 18-bit data RAM, eight priority-encoded interrupts, a sanity timer, a scratch register, and 17 general purpose registers in the arithmetic and logic unit.

The program for the processor is stored in programmable read-only memory (PROM) on three circuit packs. Each instruction is 40 bits

wide, including four parity bits. A total of 4096 instructions can be stored on a single program store circuit pack.

The PIC performs 40-bit program instructions resident in its 12K of PROM in a pipelined fashion for purposes of speed and efficiency; while the present instruction is being executed the next is being fetched by the PIC's sequencer. Each instruction's speed of execution may be set to 183 ns or 366 ns by the program itself. Data is transferred between ports on an internal bus by specifying source and destination fields within program instructions.

The No. 4 ESS processor communicates with the MAS unit via the PUC's extended internal bus. This duplicated 24-data-bit bus is routed sequentially from one MAS to the next, and enables either PUC controller to communicate with either MAS unit.

Orders are sent from the No. 4 ESS processor via the PUC over the extended bus to the MAS register, and reports of MAS activity leave the unit from the reply register. Each MAU interfaces with both buses of the duplicated EIB. The interface consists of two bus driver and receiver circuit packs and a communications register pack. Each bus driver-receiver pack includes in its circuitry an interrupt identification code generator and a bus source and destination decoder. An MAS frame and unit identification code is wired into these two circuits during office installation.

The PUC may monitor the general health of a MAS unit by reading the MAS ESR, part of the communications register. The lowest three bits (EIB parity error, invalid controller activity, and communications register interwrite error) indicate communications failure between MAS and PUC, and result in a peripheral unit failure (F-level) interrupt in the No. 4 ESS processor. The remaining 21 ESR bits indicate problems of less severe nature and, when set, they generate a request to the PUC for service. Three of these bits are directly wired in from the appropriate circuit packs to indicate PIC program memory parity failures, program sanity time-outs, and clock errors. The remaining bits are set by program tests, and include errors such as playback buffer errors, disk control errors, and DS-120 framing and timing errors. The uppermost five ESR bits provide the No. 4 ESS processor with information on ASW failure errors which occur when MAS cannot successfully complete an order from the No. 4 ESS processor. The first bit indicates ASW failure; the remaining four bits form a code that indicates the specific problem that the PIC had in handling the order.

The PUC may place the MAS unit in a particular state by writing the unit's status register, which is bit-writable by the PUC and readable by both the PUC and the PIC. The PUC can place the MAS unit in a maintenance mode or a simplex mode (mate unit out-of-service) by setting appropriate status register bits. An initialization bit forces the

PIC program address to zero and halts execution. Other status bits can be set to mask ESR summaries to the PIC and interrupt requests to the PUC.

5.4 Disk control programs and disk organization

The PIC program's principal function is to act as a disk controller; the majority of the processing power is spent transferring data between the disk and the buffers. Other functions of the program have been designed to fit into the structure determined by the disk accessing tasks.

Efficient playback of announcements from the disks is facilitated by a regular organization of the announcement storage locations on the disk. Since all announcements begin in synchronism, the exact time in each cycle when a given section of an announcement must be read is predictable. The PCM data for all announcements are interleaved so as to minimize the travel of the moving read/write magnetic heads.

Announcement storage is provided on a five-platter removable disk pack controlled by an 80-Mbyte moving-head random-access disk drive. The top and bottom platters serve only to protect the three operational platters. The three operational platters provide five data faces and one clock/servo face. The five data faces are divided into

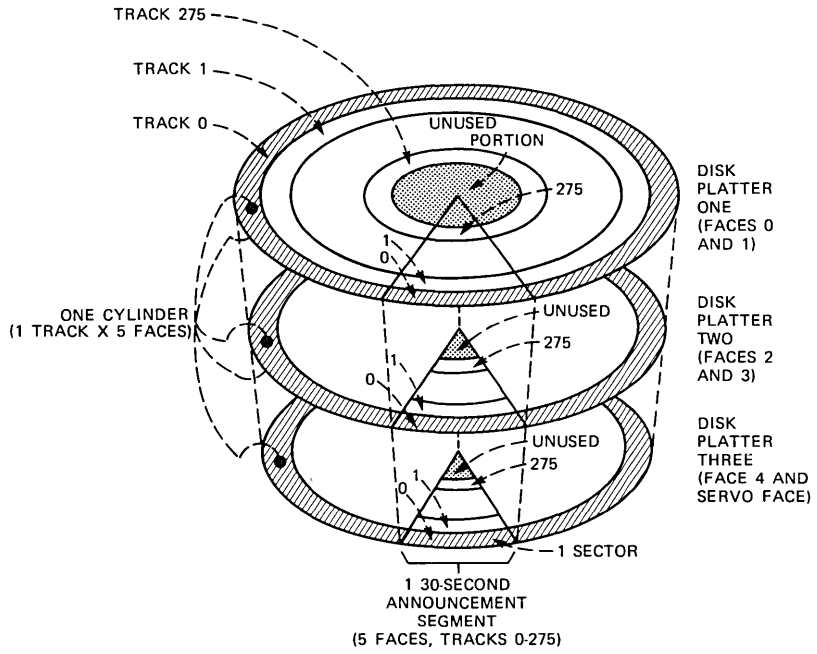


Fig. 9—Disk storage allocation for a 30-second announcement.

aligned annular tracks; 276 tracks on each face are used operationally. Each track is further divided into 86 angular sectors, with the sector boundaries aligned over all tracks and faces. The sets of corresponding tracks across all data faces form 276 cylinders. Each 30-s announcement segment is concentrated within one angular sector, but spread across the tracks and faces within that sector (Fig. 9).

The disk is scanned in 30-s cycles. This is done first by reading all active announcement sectors on the outermost track of each data face in succession to complete the scan of the outermost cylinder, cylinder 0. A small portion of each announcement segment has then been read. Then the heads are moved to cylinder 2; the process is repeated for all even cylinders to cylinder 274. Starting from cylinder 275, the direction of the head motion is reversed to scan the odd-numbered cylinders while the heads return to the outer rim of the disk. The complete process requires 30 s.

5.5 Task scheduling

In addition to controlling disk-head movement and disk-data transfer, the disk-data handler program functions as an executive controller to schedule other tasks. These tasks are scheduled for intervals when disk-data transfer must be suspended for various reasons. During disk-data transfers, the controller is fully occupied with this task.

During the time that the heads are moving from one cylinder to the next or "seeking" the next cylinder, the controller is free to execute other tasks. These tasks are called "seek jobs" and are limited to 9.6 ms in duration. In addition, since the disk-transfer rate into the buffers exceeds the rate at which the buffers are unloaded, the disk accessing must be suspended periodically or "slipped" so that the buffers do not overflow. These suspensions can occur after any track has been scanned, except when a seek is pending. During the time that the disk access is suspended, approximately 9.6 ms, the controller is free to execute other tasks, called "slip tasks." Slip tasks are reserved for self-testing, which are covered later under overall PUC/MAS maintenance. Tasks less than 100-μs long can be executed during idle sectors, which contain no active announcement data. Finally, a period of about 10 μs is available at the beginning of each sector during which no program action is necessary to maintain data flow (Fig. 10).

5.6 Operational and maintenance tasks

Program tasks in MAS that handle communication with the PUC require only a short amount of time but must be executed frequently. These actions are covered by a "preamble job" executed in a 10-μs period near the beginning of each sector. This task is also scheduled at approximately 250-μs intervals by long-duration tasks, such as seek or

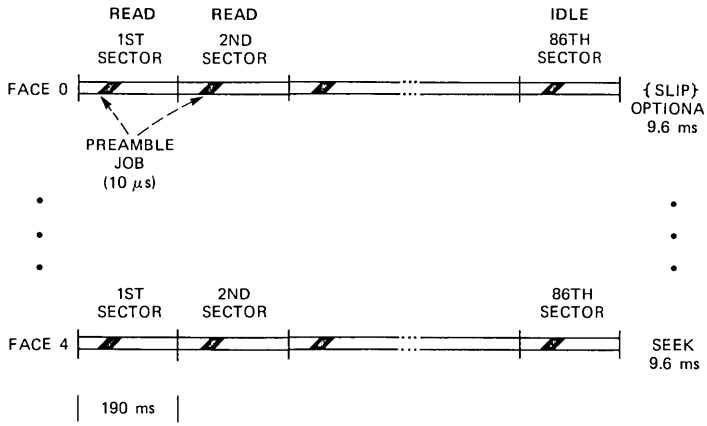


Fig. 10—Typical program flow for one cylinder.

slip jobs. The preamble job unloads the MAS unit receiving register into a queue maintained in PIC RAM, and loads the reply register from a second queue in RAM. Operational and maintenance tasks process the received orders and generate the replies.

All operational tasks are scheduled during seek or idle sector intervals. For example, during every fourth seek the seek task dispenser schedules a message-timing task. This task performs announcement phasing and concatenation of 30-s segments during recording and playback. The message-timing task also processes information in the announcement status, buffer allocation, starting point and sector allocation tables to update the sector buffer table, which governs disk data transfer. In addition, a unique task is selected for each seek interval in the 30-s cycle, and additional "once-per-cylinder" tasks are executed during each seek interval.

Self-test tasks are performed during slip intervals. We cover self-testing later along with overall maintenance of the PUC/MAS subsystem.

Certain short tasks are performed during the 100-μs intervals during idle sectors. Two idle sectors per disk revolution are reserved for scanning the tone and noise detectors of the digital signal processing circuits used in recording. Data are collected which determine the average level and noise values for each recording channel. This is used to provide automatic gain control; should the signal level or noise become unacceptable, the program aborts the recording.

During other idle sectors, the No. 4 ESS processor order-execution routine is called. This routine leads an order out of the receiving register queue and passes control to an order routine determined by a data field in the order. Typical orders involve allocation or deallocation

of sectors and buffers, recording, monitoring, and playback of announcements, and duplication of sectors. These orders change the state of the message tables, thus, allowing the message timing and control routines to properly execute the desired action. Other processor orders can be used to read or write PIC RAM locations, diagnose particular circuits, and start or stop the unit.

5.7 Update tasks

Duplication of announcements between MAS units via the update buffer is another task executed by the PIC. This transfer is performed on a per-sector basis. If a sector is to be updated, a No. 4 ESS processor order alerts the unit that acts as the source. During the cylinder 0 task, the sector buffer table is altered to begin to load the given sector into the update buffer. The processor then instructs the receiving MAS unit to set up its sector buffer table to unload the update buffer into the correct sector on the disk. The receiving unit begins to transfer data from the buffer to its disk 15 s after the source unit begins filling the buffer. Since each unit has an update buffer, duplication can occur in both directions at once.

If one unit is taken out of service, its disk must be updated before it can be restored to service. This maintenance update is initiated by processor order. In this process, the update buffer is used to transfer all of the useful data on the in-service unit to the out-of-service unit. The process begins when the in-service unit accesses cylinder 0. At this time, the in-service unit writes all of its message-timing tables into the update buffer. This data is used by the receiving unit to interpret the announcement data which is written to its disk from the other unit. The in-service disk then begins loading disk data into the update buffer one cylinder at a time; the receiving unit then empties the buffer. The cycle repeats until the update is complete; this requires about 60 s.

VI. MAINTENANCE FEATURES

Dependability and maintainability are important considerations in the design of the PUC and MAS hardware subsystem. These considerations are in line with the high reliability and maintenance objectives of the entire No. 4 ESS switching system. The PUC/MAS maintenance plan is integrated into that of No. 4 ESS. Dependability is achieved by ensuring rapid detection of failures and by providing hardware redundancy that enables acceptable service to continue in the presence of faults. Maintainability requires that maintenance personnel have available automated diagnostic tools to permit rapid isolation and repair of failures.

6.1 Maintenance architecture

Differences in maintenance philosophy exist between the PUC and MAS frames. The Peripheral Unit Control circuit is an interfacing circuit on which several MAS frames and possibly other future services rely. A PUC failure that results in loss of service on all of the connecting equipment is intolerable. To avoid this, the PUC is fully duplicated; its two simplex halves normally run in synchronism, executing identical tasks. Each is fully capable of providing full service should the other fail. The synchronization of the two halves complicates the hardware design but eases the job of fault detection, as matching between the two halves is possible. The MAS frame also consists of two identical halves, or units. As has been noted, however, the two MAS units do not operate in synchronism; they perform similar tasks but at different times. Matching between units is impossible; to aid in fault detection, regular self-testing routines are executed. In the event of failure of one unit, service continues, but longer waiting times are experienced by callers.

6.2 Fault detection

Maintenance actions begin with error detection. In the PUC, matching between controllers, self-checking logic, and internal duplication within controllers are employed to achieve a high level of on-line immediate detection of transient and permanent errors. Protection against data transmission errors is provided within all the controller data paths using coding techniques, loop around, and hardware checkers. All data in memory (RAM/ROM) are coded and checked. Critical portions of the hardware processors (arithmetic and logic unit, maintenance microprocessor, and internal bus multiplexer) are duplicated and matched. A local ESR (hardware monitor) is provided for each major functional unit to allow high resolution of error location. Exercise and pest registers are employed to control and test the hardware monitors. Errors in the controller are summarized in a primary ESR, which, when set, causes a maintenance interrupt to the No. 4 ESS processor. The processor calls fault recovery programs for appropriate actions.

The MAS units use similar fault detection techniques to those used in PUC, except that matching between units is not possible and additional self-testing is required. Hardware faults in MAS result in bits being set on the MAS unit's ESR, which, in turn, immediately sets a PUC ESR bit.

6.3 Mass Announcement System self-test

Since the MAS unit does not run in step with its mate unit, matching cannot be used for operational error detection. This requires that a

sufficient number of audits, checks, and tests must be written into the operational firmware program along with dedicated self-test hardware to detect faults during normal operation of the unit. Parity is used on RAM in the unit to aid in error detection. Also, additional hardware access and looping capability has been provided to allow the firmware to more easily test the various hardware modules.

The firmware self-test tasks are executed during seeks, slips, and certain dedicated idle sectors. Some disk sectors are not used for operational purposes; this allows execution of various miscellaneous tasks to provide self-testing. One such task is a scan of error counters which are decremented by certain operational and data handler routines when an error is found. These counters are loaded with some initial value; a negative count causes an error to be reported. This technique reduces the effect of transient errors and also reduces the load on real-time critical processes. Peripheral Interface Controller RAM is checked by performing access tests on RAM data and address registers, and parity and hash checks over software protected areas. Access tests are also done on all playback and record buffer registers. Maintenance buffers are used to do partial memory testing. The disk and disk hardware is checked by accessing a dedicated idle sector; random data is continuously written, read, and verified on this sector, and every track of that sector is processed in a 60-s period. Disk data itself is protected by a powerful error detecting and correcting cyclic redundancy check code, capable of correcting burst errors of up to 11 bits in length.

A playback and recording loop test is also performed. Data from the playback buffer is looped to a recording buffer and then verified. This loop feature is also used to test the digital signal processing circuits. Various dc levels and tones are looped through these circuits to ensure that the proper filtering action is taken. Since the signal-processing circuits are used in a time-multiplexed fashion, the circuits can be fully tested by using a maintenance time slot at the same time other time slots are being used operationally.

All slip tasks are dedicated to self-test. The slip task dispenser monitors the DS-120 framing circuit and controls update buffer testing. The tests executed include an update buffer register test, a march test on the memory fabric, and a check of the cross-unit update access circuits. Update tests are not executed if update work is in progress.

6.4 Diagnostic software

Diagnostic software is available for both PUC and MAS. It can be used under control of maintenance personnel as an aid in fault isolation. Certain portions or "phases" are invoked automatically before out-of-service hardware can be restored to service. Diagnostic software in

PUC/MAS is, in some cases, executed by the No. 4 ESS processor, and consists of a series of tests which send orders to PUC/MAS and evaluate responses. Other portions are initiated by processor orders but are executed by programs in PUC or MAS.

The PUC is a multiprocessor controller employing a hardwired processor, bit-sliced microprocessor, and a maintenance microprocessor. The diagnostic programs have been developed to suit this multiprocessor structure. The diagnostic software contains No. 4 ESS processor resident diagnostic programs that test the PUC front-end processor logic, power, clocks, the interface between the PUC and the No. 4 ESS processor, and the synchronization between the duplicate PUC controllers; and executive controller resident diagnostic routines that are invoked by No. 4 ESS processor orders. These routines perform tests on the arithmetic and logic unit, the priority interrupt encoder, the microsequencer, and in general on the logic that is directly under control of the executive controller ROM. These test results are passed to the No. 4 ESS processor diagnostic programs for analysis and decisions. Diagnostics are also resident in the PUC maintenance processor. These are executed by the maintenance processor under macro commands from the No. 4 ESS processor. These programs perform localized self-testing on the MP hardware, and also interact with the executive controller to diagnose its hardware. Maintenance processor test results are passed to the No. 4 ESS processor via the maintenance buffer.

The MAS diagnostic is composed, like the PUC diagnostic, of a No. 4 ESS processor resident part and a firmware part. The processor resident part provides all necessary interfaces between the PUC unit and the rest of the ESS system. The diagnostic first checks power and the PUC/MAS interface circuits; these phases are processor resident. The MAS firmware part of the diagnostic contains some tests which run only on specific processor orders. Other portions allow the self-test firmware and hardware to operate for a period of time after which the No. 4 ESS processor resident program checks for accumulated errors. Thus, all self-test routines, checks, and audits are designed to serve as part of the diagnostic, as well as for operational fault detection. In this way, the amount of extra diagnostic code is minimized and failures detected operationally generate useful fault-related data to help repair the unit.

VII. SUMMARY

We have described a No. 4 ESS hardware subsystem that adds a flexible capability of recording and playing back announcement messages. The subsystem is generally under the control of the No. 4 ESS processor, and has several internal microprocessor systems for control and maintenance purposes. Flexible circuits for interfacing the an-

nouncement system to the No. 4 ESS processor allow for the economical future addition of new equipment. The announcements themselves are stored in digital form on moving-head magnetic disk systems; the organization of the stored data is designed with particular care for easy access during announcement playback. System reliability is a major consideration; error detection, self-test, and diagnostic systems are important components of the subsystem.

REFERENCES

1. J. H. Huttenhoff et al., "No. 4 ESS: Peripheral System," B.S.T.J., *56,* No. 7 (September 1977), pp. 1029–55.
2. R. J. Frank et al., "No. 4 ESS: Mass Announcement Capability," B.S.T.J., this issue.
3. K. M. Hoppner, S. F. Panyko, and J. Van Zweden, "No. 4 ESS: Digital Interface Frame," B.S.T.J., this issue.
4. W. L. Harrod and A. G. Lubowe, "The *BELLPAC*™ Modular Electronic Packaging System," B.S.T.J., *58,* No. 10 (December 1979), pp. 2271–88.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 60, No. 6, July–August 1981
Printed in U.S.A.

No. 4 ESS:

Network Clock Synchronization

By R. METZ, E. L. REIBLE, and D. F. WINCHELL

(Manuscript received August 4, 1980)

Accurate clock control is a necessity for direct digital transmission of voiceband data between No. 4 ESS offices. This paper describes synchronization of the No. 4 ESS clock, which consists of phase-locking the clock oscillators to an external frequency reference. The operation performed is a second-order digital phase-lock with time constants of 1.4 hours and 3.12 days. In addition, a fast-start mode is provided with time constants of 2.6 minutes and 8.31 minutes. These characteristics provide tracking of frequency shifts due to daily propagation delay variations on transmission facilities, while filtering out higher-frequency jitter components of the references. They also guarantee stable convergent operation, even with the unlikely worst-case of linear oscillator drift. Implementation consists of a unit containing two matched and synchronous microprocessors and a microprogrammed controller. While the phase detector is in hardware, the microprocessors provide the remainder of the loop. Firmware performs the filtering algorithm, oscillator control, unit diagnostics, as well as extensive defensive operational checks. A great deal of effort is made in both hardware and firmware to ensure the integrity of data written to the oscillators. Finally, experimental results show the unit operation tracking design predictions closely.

I. INTRODUCTION

The No. 4 ESS is a digital toll switching system, whose time-division network routes standard 8-bit PCM signals.[1] Digital interfacing of the network to T-carrier facilities is provided by the Digital Interface Frame (DIF), which converts and concatenates a number of T1 facilities into higher-speed serial bit streams for the switch, and vice versa. Thus, the basic timing for the switching network is the 8-kHz frame

rate typical of T-carrier facilities. Timing for the network is provided by the Network Clock (NCLK) frame, which distributes a 16.384-MHz square-wave pulse train to each of the network frames. The 8-kHz framing information is transmitted as a missing pulse in the 16.384-MHz signal once every 125 μs. Synchronization of the network consists of controlling the frequency of the NCLK such that No. 4 ESS offices that are digitally connected run as close as possible to the same frequency. This is done by phase-locking the clock oscillators to the externally supplied Bell System Reference Frequency (BSRF) or to a T1 line from another No. 4 ESS. Thus, the No. 4 ESS is part of an overall system timekeeping plan consisting of a master-slave hierarchical timing structure.[2] The BSRF is the master timing source, distributed throughout the country to clusters of digitally interconnected No. 4 ESS switches. One switch in each cluster is designated as a master and is phase-locked to the BSRF, while the other switches are slaved in a tree-like structure to the master via timing carried in the digital interconnections.

II. SYNCHRONIZATION REQUIREMENTS

When two No. 4 ESS offices are directly digitally interconnected by a T-carrier facility, data arrives at a DIF at a rate determined by the source NCLK, and is read out of buffers into the network according to the local NCLK. Thus, differences in clock frequencies between the two offices result in DIF buffer overflow or underflow. This is compensated for by the loss or repetition of a frame of data, and is called a slip. The impairment to PCM voice is negligible for slip rates as high as several per second, while the effect on voiceband data is more drastic. Any slip is an undesirable loss of information, and thus, the end-to-end slip rate objective for the Switched Digital Network has been set at one slip in 5 hours.[3] One-half of the objective (one slip in 10 hours) is allocated to digital transmission facilities. The remaining half of the objective is allocated to local digital switching systems. Thus, essentially no slips are allowed for digital toll switching systems.

The drift rate of the No 4 ESS NCLK is somewhat better than one part in 10^{10} per day, which means that oscillator adjustment is necessary.

With the synchronization unit described here, the No. 4 ESS experiences an essentially zero slip rate, and if the No. 4 ESS is forced by trouble to run for two weeks without an external reference, it still experiences less than one slip every 20 hours.

III. NO. 4 ESS NETWORK CLOCK CHARACTERISTICS AND ARCHITECTURE

The source of the 16.384-MHz clock frequency is a set of four double-oven, quartz-crystal, voltage-controlled oscillators, connected in an

analog master-slave phase-locked arrangement.[4] (See Fig. 1.) One oscillator is designated master, and the other three are phase-locked slaves.

Phase error detectors and voting circuits are designed to implicate failing oscillators and reconfigure the master-slave arrangement as necessary. In addition to the analog frequency-control input, each oscillator has a 14-bit digital control, capable of adjusting the frequency a maximum of ± four parts in 10^7, with a nominal least-significant-bit sensitivity of 5×10^{-11}. Finally, the oscillators each provide two phase bits, indicating whether the oscillator, if it is a slave, is lagging or leading the master in phase by more than 0.5 degree.

The Network Clock Synchronization Unit (NCSU) controls the clock frequency via the 14-bit oscillator inputs. While the master is being

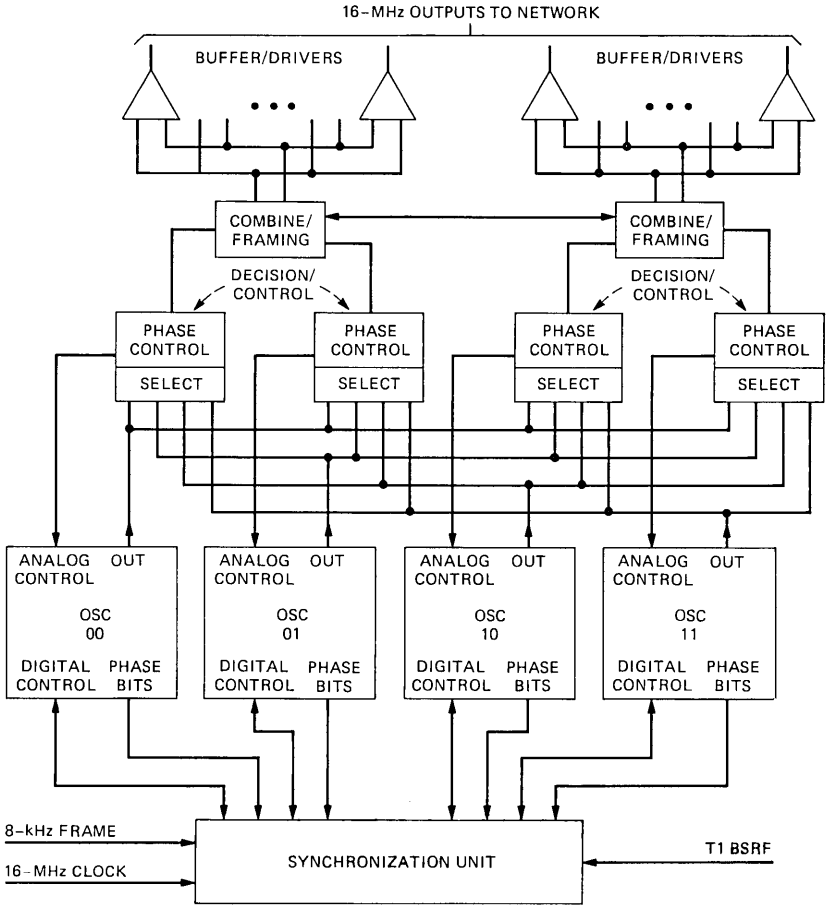


Fig. 1—Network-clock block diagram.

phase-locked to the reference signal, the three slaves are tracked to follow the master with the help of the phase bits. Slave updating algorithms keep the slaves within ±0.5 degree (at 16.384 MHz) of the master.

IV. PHASE-LOCK ALGORITHM AND TIME AND FREQUENCY DOMAIN PERFORMANCE

Since stability in time (or phase) of the 8-kHz frame is what is ultimately important in preventing slips, phase-lock, as opposed to frequency-lock or some other method, was chosen. The phase-lock scheme is similar to what is used in the Digital Data System.[5]

The major components of the loop are the phase comparator, filter section, and the master oscillator. (See Fig. 2a.)

Inputs to the phase comparator are a 4-kHz reference signal, 8-kHz frame pulse, and 4.096-MHz clock, the latter two derived from the local oscillator in No. 4 ESS. A phase comparison consists of starting a counter with the leading edge of the 4-kHz reference signal, and subsequently stopping it with the next 8-kHz frame pulse. The 4.096-MHz clock is counted in between. (See Fig. 2b). Before each count, the counter is preset to −256, thus, yielding a phase comparator output range of −256 to +255 for a given comparison, where 0 corresponds to zero phase error and the exact half-way interspersing of 8-kHz frame pulses and 4-kHz reference pulses. It is evident that a phase comparison is done every 250 μs.

During a typical 8.192-s interval, 2^{15} such phase measurements are made, added, and divided by 2^{15} to yield an average phase error for the 8-s interval. Deviations from this are described later in the operational firmware. This average phase measurement is then multipled by 2^{-15} and added to a running sum called the integral term. Finally, the running sum is added to the average phase measurement, forming a 14-bit frequency control word which is sent to the oscillator.

Since the 8.192-s sampling interval is relatively frequent compared to the time constants of the loop, we can model the system in a continuous form. (See Fig. 2c.) Note that ϕ_R is the phase of the

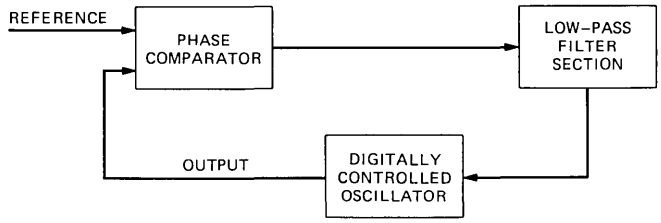


Fig. 2a—Phase-lock loop block diagram.

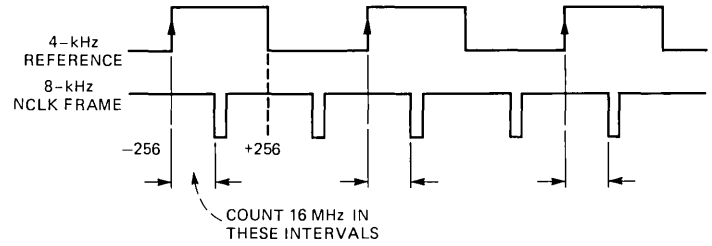


Fig. 2b—Phase-measurement timing.

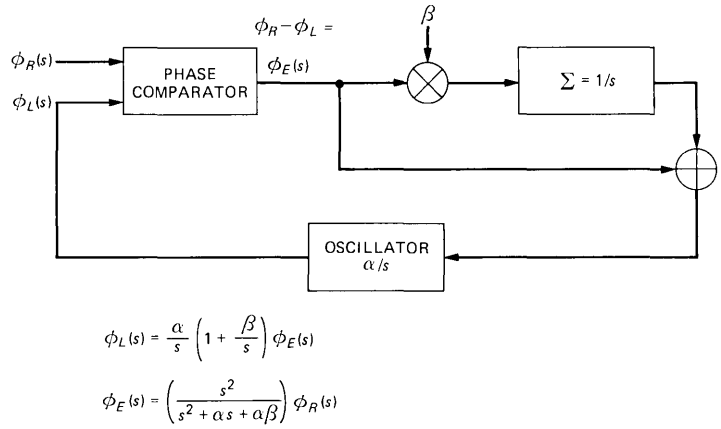


Fig. 2c—Frequency domain model.

reference signal, ϕ_L the local frame pulse phase, and ϕ_E the difference between them. The multiplier in the integral branch is β, and the oscillator sensitivity and phase comparator gain are combined into a common term α. Therefore, the phase of the local frame pulse can be expressed as

$$\phi_L(s) = \frac{1}{s}\alpha\left(1 + \frac{\beta}{s}\right)\phi_E(s), \tag{1}$$

and the phase error is

$$\phi_E(s) = \frac{s^2}{s^2 + \alpha s + \alpha\beta} \cdot \phi_R(s). \tag{2}$$

To determine α, we consider the following: Since the phase comparator has a linear region of 125 μs, corresponding to 512 bits, each bit represents 244 ns. A 1-bit change causes the oscillator to change 5×10^{-11}. Thus

$$\alpha = 5 \times 10^{-11}/244 \text{ ns} = 2.048 \times 10^{-4}/\text{s}. \tag{3}$$

Another way to think of this is in the open-loop sense: Given a 1-bit change in oscillator input, how long will it take for the phase comparator output to change by one bit? It requires 244 ns of phase shift per bit of comparator output, which corresponds to four cycles of 16.384 MHz. Therefore,

$$t = \frac{4}{5 \times 10^{-11} \times 16.384 \times 10^{6}} = 1.36 \text{ h} = \frac{1}{\alpha}. \tag{4}$$

Since the scaling factor at the input of the integral path is 2^{-15}, and updating of the oscillator is done every 8.192 s, β is given by

$$\beta = \frac{2^{-15}}{8.192} \text{ s} = 3.73 \times 10^{-6}/\text{s} \qquad \text{and} \qquad \frac{1}{\beta} = 3.1 \text{ days}, \tag{5}$$

which is the time constant of the integral portion of the phase-locked loop.

A fast-start mode is provided for synchronization under startup conditions. It has the ability to resolve much larger frequency offsets between the clock oscillators and the incoming reference. It is characterized by shortened time constants and a much wider capture range ($\pm$ 4 parts in 10^7),

$$\alpha' = 32\alpha = 6.55 \times 10^{-3}, \qquad 1/\alpha' = 2.6 \text{ min}, \tag{6}$$

$$\beta' = 512\beta = 1.9 \times 10^{-3}, \qquad 1/\beta' = 8.13 \text{ min}. \tag{7}$$

Figure 3 shows the computation algorithm for both fast-start and normal modes.

V. PHASE AND FREQUENCY RESPONSE

Since it is phase drift in the 8-kHz framing of the No. 4 ESS clock that will cause slips on transmission facilities, it is the phase response of the synchronization unit that we are ultimately interested in. Therefore, let us first determine the response to an input step in frequency. The phase error, in time, is given by

$$\phi_E(t) = \frac{\Delta f}{b} e^{-at} \sin(bt), \tag{8}$$

where

$$a = \frac{\alpha}{2}, \qquad b = \left(\alpha\beta - \frac{\alpha^2}{4}\right)^{1/2}.$$

Figure 4a shows that the phase error builds up with a time constant of $1/\alpha$ to a peak of 4.45 kHz/Hz at 19.92 ks and then dies out with a time constant of $1/\beta$.

Correspondingly, the frequency response to a step in frequency is shown in Fig. 4b. The initial error is approximately ΔF, heading

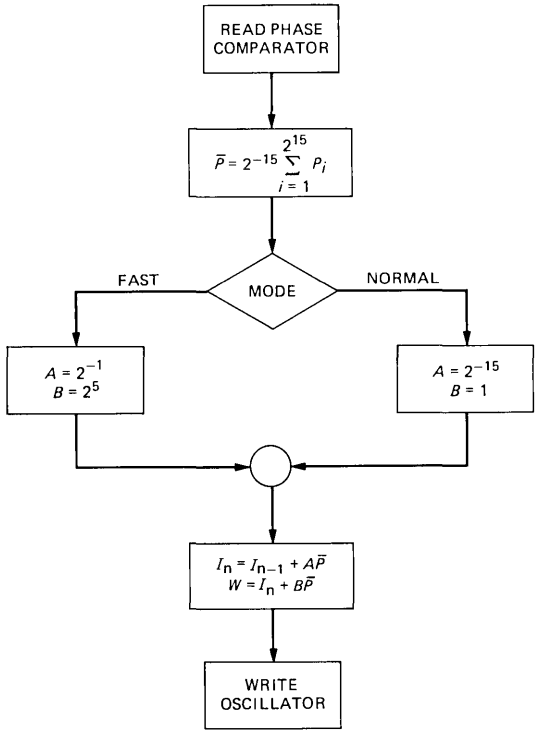


Fig. 3—Synchronization unit—main algorithm.

through zero with time constant $1/\alpha$, and then decaying to zero with time constant $1/\beta$. Note that in the limit, both the frequency and the phase error diminish to zero.

Next, if the input takes a step in phase, the phase-error response is given approximately by

$$\phi_E(t) \simeq \left(-e^{-\alpha t} - \frac{\beta}{\alpha} e^{-\beta t}\right) \cdot \Delta\phi_R. \tag{9}$$

As shown in Fig. 5a, the initial offset is of course $\Delta\phi_R$, decaying predominantly with the $1/\alpha$ time constant. The effect on frequency is shown in Fig. 5b.

Finally, we can consider what happens if there is a linear frequency drift in the input signal (or the oscillator output). Although this linear drift is an unlikely situation, it forms a worst-case bound on what the oscillator or the references can do, short of a faulty condition. For a drifting input, moving at ΔF Hz/s^2,

$$\phi_E(t) \simeq \frac{2}{\alpha\beta} + \frac{2}{(\alpha - \beta)} e^{-\alpha t} - \frac{2}{\beta(\alpha - \beta)} e^{-\beta t}. \tag{10}$$

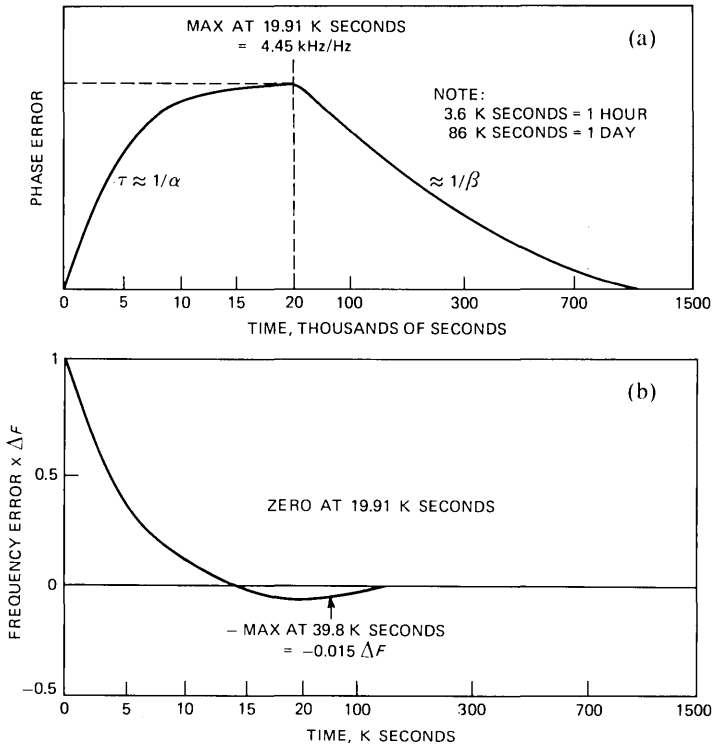


Fig. 4—(a) Phase response to input step in frequency. (b) Frequency response to input step in frequency.

Note that at zero time, the phase error starts out at zero, and in time builds up to a constant offset of $2/\alpha\beta$. This is important, since it determines the maximum phase error for a given drift characteristic of the oscillator. For example, if the worst-case drift is 1×10^{-10} per day, the constant long-term phase error would be about 3 μs, or 12 bits from the phase comparator. This is tolerable, indicating that in the worst-case, the No. 4 ESS frame might shift a constant amount of 1/40 of a frame, and in no case will phase continuously drift.

VI. SYNCHRONIZATION UNIT HARDWARE DESIGN

Figure 6 is a block diagram of the synchronization unit hardware. The heart of the system is a pair of microprocessors with a fully duplicated Read Only Memory (ROM), Random Access Memory (RAM), I/O, and interrupt complex. A bit-sliced microprogrammed controller has Direct Memory Access (DMA) and interrupt access to the microprocessor community, and performs unit control functions, as well as provides a duplicated interface to the No. 4 ESS Central Control (CC) processor via the peripheral bus. Small microcoded programs in the

controller respond to 76 different diagnostic and control orders from the CC by performing microprocessor DMAs and interrupts, data routing, register loading, etc., as required. Three orders are used by the controller for self-diagnosis, one exercising the instruction set, stack, and program counter, the other two testing the program parity, and parity check circuits. In addition to a conventional check of good program parity, one of these two orders increments through a program section where every line has bad parity in a vector form that completely tests the parity tree and check circuits. Parity should fail for every line of this test.

During phase-lock operation, the phase comparator circuit generates an output every 250 μs, representing the relative phase of the incoming reference and local frame. At initial startup, and after a phase "hit," the phase may be arbitrarily adjusted, under program control, by two special phase build-out circuits, one for T1, the other for BSRF. This maximizes the capture range of the loop, and minimizes phase excursions at startup and after reference hits or losses.

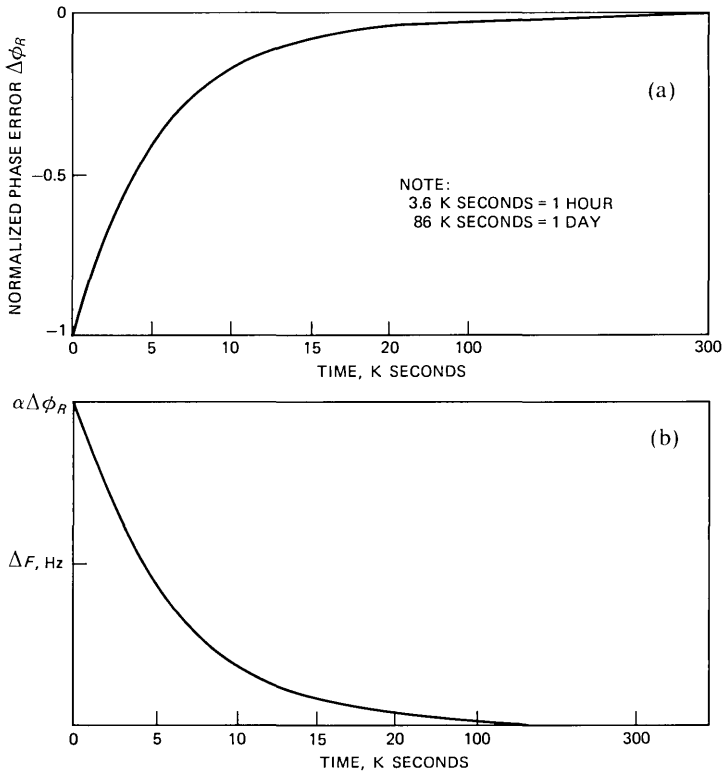


Fig. 5—(a) Phase-error response to step in phase. (b) Frequency response to step in phase.

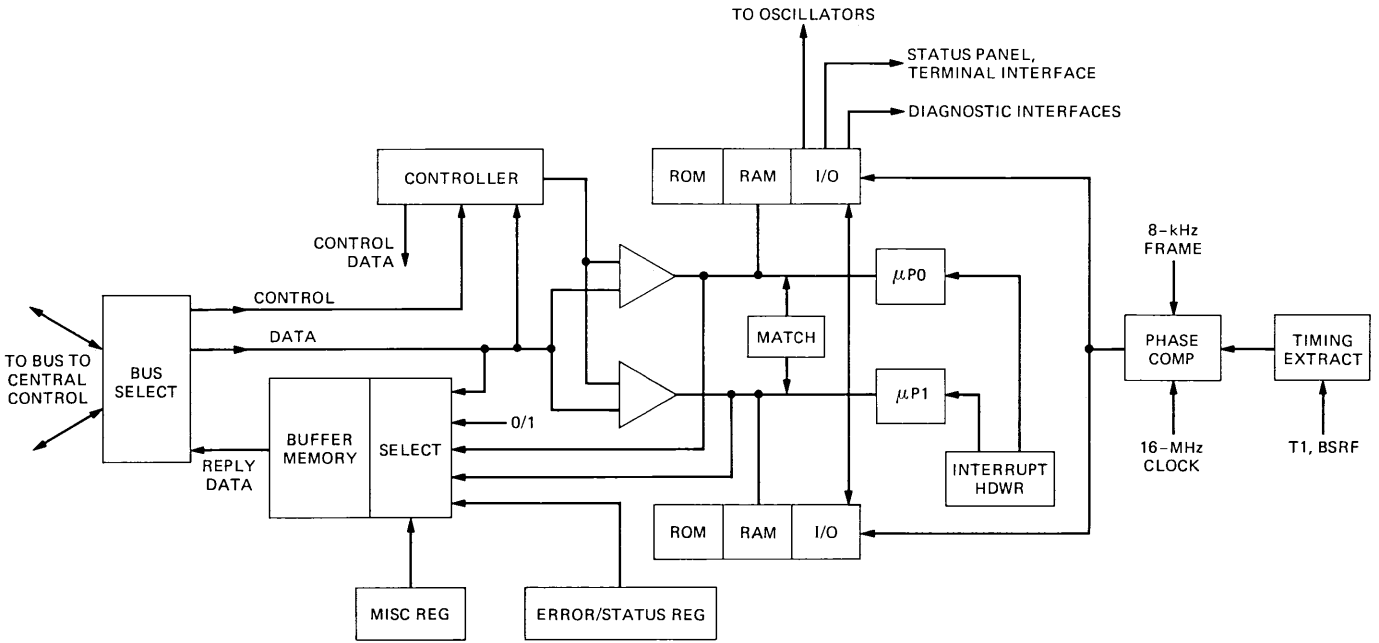


Fig. 6—Synchronization-unit hardware block diagram.

The operational program then reads the 250-μs comparison samples, and performs the algorithm described before. Oscillators are updated every 8.192 s nominally, plus computational and administrative overhead.

It is extremely important that bad data should never be written to an oscillator. Several levels of hardware and firmware protection are employed to ensure the quality of oscillator writes. The first and most important is the duplication of the microprocessors, including 32 K of program ROM, 4 K RAM, 51 I/O ports, and the interrupt hardware. The processors run synchronously and parity over address and data is matched between the two. A mismatch inhibits writing to the oscillators, interrupts and halts the processors, and sets an error in the error register. To further protect the data sent to the oscillators, I/O ports containing oscillator data are cross-coupled: That is, when a processor writes a port and then reads it back, it reads back that port from the other processor. Thus, each processor can match its data against the other, and a write is performed only if the data is the same. Even then, a write to an oscillator will succeed only if it is performed exactly simultaneously by both processors. After the write, the data is read back from the oscillator and is once again verified against what was calculated.

The cross-coupled I/O ports are also used by the diagnostics to verify programs and data stored in ROM. Each processor can read out its own ROM and exchange it with the other via these ports and, thus, check the two for equality.

Another hardware feature that aids in diagnosing the processors is a pair of address cross-coupled read/write I/O ports. When microprocessor 0 writes ports A and B, and then reads them back, it reads A and B, respectively. Processor 1, however reads them back as B and A, respectively, effectively doing an address interchange of data. A section of diagnostic code uses these ports to create differential code in RAM, which is subsequently executed to exercise and check the bus parity match circuits.

To monitor performance of the unit, a status panel provides a readout of the current phase error and integral term, as well as indications of the operating mode and reference in use. An EIA compatible interface on the status panel also allows monitoring of the synchronization process by a terminal. This terminal interface, in conjunction with about 5 K bytes of the program, provides various monitoring modes, as well as firmware utilities, described later.

VII. FIRMWARE

The synchronization unit microprocessor-based firmware is comprised of three parts: Operational programs, diagnostics, and utilities.

The purpose of the operational program is to implement the phase-lock algorithm. The diagnostic portion performs tests on the hardware. The utilities provide the tools necessary for program development. In the following sections, these three areas will be described in detail. A hierarchy chart appears in Fig. 7, and a state diagram in Fig. 8.

7.1 Operational firmware

As shown in Fig. 7, the operational program consists of two parts, the "main loop" and the interrupt handlers. The main loop is where the unit usually resides, and is where the phase-lock algorithm is performed. The interrupts are the means by which the main loop is entered, or they change the operating environment, or they perform various other tasks related to synchronization.

A state diagram is shown in Fig. 8. The "phase-lock mode" state is where the main loop is executed. The processors are in a "halted" state after reset and trap events. In the "free run" state the oscillators are not written. Certain events (like interrupts) cause a transition from one state to another.

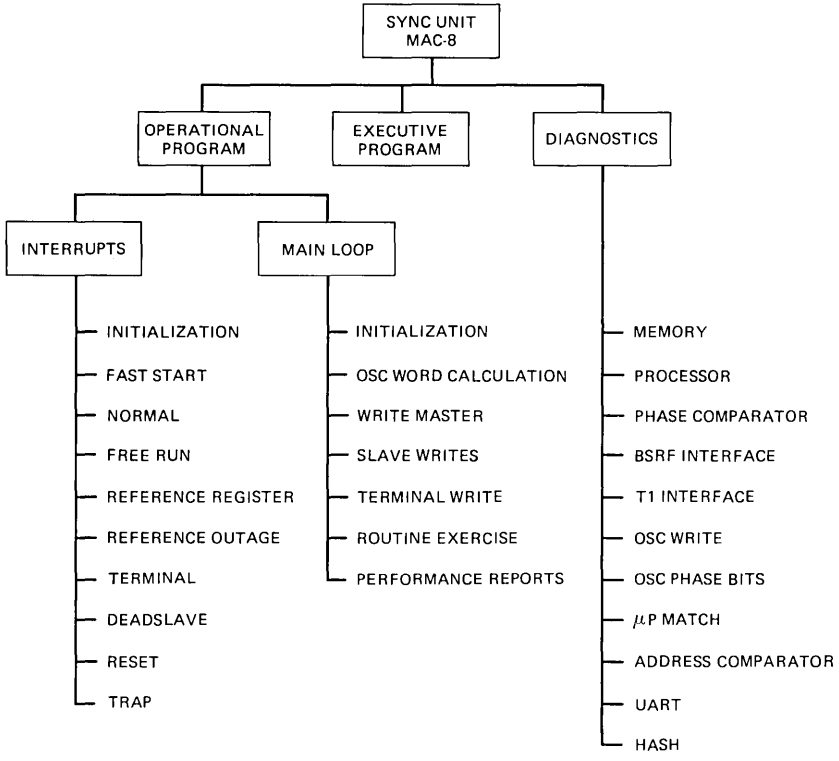


Fig. 7—Firmware structure.

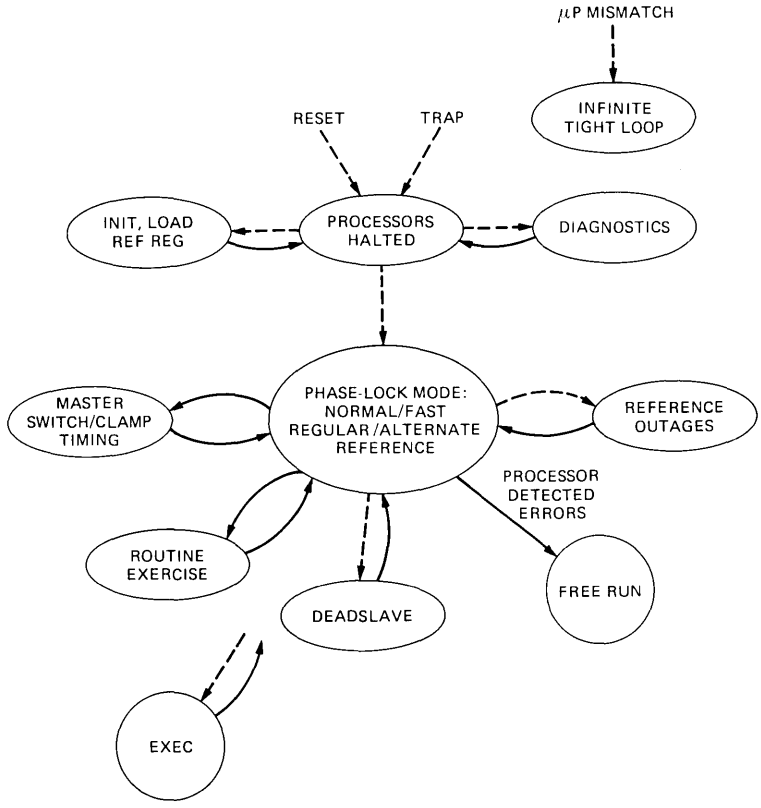


Fig. 8—Processor states.

7.1.1 Main loop

The main loop is an 8-s cycle that performs the synchronization algorithm in Fig. 3. In this loop, a phase measurement is made, the master written, and the slaves written. When the synchronization unit is in fast start or normal mode, this loop is executed repeatedly. There are seven main parts to the cycle (Fig. 7).

The first part of the main loop consists of the miscellaneous tasks. One such task is the maintenance of event counters such as:

(*i*) The number of 8-s cycles executed since the beginning of fast start.

(*ii*) Time elapsed since the last power up.

(*iii*) The duration of reference outages.

Also, the I/O is read for the master oscillator number. The RAM is read in a special location called the reference register to see whether BSRF or T1 is the reference signal to use, and bits in the I/O are written to select that signal. An I/O port is read to find out which references

have outages and another I/O port is written to screen irrelevant or unwanted interrupts. The integral terms and oscillator words are copied into a special area of RAM which is read by the CC and put into the "critical registers" for the synchronization unit.

The next part in the process is the calculation of the master's new 14-bit control word. This program first decides whether to call a routine to perform a phase build out. (The shifting of the reference signal to produce a desired phase error.) For the beginning of fast start, it is necessary to build out to a phase error of zero, whereas after a phase hit we build out to the previous phase measurement. T1 and BSRF build out are performed by different hardware. From the firmware's point of view, the build out algorithm is as follows:

(*i*) Write the build-out register (T1 or BSRF) to 0 and take a phase measurement, A.

(*ii*) If B is the desired phase error, then write the build-out register to C, where

$$C = \frac{A - B}{16} + 32, \qquad A < B,$$

$$C = \frac{A - B}{16}, \qquad A > B.$$

The accuracy of this procedure is ±9 bits of phase error. For the BSRF, the accuracy is enhanced to ±3 bits by using a diagnostic access port to cut off the BSRF signal for 13 μs, which causes a shift in the reference signal as it appears at the phase comparator. This is done repeatedly until the desired accuracy is achieved. The process never requires more than seven iterations.

The next step in the process is taking a phase measurement. The phase comparator comes up with a phase measurement every 250 μs. The phase error takes on the values between −255 and +255, and appears in a 16-bit I/O port. A firmware routine forms the sum of 2^{12} consecutive phase measurements (takes 1 s) and then computes the average phase error. During the process, adjacent 250-μs phase measurements are checked for jitter by requiring that they differ by no more than two bits. If they do differ by more than two, then the event is recorded and the summing routine starts over. Starting over too often results in an error reported to CC.

The time spent updating slaves can consume a considerable portion of the 8-s cycles, therefore, the phase measurement time is varied to compensate by subtracting the time spent updating slaves in the last cycle from 8 s. The resulting time dictates the number of consecutive 1-s phase measurements which are made, which varies from 1 to 7. The average phase error is then computed from the 1-s averages. The difference between adjacent 1-s phase measurements is monitored,

and, if it exceeds three, then a phase hit has occurred, and a build out to the last good phase measurement is performed.

The 8-s phase measurements are monitored from cycle to cylce. The decision to transfer from fast start to normal mode is made based on the magnitude of the phase error (must be ≤ 1) and the slope of the phase error (must be $\leq 1/80$ bit/s). In normal mode, if the slope of the phase error curve exceeds 164/40 or 164/400 or 164/4000 bits/s, then an error is reported to CC and the unit enters a free run state. If the magnitude of the phase error exceeds 230 at any time, then an error is reported and the free run state entered.

Having gone through the checks in phase error, the next step is to update the integral term (see Fig. 3). The new integral term is equal to the old one plus a constant, A, multiplied by the phase error. The master's word calculation follows as the integral term, plus another constant, B, multiplied by the phase error. The integral term, a 29-bit quantity, can be thought of as the 3-day average of the oscillator word, a 14-bit quantity.

Before writing the master with the just-calculated word, a basic check is made for a reasonable value. If the word presently in the master differs from the new word by more than 256 bits (4096 for fast start) then processor-detected error is set (a bit in the I/O which results in an interrupt to the CC) and free-run mode is entered. If the difference is greater than 64, then the master is walked by steps of 64 to the desired value, with the slaves updated at each stage. This prevents NCLK frame detected phase errors from occurring.

Anytime an oscillator is written, a common routine is called which performs some defensive checks and then writes the oscillator. The first check made is that the word to write cannot differ by more than 256 from the word in the oscillator. The second is that both processors must agree on the word that is to be written. This is accomplished by using a crossed I/O port where each processor can read what the other wrote.

The next line of defense is in the hardware. The program writes an I/O port, which will strobe the data in the crossed ports into the oscillators if (1) the oscillator clamps (CC-controlled gates which, when set, inhibit oscillator writing) are not set, (2) the processors are synchronous, and (3) there are no errors in the hardware error source register.

The last line of defense is that the program always executes a 100-ms pause between oscillator writes. This allows time for the NCLK frame to detect a phase error and interrupt the CC, which, in turn, will set the oscillator clamps so that no more writes can be made. Since the clock frame can run on three oscillators, no degradation of service occurs—only a loss of redundancy.

The next step in the process is to deal with the slaves. There is an I/O port which reflects the state of the oscillator clamps. Those slaves which are unclamped are updated. As a first approximation, whatever frequency control word offset was last written to the master is written to each slave. Next the oscillator phase bits are used to more precisely align the slaves to the master.

There is a 2-bit phase indication (slave relative to master) in the I/O for each slave. These take on the values 0, 1, or 2 depending on whether the slave is within ±.5 degrees of the master, greater than +0.5 degrees, or less than −0.5 degrees, respectively. The zero condition window, usually about 50 oscillator word-bits wide, is measured periodically. If a slave is found with nonzero phase bits (after writing the master's change to it), then one-half of the window width is added or subtracted to the word in the slave. If the phase bits are still nonzero for the slave, then it is walked towards its window until the phase bits become zero.

The next step in the main loop is the writing of periodic data to the terminal. The following information is written: phase error; the four oscillator words; the integral term; which oscillators are enabled; the master; the mode (fast start or normal mode); the reference (BSRF, or T1); and the cycle counter.

As part of the main operation loop, routine exercise is performed every hour when the unit is in normal mode. The following diagnostic routines are called: ROM test, I/O test, processor self-test, phase comparator test, BSRF interface test, T1 interface test, oscillator buffer test, and the oscillator write tests. Any failures result in a processor-detected error and free run.

A set of performance reports is maintained. These reports include: number and duration of free runs, reference outages, phase hits, frequency offsets, and fast starts.

7.1.2 Operational interrupts

The operational interrupt handlers perform various functions while the unit is in a phase-locked mode or they may be used to place the unit into a phase-lock mode. The interrupts come from the CC or directly from hardware in the unit. The interrupt handlers are listed in Fig. 7.

Init is an order from CC to initialize RAM and the I/O ports. Fast start, normal, and free run are all orders that cause the unit to enter the respective states. During free run, the program is in a loop which increments a free-run duration counter. The reference change order is used by the CC to load the reference register.

The terminal handler processes requests from the terminal. A set of commands exist which allows one to change the terminal printing

format, have the machine performance report printed, run routine exercise, or print information about the last processor-detected error. In addition, the status panel may be placed in a trace mode, where the present program state and last interrupt processed are displayed. With another command one may exit to the executive program.

The dead-slave interrupt is used to bring on line a new oscillator, or one whose frequency is very different from the master's. In this region, the slave is out of phase-lock with the master and the 2-bit phase indication in the I/O is meaningless. The phase comparator is used to measure the frequency difference between the slave and the master. The output of the slave is used for the reference input to the phase comparator and then the rate of change of phase error measured. Based on the frequency difference calculated, a new slave word is calculated and written to the slave. The process continues until the clock frame is able to phase lock the slave and then the phase bits are used to bring the slave in.

Reset is the highest level of interrupt. Upon receiving this interrupt, some code is executed which initializes variables and then the processors halt. Interrupts from the halted state result in synchronous operation of the processors.

Executing an illegal OP code results in a trap. The trap handler records the program counter in RAM, prints it on the terminal, and then halts.

7.2 Diagnostics

The diagnostic routines are listed in Fig. 7 and are briefly described here.

The memory tests are divided into ROM check and RAM check. For ROM check a crossed I/O port is used where each processor reads what the other one wrote. By reading each ROM location, writing the data to the crossed port, and reading the crossed port, the processors compare the two ROMs.

The RAM test uses a conventional walking 1's and 0's algorithm. In order to discover a wider class of faults, the RAM test routine uses I/O ports for scratch pad memory, rather than RAM. The read/write I/O ports have been previously tested by CC-based diagnostics.

A processor self-check is implemented in the firmware. Here the instruction set is verified for various addressing modes by performing sample problems and comparing results against constants.

The phase comparator, BSRF interface circuitry, and T1 interface circuitry are tested by firmware routines. The build-out circuits are also tested by these routines. Test vectors are applied through diagnostic access points in the I/O.

The ability to read and write oscillators is verified by a firmware

routine. Also, the performance of the oscillator phase bits is checked. The clamps between the synchronization unit and the oscillators are also tested here.

Other diagnostic routines test the processor match circuitry, the address comparator, and the terminal interface. A routine can be called by CC, which performs a hash sum over the memory to verify the version of the program.

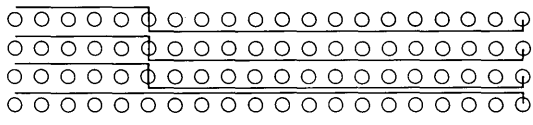
7.3 Utility program

A utility (executive) program is also included in the synchronization unit firmware. This program deals with commands typed in at the terminal. The commands enable one to read and write memory, load program in ROM-emulation, set break points, add patch code, and initiate execution of program. The utility program is a useful tool for program development as well as hardware troubleshooting.

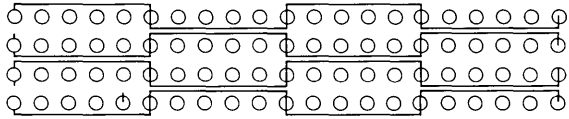
VIII. PHYSICAL DESIGN

The NCSU is a J-coded 8-inch-high by 36.25-inch-wide unit, using 1A Technology hardware. The unit contains 18 "FG" type circuit packs, each being uniquely coded apparatus with two codes having additional MC coding for the documentation of firmware. These packs are arranged in three side-by-side apparatus housings occupying the majority of the unit. (See Fig. 9.) The circuit packs are connected by backplane wiring. In the case of the microprocessor bus leads and all of the +5 volt power and ground distribution, the wiring is via the multilayer printed wiring board. The bus leads being in the multilayer board provide more noise immunity than possible with wired connections. The natural flow of these leads would form long parallel horizontal runs. The danger in so much parallel exposure is crosstalk. To counteract the natural flow, a system of routing the leads as shown in Figure 10b was adopted. Contrast this with the natural flow as shown in Figure 10a; note that while the length has been slightly increased, the amount of parallel exposure has been cut in half. Figure 11 is a photograph of the A section of one the layers of the multilayer printed wiring board, showing the actual routing.

Fig. 9—Network-clock synchronization unit (front view).



(a)



(b)

Fig. 10—Patterns. (a) Natural flow. (b) Anticrosstalk.

The unit power consists of a 140-V to 5-V power converter, power control relays, power switch, and two circuit packs.

The unit also contains a status panel (see Section IV), a terminal strip, and the jack for the BSRF.

IX. TESTING AND PERFORMANCE EVALUATION

To verify the phase-lock algorithm and its implementation, an experiment is performed. We start with an oscillator whose frequency is within 5×10^{-11} of a stable reference. Then the oscillator's word is changed by 3814 bits or a frequency offset of 1.83×10^{-7}. The unit is placed in the fast-start mode and the phase error versus time history is recorded.

Using eq. (8) with $\alpha = 6.29 \times 10^{-3}$/s and $\beta = 1.96 \times 10^{-3}$/s (for a cycle time of 8.0 s and an oscillator sensitivity of 4.8×10^{-11}/bit), we

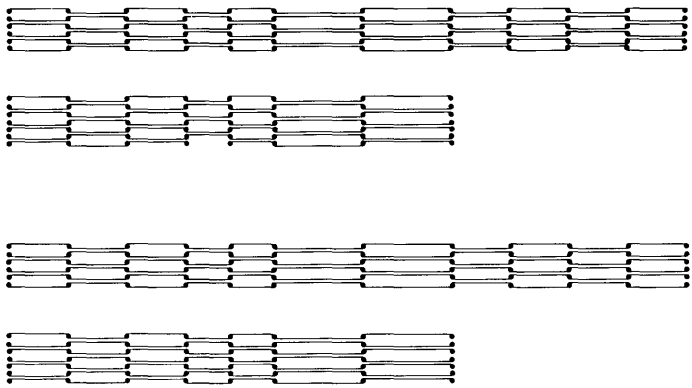


Fig. 11—Actual bus routing.

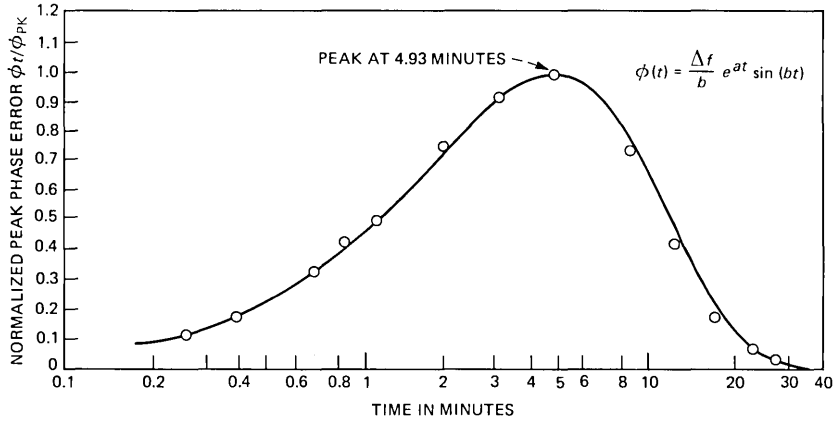


Fig. 12—Fast start.

can calculate that the peak phase error should be 85 bits of phase comparison and should occur at $t = 296$ s. The experimental result is a peak of 87 at $t = 284$ s.

The above experiment is repeated for normal mode with a frequency offset of 200 oscillator control word bits, which is equal to 0.962×10^{-8}. The experimental peak phase error is 188 and the predicted is 188.2. The peak time for the experiment is 346 min versus a predicted time of 344.

The theoretical phase error expression is normalized with respect to the theoretical peak value, and the resulting curve plotted in Fig. 12 for fast start and Fig. 13 for normal mode. The normalized data is plotted as x's in the figures. The agreement between theory and reality is quite good.

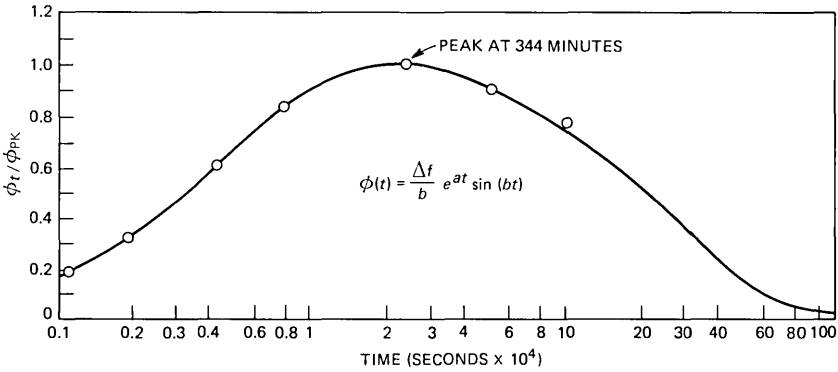


Fig. 13—Normal mode.

X. SUMMARY

Synchronization of the No. 4 ESS is performed by phase-locking the oscillators in the NCLK to the Bell System Reference Frequency or to the framing on a T1 line from another No. 4 ESS. Thus, the No. 4 ESS is in the upper layers of the treelike timing structure of the Switched Digital Network. A digital second-order phase-lock is used, exhibiting excellent convergence and stability characteristics. It is implemented with duplicated and matched microprocessors as part of the loop. This, along with extensive operational and hardware checks ensures a high degree of confidence in data written to the oscillators. Most diagnostics for the synchronization hardware are contained in the microprocessors. Finally, experimental testing of the unit has shown actual performance to conform very well to the predicted time domain response.

REFERENCES

1. A. E. Ritchie and L. S. Tuomenoksa, "No. 4 ESS: System Objectives and Organization," B.S.T.J., *56,* No. 7 (September 1977), pp. 1023–4.
2. J. E. Abate et al., "The Switched Digital Network Plan," B.S.T.J., *56,* No. 7 (September 1977), pp. 1312–3.
3. J. E. Abate et al., "Synchronization Considerations for the Bell System Switched Digital Network," ICC 1979 Proceedings, Boston, Mass., June 10, 1979.
4. J. H. Huttenhoff et al., "No. 4 ESS: Peripheral Systems," B.S.T.J., *56,* No. 7 (September 1977), pp. 1034–5.
5. B. R. Saltzberg and H. M. Zydney, "Digital Data System: Network Synchronization," B.S.T.J., *54,* No. 5 (May–June 1975), pp. 881–6.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 60, No. 6, July–August 1981
Printed in U.S.A.

No. 4 ESS:

Digital Interface

By K. M. HOPPNER, H. MANN, S. F. PANYKO, and J. VAN ZWEDEN

(Manuscript received August 22, 1980)

This article describes the hardware and maintenance software implementation used for the Digital Interface (DIF), a new transmission interface for the No. 4 ESS. The DIF replaces the Digroup Terminal (DT) and the Signal Processor 2 (SP2) used for terminating digital carrier trunks. It also provides a more economical transmission interface for the No. 4 ESS, enhances the use of the LT-1 connector for terminating analog carrier facilities, and provides a standard peripheral/processor interface for maintenance. In conjunction with the DIF, a hierarchical modularly structured maintenance software system was introduced to support the new No. 4 ESS peripherals. Incorporated into this system were the mechanisms required to support the introduction of microcomputer-based peripherals such as DIF.

I. INTRODUCTION

The Digital Interface (DIF) is a newly introduced No. 4 ESS peripheral frame whose functions combine the operations performed by the Digroup Terminal (DT) and the Signal Processor 2 (SP2), with the exclusion of the supplementary matrix frame which is an SP2 optional adjunct.[1,2]

The main functions of a DIF are to terminate DS-1 level signals and to multiplex them to a form suitable for the No. 4 ESS digital switch; to perform the necessary signaling interchange between the transmission and switching facilities; and to provide adequate fault detection and reconfiguration capability. These functions are identical to those performed by the DT/SP2 complex but the DIF achieves them in a more compact and modern fashion. The DIF occupies less than one-half the

space and uses two-thirds less power than the DT. This reduction was achieved in a number of ways:

(*i*) by combining the equivalent of four DTs and an SP into a single structure thereby eliminating a number of duplicated functions and a complex interface,

(*ii*) by using custom and catalog large-scale integrated circuits instead of the 1A Technology previously used, and

(*iii*) by utilizing both metal oxide semiconductors and bit-sliced microcontrollers to replace the hardwired logic of the DT and SP controllers.

As will be discussed in more detail later, these changes resulted in a frame with an entirely different internal philosophy which was more flexible and more consistent with the architecture of the other peripheral frames in the No. 4 ESS. Sections II and III give a high-level logical and physical view of the DIF. The article then describes the Digital Interface Unit (DIU) architecture in Section IV, followed by the Digital Interface Controller architecture in Section V. Finally, the maintenance software developed for the DIF is presented in Section VI.

II. OVERALL FRAME ARCHITECTURE

A fully equipped DIF consists of a duplex controller (DIC), duplex Interface to the Peripheral Unit Bus (IPUB), 32 working DIUS, and two protection spare DIUS as shown in Fig. 1. The DIF connects directly to the Peripheral Unit Bus (PUB), interfaces directly or via an echo suppressor terminal with the Time Slot Interchange (TSI) of the No. 4 ESS network in the DS-120 format, and provides a DS-1 transmission interface with T1 facilities or LT-1 as shown in Fig. 2. Each DIU terminates five DS-1 signals, giving the DIF a capacity to terminate a

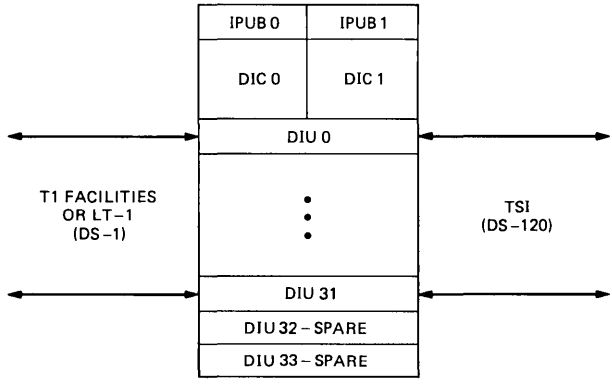


Fig. 1—DIF block diagram.

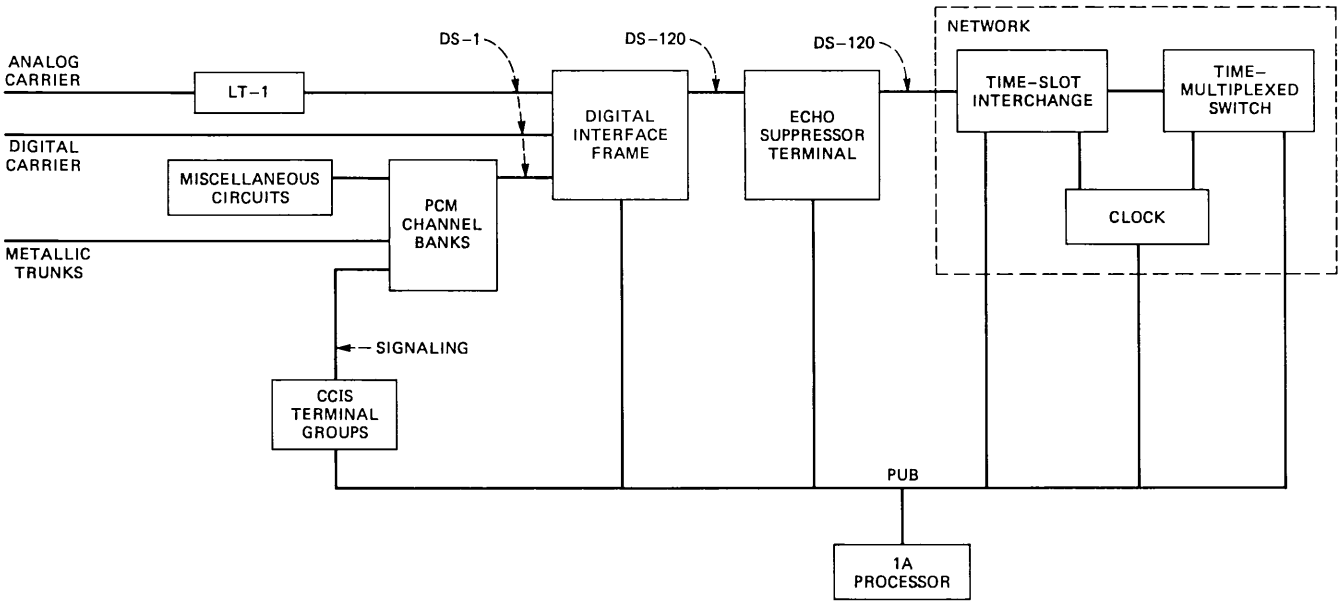


Fig. 2—No. 4 ESS block diagram.

total of 160 DS-1 signals (3840 trunks). A photograph of the DIF appears in Fig. 3.

III. PHYSICAL DESIGN

3.1 Overview

The circuitry which makes up the DIF is of two types. One type is of a transmission nature which is characterized by relatively few functional blocks arranged in a serial fashion. The other type is of a processor nature consisting of many functional blocks which have a high degree of interconnectivity typically in the form of large high-speed parallel buses. Within the limits of technology used in the DIF, circuit pack partitioning of the processor function indicated the need for a large Input/Output (I/O) circuit pack capability, whereas the transmission type of circuits required significantly less I/O. This difference influenced the physical design of the DIU, which is basically a transmission function, and that of the DIC, which is basically a processor or computer type of function.

The physical design objectives of the DIF included the following:

(*i*) Compatibility with the No. 4 ESS environmental, reliability, and frame I/O connector requirements.

(*ii*) Significant cost, space, and power reduction over the DT/SP2 complex which it replaces.

(*iii*) Circuit partitioning which facilitates interconnection, fault detection, fault diagnostics, and maintenance.

(*iv*) Physical embodiment which enables proper electrical functioning.

3.2 Device technology

The DIF uses low- and medium-power transistor-transistor logic, low- and high-power Schottky TTL, emitter-coupled logic, complementary metal-oxide semiconductor, and N-channel metal-oxide semiconductor integrated circuit technologies. The level or scale of integration ranges from small-scale integration to large-scale integration (LSI) with a fully equipped DIF using a total of about 18,000 devices of about 200 codes.

3.3 Frame and circuit-pack partitioning

The circuit-pack size chosen for the DIUs was determined by the DS-1 interface, described in Section IV, since this is the dominant entry in each DIU (five out of a total of nine circuit packs). Thus, a printed wiring board (PWB) which is nominally 8 inches high and 9 inches deep with a 114-lead connector was chosen for the DIU circuitry. This size made possible the double-sided PWB implementation (see Fig. 4) of the

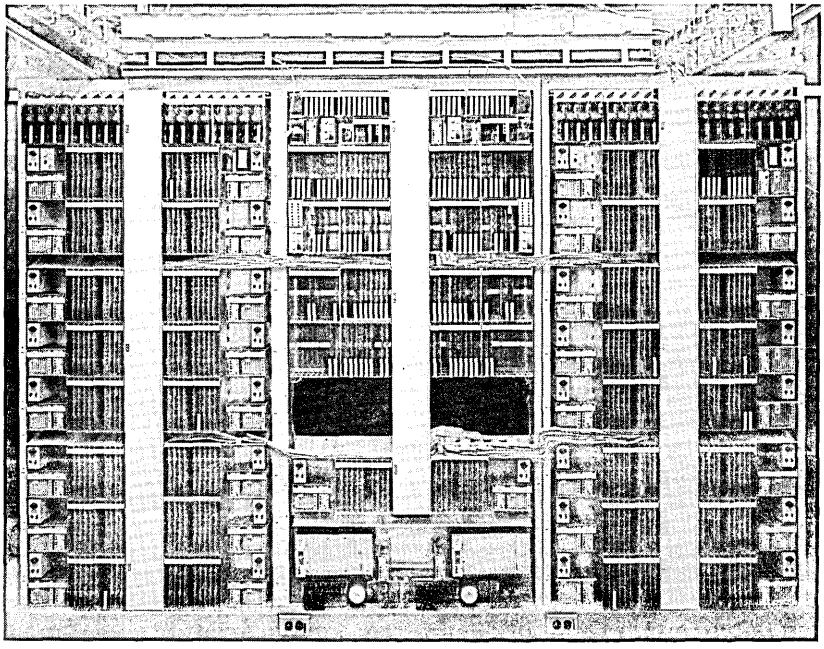


Fig. 3—Photo of DIF.

DS-1 interface which is the most cost-sensitive circuit in the DIF because of the large number used (170/DIF). The remaining DIU circuitry required two double-sided and two four-layer PWB circuit packs. The four-layer configuration is capable of accommodating approximately 50 percent more circuitry than the double-sided boards.

The same basic circuit-pack size is also used for the DIC; however, because of its large I/O requirements, a 184-lead connector is used instead of the 114-lead connector. The DIC consists of 94 circuit packs of which 48 are four-layer PWBs and the remaining 46 are double-sided.

Since all DIUs typically communicate with the controller via a bus, locating them about a central cable-duct (see Fig. 5) permits the bay cabling to be kept to a minimum length with a common access point for a pair of DIUs which minimizes the number of cable connectors required. Those circuit packs that interface with the DIC are located near the cable duct. Therefore, the circuit pack positions of the DIU on the right side of the duct are a mirror image of those of the DIU on the left side.

The DIF uses the No. 1 ESS framework: one double-bay frame which is 7 feet high, 6 feet 6 inches wide, and 12 inches deep; and one single-bay frame which is 3 feet, 3 inches wide. Two DIUs and their associated power units and switches are mounted on an 8-inch-high shelf. At the

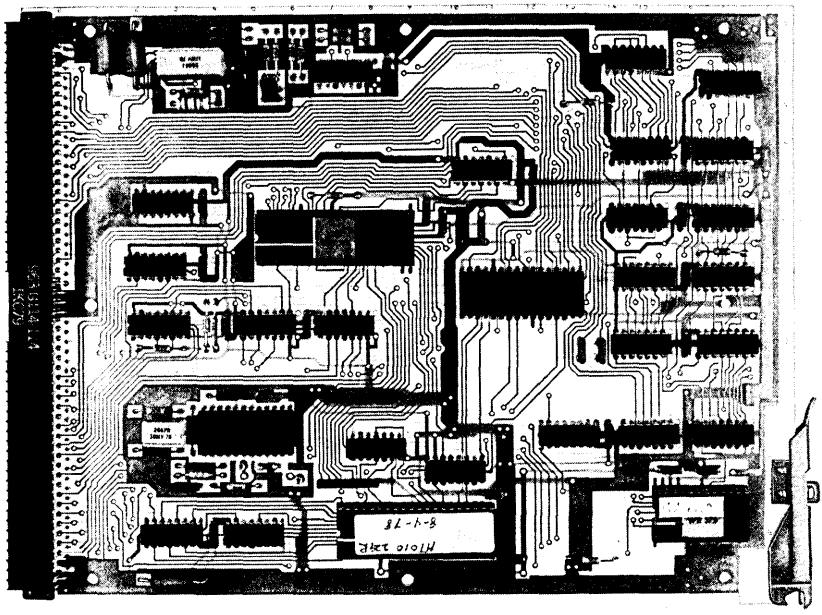


Fig. 4—Double-sided PWB.

top of bay 0 and bay 2 are the protection switch circuit packs and the DS-1 and DS-120 I/O connectors. These circuit packs are nominally 6 inches high and 9 inches deep. They contain the relays required for transferring the DS-1 and DS-120 signals from a DIU to a spare DIU. They also contain equalizers for the DS-1 lines and the transmit and receive coax connectors for the DS-120 signal. DS-1 connectors are located on a panel directly above the protection-switch circuit packs. The spare DIU in bay 0 provides backup protection for 15 DIUs located in bay 0 and the left DIU located in bay 1. Likewise, the spare DIU in bay 2 provides backup protection for the 15 DIUs located in bay 2 and the right DIU located in bay 1.

The duplex PUB interface is located at the top of bay 1 with its associated power units as shown in Fig. 5. The circuit packs which make up this function are nominally 4 inches high and 9 inches deep and use a 92-lead connector. The duplex controller is immediately below it and the associated power units are located at the bottom of the bay.

The circuit-pack positions of the PUB interface are not mirror-imaged about the cable duct as are those of the DIU and the DIC. This is for two reasons. First, the PUB interface does not use the cable duct for interconnecting to the controller. Second, the PUB is connected via cable connectors to the same backplane pins which the circuit packs

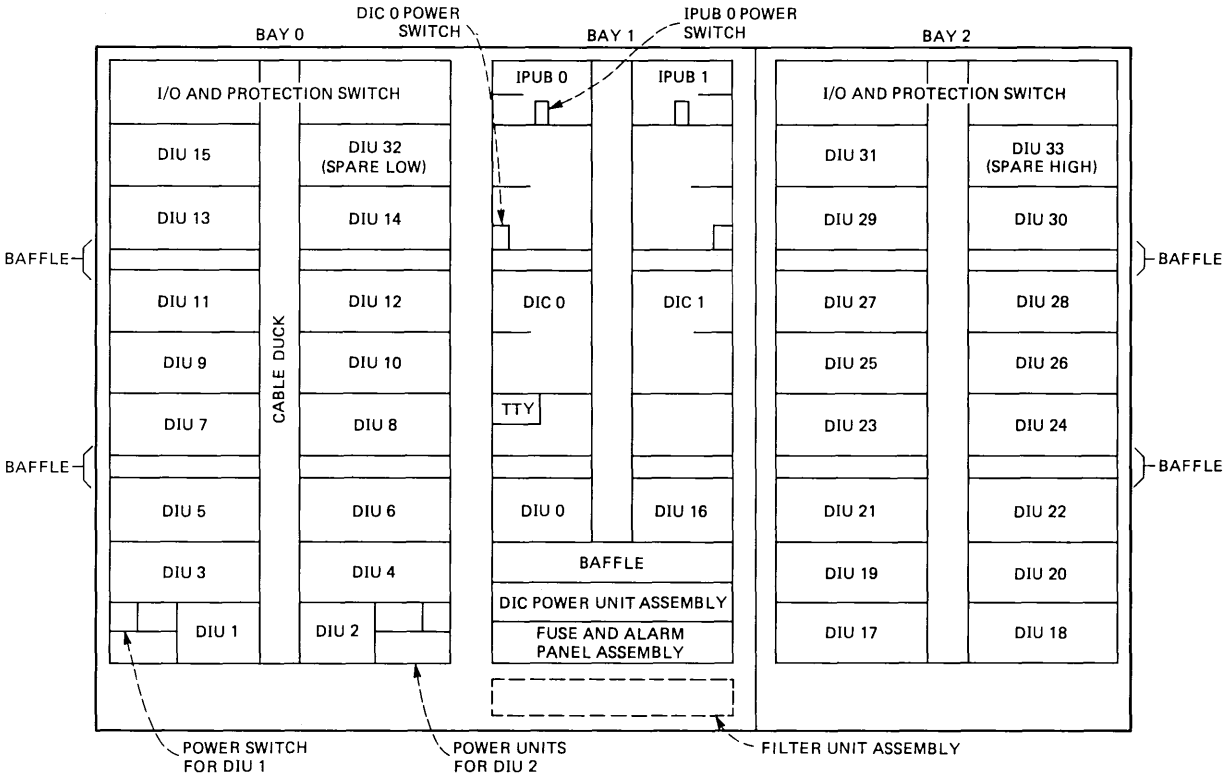


Fig. 5—DIF diagram (PUB interface and power units).

connect to; thus, to provide a PUB connector arrangement which is identical for both PUB0 and PUB1 to simplify installation, the PUB interface 0 and 1 circuit packs must have the same order. This arrangement is also used for other No. 4 ESS peripherals.

The power switches for the PUB interface, the DIC, and the DIUS are collocated with their respective functions to make their association self-evident.

Heat baffles are used to divert air flow out of the back of the frame and to cause aisle air to be taken in from the front of the frame at various levels. This prevents the cooling air from becoming excessively heated as it rises through the frame.

3.4 Frame cabling

Most of the frame cabling is 26 American wire gauge (AWG) twisted-pair, 24-conductor flat cable. The wires are on 0.062-inch centers with an untwisted section every 18 inches for terminating connectors (see Fig. 6). The cable is terminated to connectors which have insulation displacing terminals. These connectors plug onto 25-mil square pins located in the cable duct area.

The frame cabling is located on the circuit-pack side of the backplane

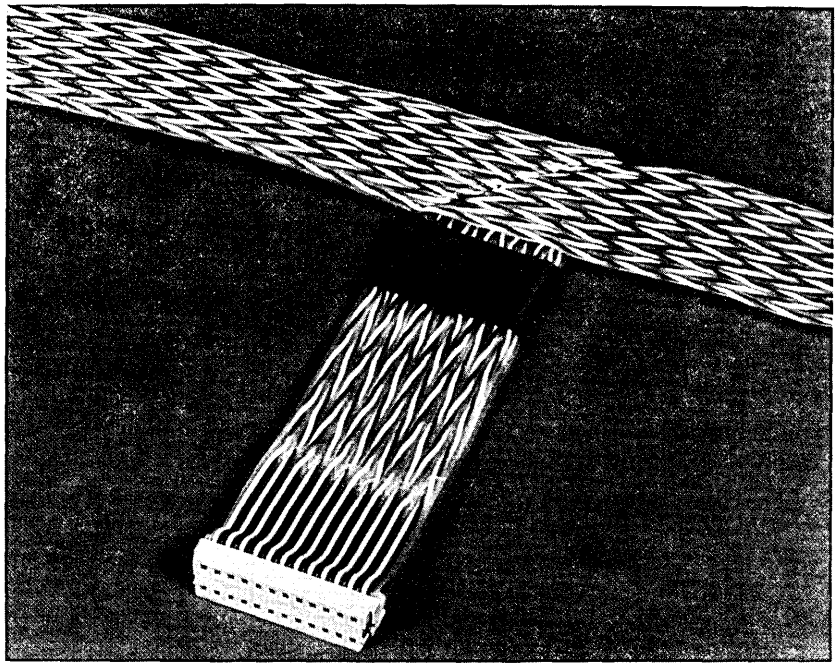


Fig. 6—Frame cabling (flat cabling.).

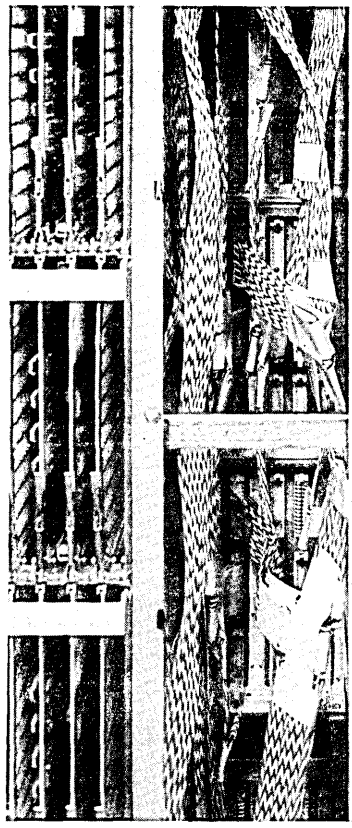


Fig. 7—Frame cabling (cable duct and local cable).

(see Fig. 7), and wiring to the cable connector pins is done in the backplane with surface wiring. This cabling scheme isolates the frame cabling from the backplane surface wiring by designating a specific area for frame cabling. The cable is routed horizontally between cable ducts in designated areas located in front of the frame upright on the top surface of heat baffles.

3.5 Circuit-pack design

The circuit-pack PWB sizes used in the DIF conform with the *Bellpac** standard circuit-pack size. The circuit-pack connectors and the latch used on these circuit packs are part of the *Bellpac* hardware system.[3] Before the design of circuit packs, design standards were determined to assure a functional and manufacturable design. These

* Trademark of Western Electric.

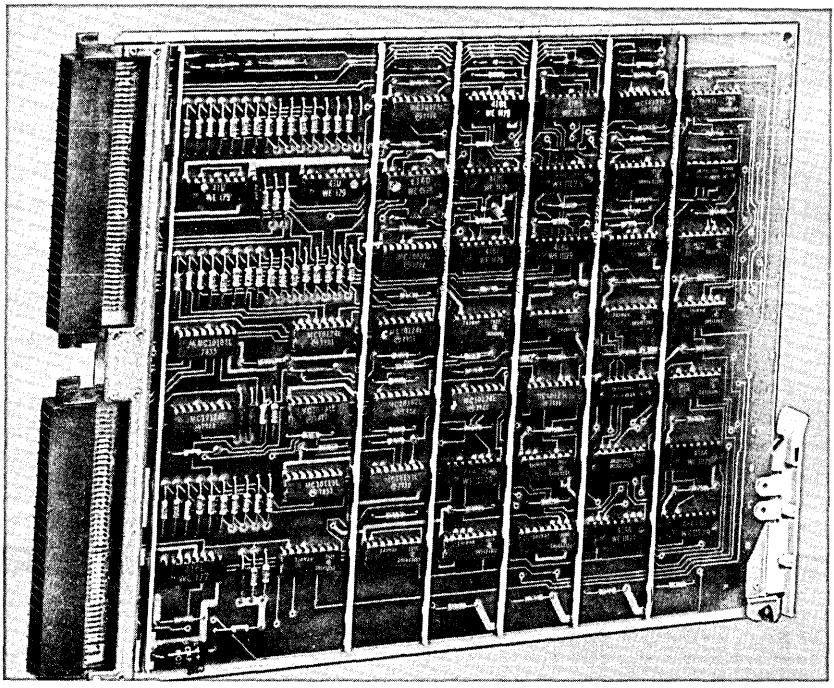


Fig. 8—DIC circuit pack utilizing appliqued bus bars.

standards include maximum component heights; component placement constraints; conductor path grid, sizes, and routing; plated-through hole sizes, lands, and placement; connector fanout pattern; placement of I/O devices; and use of power/ground decoupling capacitors.

An appliquéd bus bar is used on some double-sided PWB circuit packs for distributing power and ground (see Fig. 8). This makes more area available for routing signal paths, since wide printed paths are not needed for distributing power to the integrated circuits.

The four-layer PWB circuit packs consist of a power surface layer, two buried signal layers, and a ground surface layer (see Fig. 9). The signal layers are buried so that their fine features (metalization and clearances as small as 12 mils) are protected, whereas the power/ground layers which consist of relatively conservative features (metalization and clearances of 25 mils or greater) are placed on the outer layers.

3.6 Backplanes

The backplanes which interconnect the circuit packs consist of pin-populated PWBs—see Fig. 10. These backplanes use the *Bellpac* com-

pliant pin and are designed to *Bellpac* requirements.[3] Power and ground are distributed to the circuit pack positions by printed paths. The signal interconnections are in the form of 30 AWG single-ended and twisted-pair wire, terminated by wire-wrap connections.

3.7 Power

Duplicated 140-volt and 24-volt power feeds are required by the DIF. The 140-volt power is used for powering the power units which provide ±5 volt and +12 volt power for the integrated circuits. The 24-volt power is used for the alarm relays, power control circuitry, IPUB drivers, protection switch relays, and indicator lights. The DIF draws 8.3 amps from each 140-volt bus and 1.7 amps from each 24-volt bus.

3.8 Operating environment

The DIF was designed to operate in the No. 4 ESS office environment. This required that it be capable of operating over a temperature range of 4°C to 38°C and a relative humidity range of 20 percent to 55 percent, except for a total of 15 days per year during which time it must operate from 2°C to 50°C and 20 percent to 80 percent relative humidity for no more than 3 days at a time. To achieve these objectives, hermetically sealed integrated circuit (IC) packages or beam-

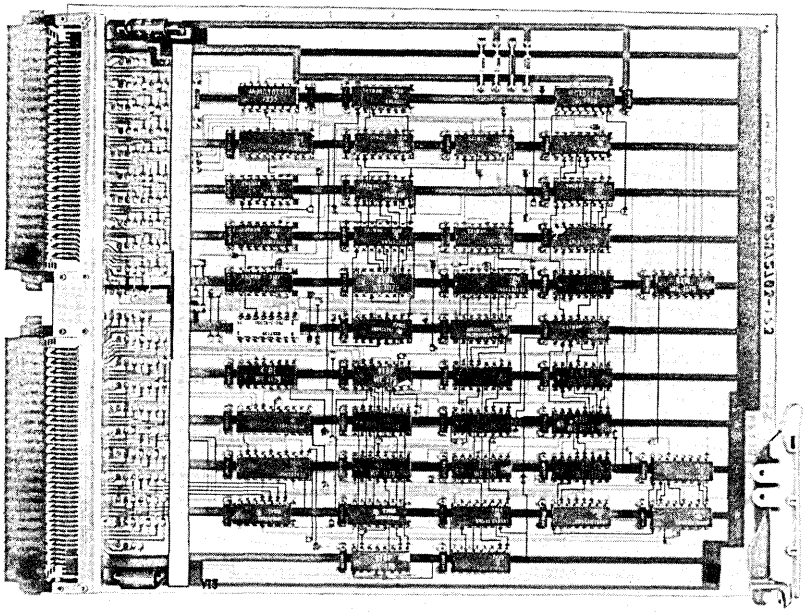


Fig. 9—DIC 4-layer PWB.

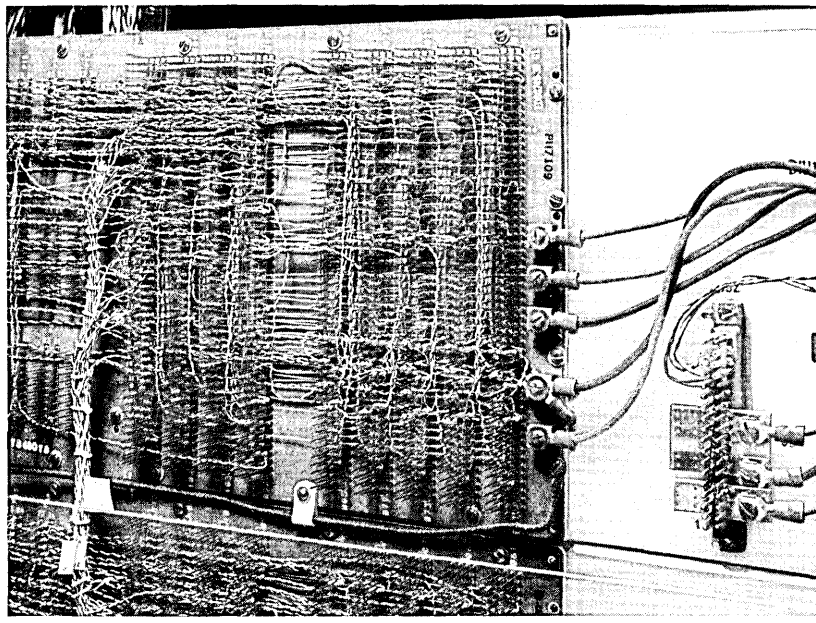


Fig. 10—DIU backplane.

leaded sealed junction ICs were used with a typical operating ambient temperature capability of 90°C. This operating temperature allows for some margin since the maximum temperature rise above the aisle ambient in the DIF is 30°C.

3.9 DT/DIF comparison

The DIF represents a significant size, power, and cost reduction over the DT/SP2 complex. One DIF interfaces 160 T1 lines, whereas four DTs and one SP2 are required to interface this same number of lines. The DIF lineup is 9 feet 9 inches long versus 23 feet 10 inches for the DT/SP2 complex. The power dissipation of the DIF is 2500 watts versus 7500 watts for the DT/SP2 complex. A DIF is around 50 percent cheaper than the DT/SP2 complex.

IV. DIGITAL INTERFACE UNIT

The DIU consists of the four following major blocks (see Fig. 11):

(*i*) the DS-1 interface (one per-digroup),

(*ii*) the DS-120 interface,

(*iii*) clock generation, (*iv*) control interface.

The DIU differs from the digroup terminal unit in that a number of per-digroup functions which were performed using common control

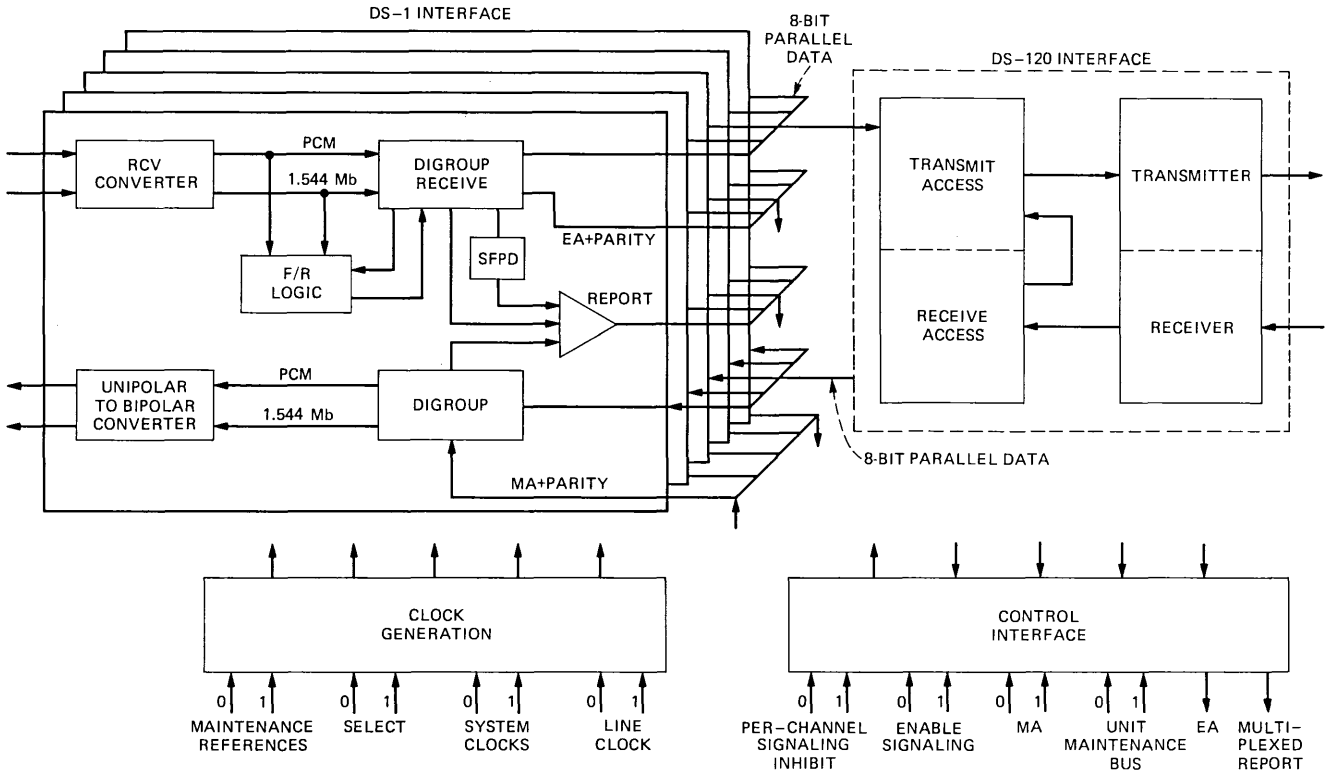


Fig. 11—DIU block diagram.

techniques are now done on a per-digroup basis. This permits the use of custom LSI, which can be shared with other systems and reduces the complexity of maintenance as test vector generators and references are no longer required.

4.1 DS-1 interface

Figure 12 is a detailed diagram of the DS-1 interface which consists of a receive section, a transmit section, and a maintenance and report section.

4.1.1 Digroup receive function

It is the function of the DS-1 receive circuitry to terminate the T1 line, to recover line timing and data, and to convert the incoming data to a form suitable for multiplexing with other digroups into a DS-120 level.

The conversion from bipolar to unipolar, clock extraction and signal regeneration are done with a custom LSI receive converter complementary bipolar integrated circuit chip. The output of this chip is a single-rail unipolar PCM serial stream and a properly phased 1.544-Mb clock. These signals feed the framing and receive logic chip (F/R) and the digroup receive chip (RCV LSI) both of which are custom LSI devices. The F/R recovers the DS-1 data and signaling framing. To function, this chip requires an external line channel counter which is located on the RCV LSI (see Fig. 13). Framing is accomplished by the F/R interrupting the line channel counter a sufficient number of times to restore framing.

The F/R chip supplies a signal indicating a framing error. This signal is used to estimate the error rate and to block the updating of signaling information. The chip also generates two signals which mark the A and B signaling frame.

The major per-digroup receive functions are contained in the RCV LSI (Fig. 13). The primary function of the RCV LSI is to reduce the incoming PCM data 125-μs frame to a specific 23.4-μs digroup interval, to recover A and B signaling bits, and to compensate for differences in line-frame frequency and phase with respect to office-frame frequency and phase. Although the RCV LSI recovers both A and B channels, only the A channel is implemented in the present version of the DIF.

PCM data is converted from serial to parallel (S/P) form and complete 125-μs frames are stored in an A and B random-access memory (RAM) store. (A and B here should not be confused with A and B signaling bits.) The stores are alternately written and read so that in general, while one store (e.g., A) is being written under line timing, the other store (e.g., B) is being read under office timing. If the relationship of line frame to office frame allows reading and writing the store in such

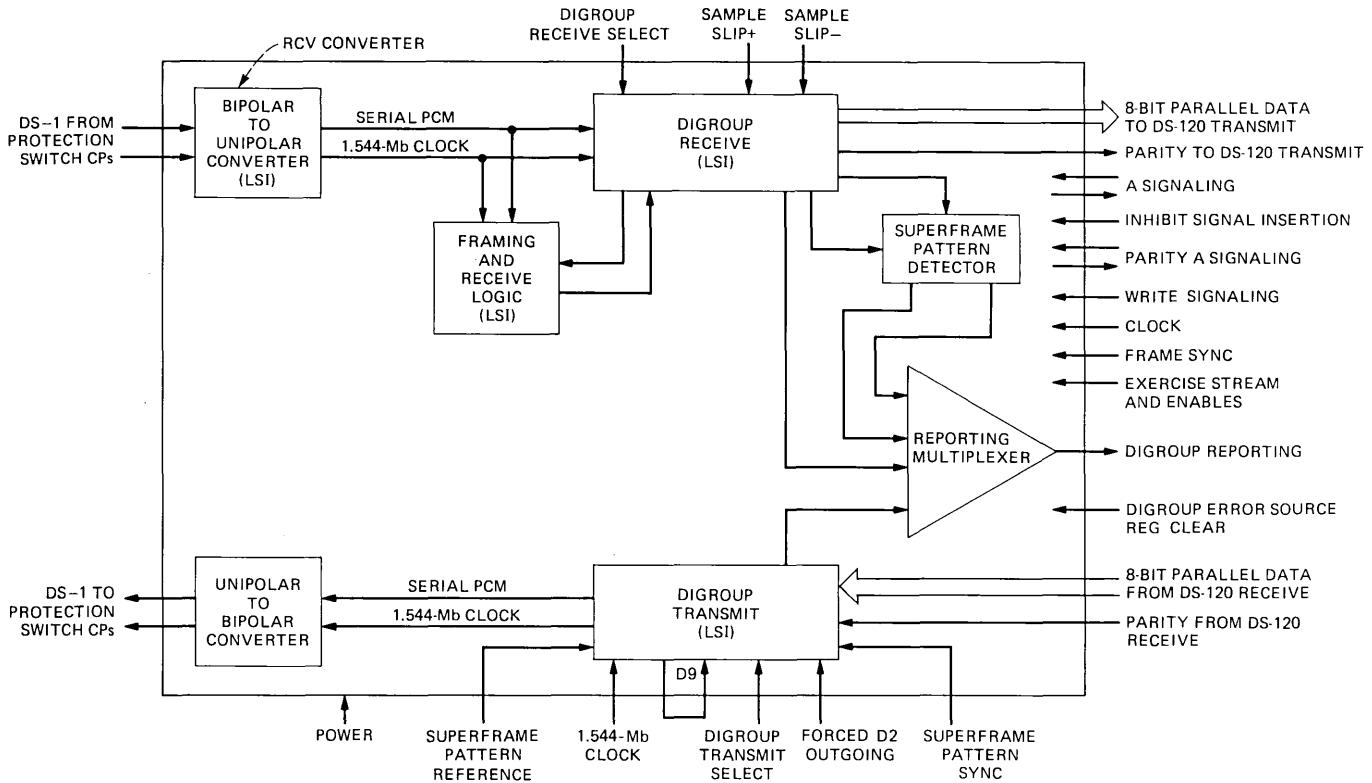


Fig. 12—DS-1 interface.

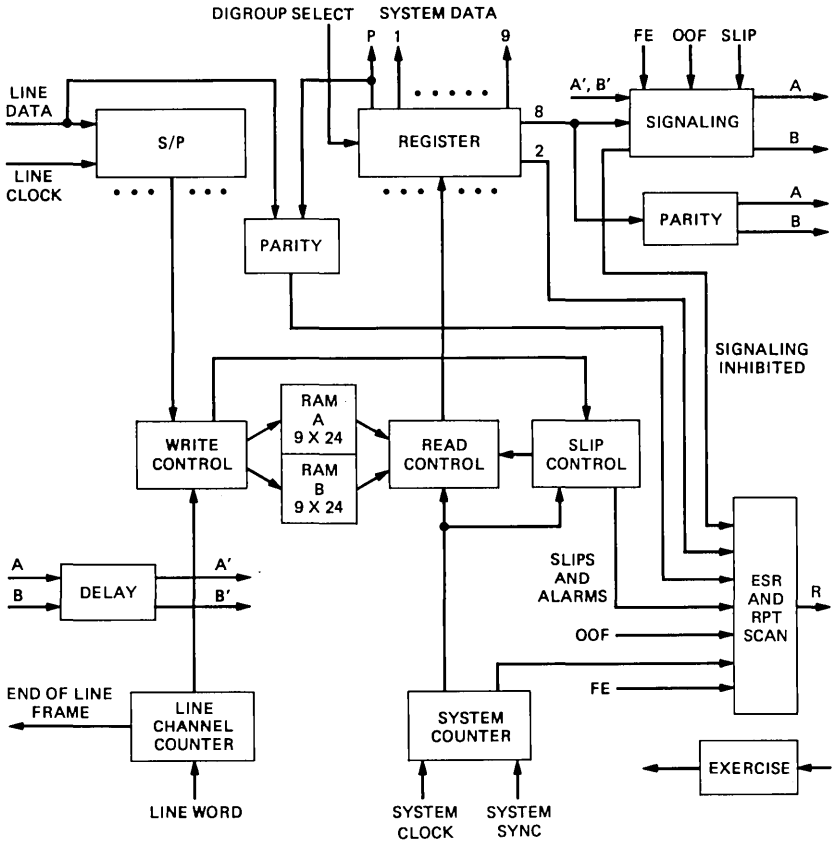


Fig. 13—RCV LSI block diagram.

a manner as to intermix frames, a slip control generates a double A read to correct the situation.

PCM data through the stores is maintained by serial parity over a frame of data. Parity generated at the input to the RCV LSI is stored and compared with parity generated at the output of the A and B stores. In addition to parity, the framing bit, D9, is passed through the stores for maintenance purposes. This is possible since the contents of the D9 bit are defined by signals from the F/R chip.

Extraction of signaling information occurs at the output of the A and B RAMs. Signaling information is stored in two 24-bit shift registers (signaling) and the data is checked by parity. Data is entered into these stores under the command of signals from the F/R chip which indicate when to extract signaling information.

The RCV LSI also contains a detector which determines if PCM bit 2 is held at zero for all frames in a 32-ms interval. This signal is forwarded

to the DIC [via the error-source register (ESR) and report (RPT) scan], which, in turn, times it for the remote (yellow) alarm.

A number of functions are combined on the RCV LSI chip to form a report function stream:

(*i*) line bit D2 stuck at zero,
(*ii*) out of frame (OOF),
(*iii*) framing error (FE),
(*iv*) positive slip (positive slip occurs when the system frame exceeds the line frame rate),
(*v*) negative slip,
(*vi*) alarms.

These reports are used by the DIC to determine the status of the receive portion of the DS-1 interface.

The RCV LSI contains a number of matchers and alarms. Detected failures are combined into a single alarm list. To determine if the alarms and matches are functioning, exercises can be sent to the RCV LSI, which generates alarms without interfering with the processing of data. A digroup clear input is used to clear alarms.

4.1.2 Superframe Pattern Detector

The Superframe Pattern Detector (SFPD) shown in Fig. 12 maintains the integrity of the data flow through the RCV LSI by checking that the D9 bit contains the subframe pattern and that the phasing of this pattern corresponds to that derived from the F/R chip. To avoid the complication of accounting for slips, framing errors and out-of-frames, the SFPD is inhibited during these states.

The ability of the SFPD to detect superframe pattern errors is tested by frame-resident exercise functions (Sections 4.5 and 6.5).

4.1.3 Digroup transmit functions

In the transmit direction, the DS-1 interface receives parallel data, plus even parity from the DS-120 interface. The main functions of this portion of the interface are to select the data for the appropriate digroup, convert it to a 1.544-Mb/s serial stream with the appropriate framing information, and to insert signaling information at the appropriate time. The transmit circuitry consists of two major blocks, the digroup transmit chip (TMT LSI) and the unipolar to bipolar converter, which is comprised of discrete components. The major functions are performed in the TMT LSI (see Fig. 14). Although each digroup interface receives its data during a different 23.4-μs interval, the outgoing DS-1 data are frame and framing pattern aligned for maintenance purposes.

The TMT LSI, like the RCV LSI, contains an address generator, a read-write control, and a RAM divided into A and B sections. While a whole frame of data is being written into one section under system timing,

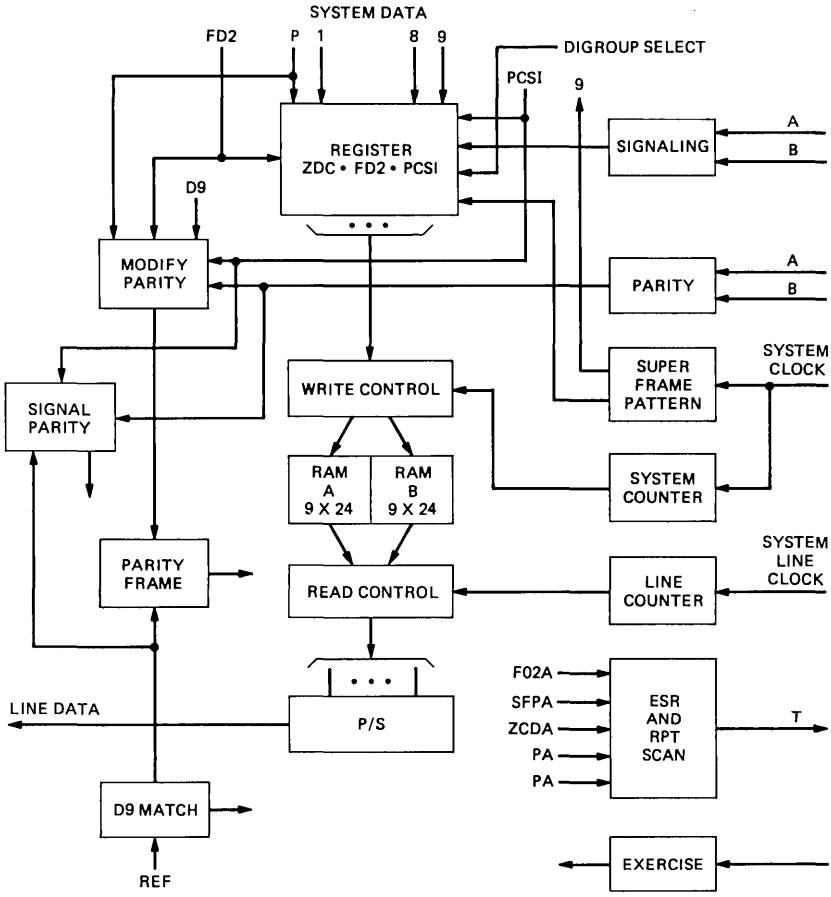


Fig. 14—TMT LSI block diagram.

the other is being read under line timing. Two stores are used for convenience as this allows all digroups to be checked with a reference pattern at the same time on the T1 line side. The TMT LSI also receives signaling information (plus parity) from the control interface (CI) (see Fig. 11) for the A signaling channel, which it stores in a 24-bit shift register in which it is recirculated until an update interval occurs. A superframe pattern generator driven by a synchronizing signal from the CI defines the time at which signaling insertion should take place. The TMT LSI has the capability of handling both A and B signaling storage and insertion, but in the present DIF, only the A channel is used. The TMT LSI also has the ability to prevent the insertion of signaling data in bit position 8 via a control signal from the CI. When signaling is inserted, the parity over data is altered.

At the output of the TMT LSI, serial parity over a frame of data is compared against stored input serial parity over the same data. In addition, the output serial parity over D8 for a frame is compared against stored output serial parity over D8 for signaling insertion maintenance. The TMT LSI contains a zero code detector and a forced D2 inserter. The zero code detector forces bit 7 to a one if all bits of a word are zero. The forced D2 inserter under command of the CI forces bit 2 to a zero to transmit the yellow alarm to the distant terminal. This action is initiated by a command from the DIC. The forced D2 feature is maintained by returning a status bit to the CI, while the zero code detector is duplicated.

As in the RCV LSI, the TMT LSI multiplexes all of its status and alarm bits into a single stream which is forwarded to the reporting multiplexer. In addition, it has exercises to test the various matches and alarms.

The serial PCM data from the TMT LSI is modified by the unipolar to bipolar converter. This block is made of discrete components and puts out a bipolar signal of the proper amplitude and shape for driving a T1 line.

4.2 DS-120 interface

The DS-120 interface consists of four functional blocks: the receive access, the receiver, the transmit access, and the transmitter (see Fig. 11).

4.2.1 Receiver and receive access

The DS-120 termination contains an analog line receiver similar to the one used in the digroup terminal whose function is to terminate the coaxial cable, amplify the PCM data signal from the TSI, and supply a sampling clock for the data. The data signal framing is determined and the signal is converted from the bit/bit complement serial format of the DS-120 link to an 8-bit parallel form with parity.

The receiver contains a squelch function. If two successive frames have pair violations, the squelch is applied. When the receiver recovers frame, the squelch hangs over for 8 ms. The squelch applies an all-ones code to the outgoing data to prevent analog carrier overload.

In the receive access, test vectors (derived from unit clock signals) are inserted in spare time slots 127 and 0. These vectors cannot propagate through the per-digroup equipment. However, time slots 125, 126, 127, and 0 are forwarded to the transmit access. A two-time slot delay occurs between the transmitter and the receiver. Hence, when a DIU is looped on itself, data in time slots 127 and 0 are sent to 1 and 2, 3 and 4 are sent to 5 and 6, and so forth. Thus, when looped, the vectors eventually occupy all working time slots and can be used

to test the looped DIU. This condition pertains to the spare DIU when not in service and for a DIU when it is protection-switched while in an out-of-service condition. If a DIU cannot be protection-switched, it is possible to notify a distant office of an out-of-service condition under most circumstances, by sending an exercise that blocks the transmission of the framing signal. Time slots 125 and 126 are looped to permit the TSI to test transmission to and from the DIF.

4.2.2 Transmitter and transmit access

In the transmit access, the receive and transmit streams are combined together. To ensure proper operation of the access gates, parity over the spare time slots 125 and 126 is even while transmit data parity is odd. A failure to multiplex generates a parity failure. The conversion of data from a parallel form to the bit-bit prime serial format of the DS-120 link is protected by recomputing serial parity, in test vector time slots 127 and 0, and comparing it with a reference parity for those time slots.

The transmitter consists of a TSI line driver similar to the type used in the DT which amplifies and buffers the DS-120 signal to drive up to 1000 ft of 100-Ω coaxial cable.

4.3 Clock selection, generation, and decoding

To reduce the number of leads between the DIC and the DIUs, each DIU generates its own clock chains. All DIUs must be frequency- and phase-locked to the DIC; this is achieved by furnishing to each DIU six signals over a duplicated link:

(*i*) System signals
  (*a*) 16.384-Mb/s square wave clock,
  (*b*) 8-kHz synchronizing signal,
  (*c*) 31.25-kHz synchronizing signal.

(*ii*) Line signals
  (*a*) 1.544-Mb/s square wave clock,
  (*b*) 666.67-kHz synchronizing signal for the T1 line superframe pattern,
  (*c*) 8-kHz synchronizing signal for the T1 channel counters.

In addition to the above signals, two reference signals are sent to the DIUs to ensure that the local clock generation is proper. The first is a signal representing parity over all the system clocks generated in the DIU having periods from 250 μs to 32 ms. The second is a reference signal which is compared with superframe patterns generated in the digroup transmit chip for maintenance purposes.

All of the above signals are duplicated and selected by a 1A Processor command via the DIC.

4.4 Control interface

Information exchanged between a DIU and the DIC is processed by the CI.

From the DIC to each DIU there are four data streams:

(*i*) M signaling (A channel outgoing),
(*ii*) enable signaling (ENSIG),
(*iii*) per-channel signaling inhibit (PCSINH),
(*iv*) unit maintenance bus (UMB).

These signals are originated in each of the controller halves, distributed to all units serially via balanced drivers on a duplex basis, and selectable by a control command via the CI.

M signaling contains the multiplexed signaling data to be distributed to the various DIUs and digroups. Each DIU has a hardwired time of unit code (TU), so that a particular DIU can only off-load signaling during a unique 125-μs interval every 8 ms. Data integrity is maintained by odd parity in time slot 127. The ability of a DIU to correctly decode the TU is checked every 8 ms by the DIC via the looped spare time slots. The CI generates serial parity over each stream and forwards them to the five DS-1 interfaces along with the multiplexed signaling stream. The per-digroup circuitry picks off its signaling data based on a digroup select clock signal. If a parity failure occurs for the incoming data, the CI alarms but does not block updating as that would require a 128-bit store. Signaling information, however, cannot be updated unless the ENSIG is in the proper state. Enable signaling permits the DIC to block updating of DIUs from a faulty DIC half. If the signaling failure occurs because of a detected DIU failure, the unit is protection-switched.

Per-channel signaling inhibit allows a DIU to block the insertion of signaling information into the DS-1 stream for any channel in that DIU. The CI contains a 128-bit store which holds the PCSINH status. This signal permits the DIU to pass full 8-bit information for common-channel interoffice signaling (CCIS) or for permanently connected special service trunks. The CI loops and multiplexes the received spare PCSINH time slot data with the M signaling spare time slot data so that the DIU can maintain the distribution to the DIU. Distribution is checked every 16 ms by the DIC. Data to the DIUs are protected by odd parity in time slot 127. The CI distributes PCSINH on a looped basis to all digroups and the parity over the data returned to the CI is compared against the parity received from the DIC.

From each DIU to the DIC two streams of data exist:

(*i*) multiplexed report stream,
(*ii*) E signaling stream (A channel incoming).

The report stream is composed of the per digroup reports (24 bits per digroup) as shown in Fig. 15. Thus, the report stream is composed of 120 bits for each DIU. The various per-digroup report streams are multiplexed in the DIU backplane and this multiplexed stream is then combined with the 8 common alarm bits in the CI. The DIU repeats the reports every 125 μs. The common control reports are autonomously cleared every 32 ms, whereas the per digroup reports are cleared via a command from the DIC. The autonomous clear for each DIU occurs at staggered times so that the DIC can interrogate the DIUs without danger of missing a failure report because of a DIU self-clearing action.

The multiplexed report stream which is transmitted in simplex form to the DIC from each DIU is driven from an unbalanced source and its ability to transmit data is maintained by frequent controller exercises.

The signaling stream is comprised of five multiplexed digroup E signaling channels and looped channels. Seven spare signaling channels (121 to 127) received by the CI are looped back to the DIC via the A signaling stream with a 125-μs delay. The DIC will only look at the

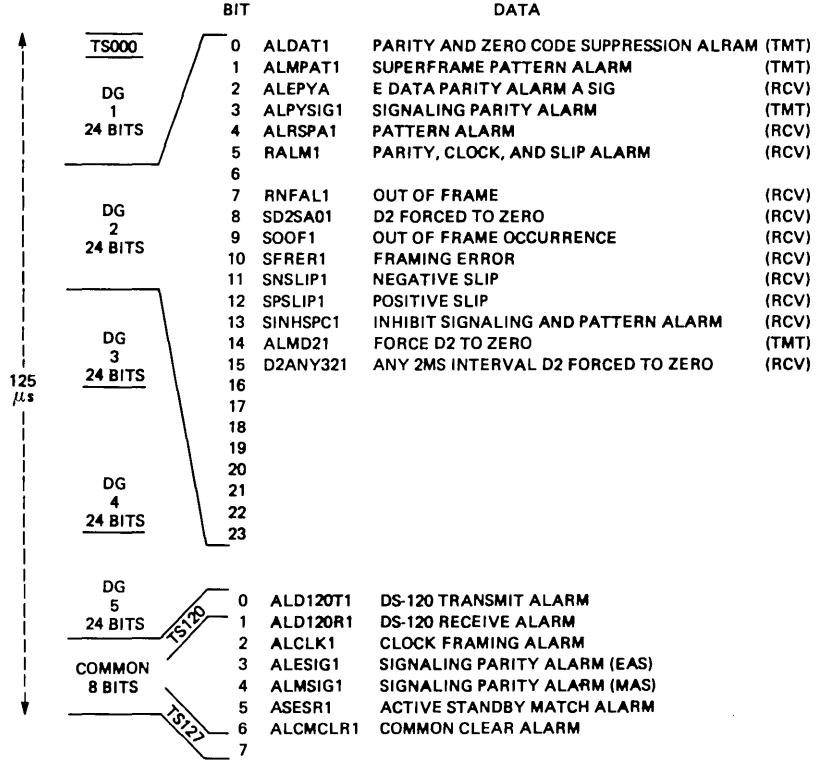


Fig. 15—DIU multiplexed report stream.

looped data during its normal E scan once every 8 ms. Reception of M signaling and PCSINH is tested by verifying that the looped channels contain these signals during alternate 8-ms scans. The signaling stream, like the report stream, is simplex from each DIU and driven by an unbalanced source.

The UMB is the communication link from the DIC to the DIUs. It is via this stream that the DIC can exercise the DIUs and insert the remote alarms. The UMB line format is shown in Fig. 16. Data on the maintenance bus is frame synchronized to DIU clocks but is not distributed by decoding a TU. Instead, a DIU decodes six address bits, U0 to U5, to determine if the data is for it. A seventh bit, ALLUN, is used to address all DIUs simultaneously.

The function enable determines the action to be taken by a DIU. If the action to be taken is for a specific digroup, then the digroup identity will be flagged. The exercise fields are not coded, i.e., each bit of an exercise word is a specific common or digroup exercise state.

The UMB distribution to the DIUs is maintained by odd parity over all data in TS 127 within the DIC.

4.5 Exercises

The DIU contains a more comprehensive list of exercise routines than the digroup terminal unit does. Normally, exercises are used to test on a periodic basis the error source registers and matchers in the DIU. However, in the DIU a set of exercises tests the per-digroup equipment when the DIU is protection-switched. More specifically, the DIC has a set of diagnostic routines that tests for the ability to:

(*i*) detect framing errors,
(*ii*) reframe,
(*iii*) detect forced D2,
(*iv*) send forced D2,
(*v*) transmit and receive signaling.

For items (*ii*) and (*iii*), time limits are set on the response of the DIU so that deterioration of the F/R logic to respond properly can be detected.

V. DIGITAL INTERFACE CONTROLLER

5.1 Design objectives

The DIC represents the latest development in the evolution of the No. 4 ESS peripheral frame controllers.[4] Its design draws heavily on experience accrued over earlier transmission/switching designs, and capitalizes on microprocessor technology to allow flexibility to respond to change.

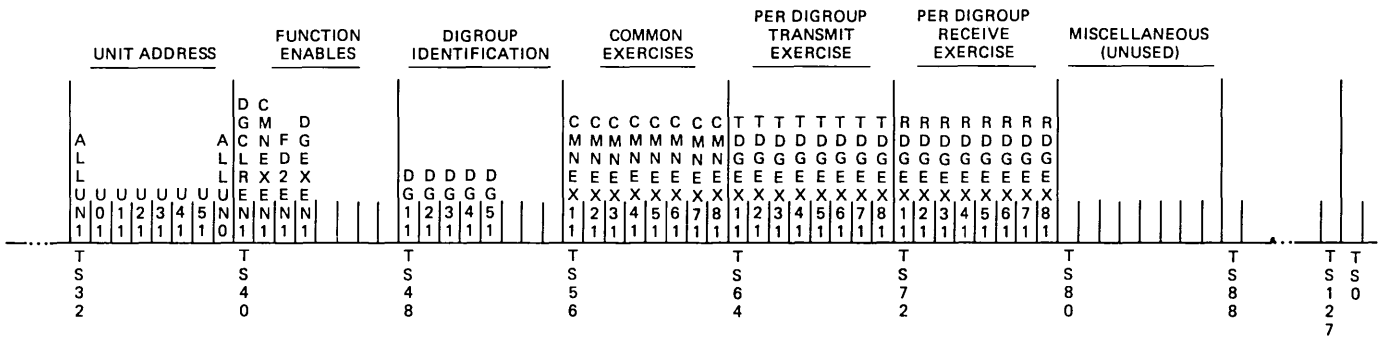


Fig. 16—DIU (DIC maintenance bus format).

Functionally, the DIC supports the DIUs in much the same way that the digroup terminal controllers and the related portions of the Signal Processor 2 supported the digroup terminal units. It provides for the collection, processing and distribution of supervisory and address signaling information, a reliable clock source, and an interface to the PUB of No. 4 ESS.

The DIC provides reconfiguration of the DIUs under fault conditions by controlling the protection switching of the spare units and appropriately redirecting signaling interchanges. The transmission facility maintenance and DIU maintenance are provided via a maintenance microcomputer developed in the controllers, around the *Bellmac**-8 microprocessor.

Many software considerations went into the design of the DIC. The repertoire of operational (call processing related) peripheral bus orders was chosen to be compatible with those of the Signal Processor 2. This minimized the associated call processing software development. The maintenance software in the 4E5 generic was dramatically restructured to allow for "intelligent" controllers and to minimize and modularize hardware dependent software. The DIC was designed to minimize generic DIU reconfiguration software, and controlled most DIU reconfiguration actions with frame resident "firmware."

The DIC was designed so that a significant portion of the controller hardware and diagnostic software could be used in common with the Peripheral Unit Controller (PUC) associated with the mass announcement system feature in 4E5.[5] This minimized overall development effort, and allowed the PC to capitalize on the larger scale of manufacture of the DIC. The maintenance software development for the PC was also facilitated, in that common hardware characteristics minimized differences in the frame-dependent software.

5.2 Controller architecture

Figure 17 shows the overall architecture of the DIC. Since a failure of a DIC could affect up to 3840 trunks, the controller is fully duplicated. Either controller can support all the DIUs, while its mate is being diagnosed and repaired. The two controllers are independently powered and are provided independent clock inputs (Master Timing Links or MTLs) from the associated TSIs or echo suppressor terminals. Two MTLs are provided per controller.

The controllers derive their internal timing from either connecting MTL, and each controller has its own independent countdown chain. Synchronization signals are provided by both controllers to the DIUs which select one and only one controller's timing to drive the internal

* Trademark of Western Electric.

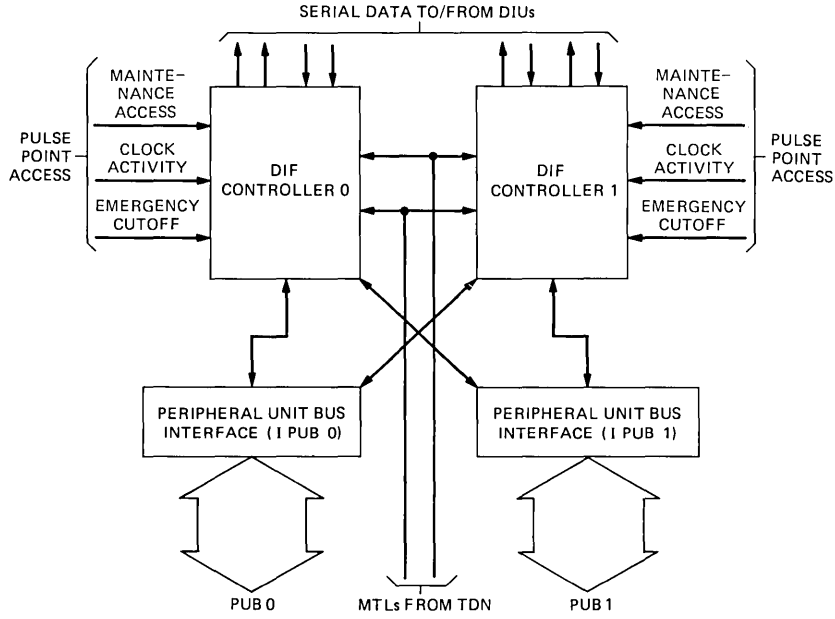


Fig. 17—Duplex DIC overview.

DIU functions. The ten signals provided to the DIUs allow their DS-120 outputs to be synchronous to the No. 4 ESS time division network and also determine the outgoing T1 line frequency of 1.544 Mb/s.

The PUB interface, which is also duplicated, terminates in the bus access circuits as shown in Fig. 17. This interface is fully configurable so that either controller can take its input from either bus, and can reply on either or both buses. In addition to the PUB, the 1A Processor is able to access the DICs via pulse points. These pulse points are employed for recovery actions, such as putting a controller in the maintenance state, selecting clocks, or disabling a controller's protection-switch capability. The pulse points are provided via independent signal processors for each controller.

The maintenance and operational data which pass between the DICs and DIUs are time-division multiplexed. Consequently, a transmission path is not required from each DIU for each signal. This results in simplified intraframe cabling and minimal select circuitry in the DIUs.

5.3 Interface to the peripheral unit bus

The DIC terminates the entire PUB. The bus consists of four duplicated bus groups:

- The PU enable/address bus, which conveys address information that determines which frame in the No. 4 periphery should respond to a particular peripheral order.

- The PU write bus (PUWB), which conveys the data to be accepted by the DIC and processed.
- The PU reply bus (PURB), which is used to return data to the 1A Processor.
- The PU control bus over which control and maintenance information is transmitted to and from the peripherals.

The bus access circuitry terminating PUB0 is independently powered from that terminating PUB1, and both are powered independently of either controller. In this way, a failure of either bus access circuit or either controller does not affect the balance of the controller and bus circuitry.

As part of the bus access circuitry, "bus clamps" are provided to prevent a faulty controller from "babbling" onto either bus. These clamps are controlled with a combination of the maintenance access pulse point and either the member interrogate or group interrogate bits of the control bus. When a bus or controller is powered down, the clamps are manipulated by power sequencing logic to prevent babbling.

The PUB signals incoming to the DIC are terminated in series receivers before passing on to the next peripheral frame. So that replacement of a receive pack does not interrupt the continuity of these signals, bypass resistors are provided. If the DIF is the last peripheral on the bus, optional terminating networks are employed.

5.4 Simplex controller architecture

Figure 18 shows the internal architecture of one of the duplicated controller circuits, commonly referred to as a simplex controller. PUB orders coming to the DIC are stored in the receive logic, and tested for validity. If the order is directed to this particular DIF and obeys the appropriate protocol, the receive logic interrupts the Executive Controller (EXEC). The EXEC routes the order to the appropriate function via the internal bus. If the order requires a reply, the data from the subject function is routed through the internal bus and into the reply logic. The reply logic then controls the appearance of the reply data on the PURB.

The function of the receive logic is to store the bits received from the PUB and test the received information for the format and addressing. In addition, it provides synchronization between the asynchronous PUWB data and the synchronous interface of the EXEC controller. The reply logic provides the necessary reply formating of the information returned to the 1A Processor via the PURB. There are a number of special reply bit fields (e.g., maintenance data) that are required of the DICs. The receive and reply logic provides the control signals to handle these special requests from the 1A Processor.

The internal bus structure of the DIC consists of a multiplexed bus. This bus structure was chosen to minimize fault susceptibility, to

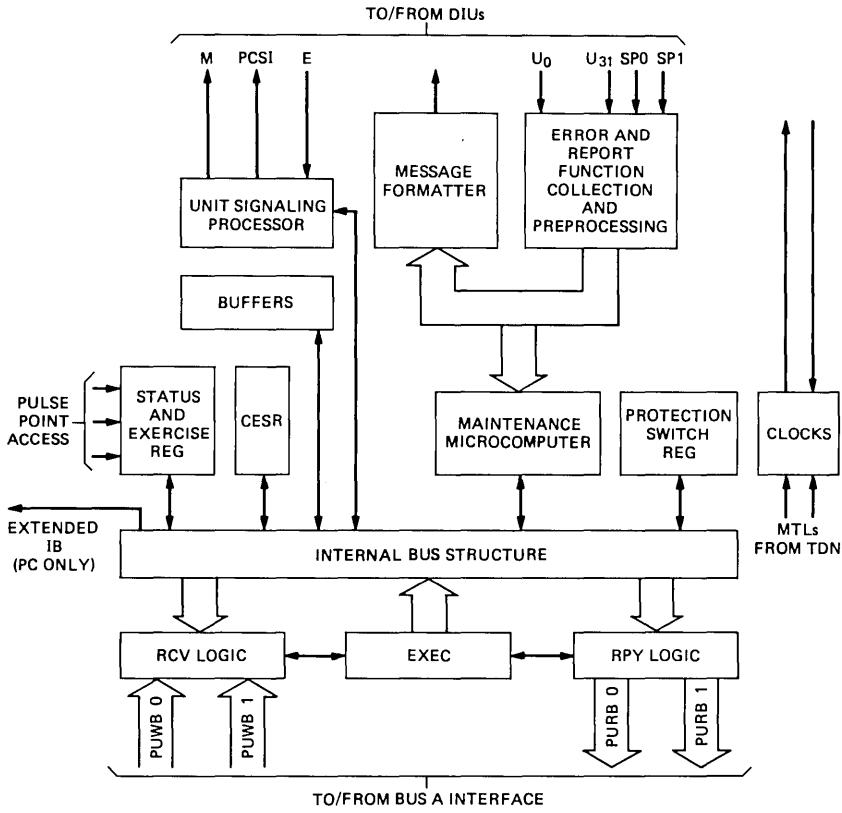


Fig. 18—Simplex DIC.

alleviate the necessity for multiple high-powered drivers, and to allow better control of the bus topology. All internal bus lines are looped back to the internal bus circuit pack driving them, for impedance termination and maintenance. The internal bus, which is under control of the EXEC, selects one of 16 possible ports or data sources to be placed onto the internal bus. Each port consists of 24 data bits and a parity bit. Each functional entity within the DIC is considered a port on the internal bus. The receive and reply logic is treated the same as the other functional entities within the DIC. This parallel high-speed structure allows flexible communication within the DIC.

The EXEC configures to the internal bus multiplexers with four bits of select information to specify which data of the 16 ports it will route through the internal bus. These four internal bus select leads are duplicated for maintenance. The EXEC also sends to the functions within the DIF six bits of source address, six bits of destination address, four bits of internal bus operation code, and an internal bus load pulse. The source address specifies which function will be placing the data

onto the internal bus. This allows for premultiplexing, where necessary, in the various functions. The six-bit destination address specifies the recipient of the data. The function addressed loads its internal register with the data on the internal bus upon receiving the load pulse. Each function also returns to the EXEC a source and destination acknowledge. This is an acknowledgment of a function's receipt of a valid source or destination for one of its registers. This acknowledge allows the EXEC to check the source and destination decoders in the various DIF functions.

The controller handles all internal communications in the DIC and communication with the 1A Processor. The EXEC is a bit-sliced, bipolar microprocessor. The microstore is modular with a maximum capacity of 4000 words of microprogram. The EXEC contains a 16-level priority interrupt control circuit, a 12-bit microsequencer, and an 8-bit arithmetic logic unit (ALU). Internal bus access logic allows the ALU to access 24-bit data from various DIC functions. Inputs to the EXEC consist of interrupt signals from various functions within the DIC. These signals indicate a request for a particular task routine to be executed.

The microprogram of the EXEC consists primarily of an interrupt handler and numerous special purpose task routines to control internal bus transactions. Interrupts from the receive logic (i.e., from the 1A Processor), from the maintenance microcomputer, from the unit signaling processor, and from a real-time clock (10 ms) source can all initiate such task routines. As an example of the microprogram execution, consider the receipt of an interrupt signal from the receive logic. The execution of an EXEC order can be broken down into three parts. The first of these three is the time from the presentation of the interrupt source until the interrupt is checked and detected. This is a noninterruptible state and consists of tasks that cannot be subdivided or "segmented." An example of a noninterruptible task is a bus transfer with an internal bus client with volatile data.

The second part of EXEC order execution is the time to transfer control to the interrupting routine. This is the time for the interrupt to be acknowledged, for control to be passed through the interrupt jump vector table, and for the resetting of the interrupt circuit. The third portion consists of the actual execution time of the requested function. The EXEC must complete the data transaction within a finite time window or the receive logic triggers a controller alarm. This is done as a sanity check against the EXEC.

In addition to interrupt initiated task routines, the EXEC contains microdiagnostics which are invoked by 1A Processor resident diagnostic programs. These microdiagnostics extensively check the ALUs, microsequence controller, and microprogram store of the EXEC.

The overall configuration of the DIC is determined by the data in the controller status register. Routing of PUB data, MTL selection, and controller selection are determined by the state of this register. The exercise register is used during diagnostics to exercise the fault-detection circuitry of the DIC and verify their ability to report. Under normal conditions, all hardware fault detector outputs are stored in the controller's error-source registers. Each major functional element in the controller has a local error-source register, and a summary of these are recorded in the primary controller error-source register (CESR). This register is interrogated by maintenance software during a fault condition to isolate the fault to a particular controller. Unlike some of its predecessors, a minimum of fault detection in the DIC relies upon cross controller matching. Many processing elements are duplicated within a simplex controller to assure autonomous maintenance even under simplex operation.

The disposition of the spare DIUs is controlled by the protection switch registers. The protection switch register outputs of the two controllers are logically ORed in the protection-switch equipment. Special cutoffs are provided to prevent a faulty controller from affecting a protection switch. The power sequencing circuitry activates these cutoffs (in addition to software control) when a controller is powered down.

The operational functions of E signaling reception, M signaling distribution, and PCSINH are controlled by the unit signaling processor. The unit signaling processor function includes the collection and distribution of supervisory signaling, as well as dial pulse reception and outpulsing. The signaling processing is determined by state translation firmware, and therefore can be modified to respond to changes in pulse width requirements or general timing changes. The signaling on all trunks is processed every 10 ms, as initiated by the EXEC in response to the real-time clock interrupt.

The results of signaling processing are reports which must be communicated to the 1A Processor. The SP deposits reports via the internal bus in one of four "buffers" or scratchpad RAM regions. These buffers are designated high priority, low priority, seizure, and digit buffers. The EXEC routes the data into the buffers, and administrates the appropriate read and write pointers.

The 1A Processor periodically polls the DIF, just like an SP2, to determine if any reports are present. If reports are present, the DIF acknowledges the poll, and call-processing software reads the buffers and either directs the unit signaling processor to continue processing that trunk, or proceeds to connect a path through the No. 4 ESS time-division network.

The DIC has a maintenance microcomputer (MMC) to maintain the

DIUs, aid in fault-recovery actions, or participate in diagnostics of other portions of the controller. The MMC has a byte-addressable serial bus with which it sends information to the DIUs. The DIU report streams are converted to parallel format and deposited in the MMC RAM via direct memory access (DMA). Following a complete DMA cycle, which occurs once every 32 ms, the preprocessing hardware generates an interrupt which initiates processing of the DIU alarm data. In this way, DIU common alarms are "hit timed," local and remote (T1) alarm indications are detected, and facility reports such as slip, out of frames, and error rates are prepared.

The MMC is built around a *Bellmac*-8 central processing unit (CPU) which is duplicated within a simplex controller for fault detection. The MMC executes tasks under direction of an interrupt-driven operating system called "OS8" written in the C programming language. All of the MMC resident application programs are written in C as well, with the exception of real-time intensive unit alarm processing tasks which are programmed in assembly language.

The 1A Processor can access any memory location in the MMC via DMA. Autonomous reports from the MMC are deposited in a portion of buffer RAM designated the maintenance buffer. Like the operational buffers, this is administrated by the EXEC and periodically polled by the 1A Processor.

The MMC is capable of (macro) peripheral order expansion, under direction of the 1A Processor. In this way, complex tasks such as initialization and unit reconfiguration are relegated to the MMC and the 1A Processor maintenance software is considerably simplified. This was consistent with the restructuring of maintenance software, a significant part of the 4E5 generic.

Each simplex DIC independently derives clocks that drive the controller's circuitry, and which are sent to the DIUs. The clocks are derived from the MTLs originating in the time-division network (TDN) of No. 4 ESS. In this way the DS-120 signals of the DIUs are fully synchronous with the timing of the associated TSI.

To preserve commonality between the DIC and the PC, two major features were a part of the design. First, a port of the internal bus was equipped with bidirectional drivers/receivers to form the extended internal bus. Since this extended bus is tristate, and any client output shorting low would affect all clients, special extended bus cutoff signals are provided. These can be used to quarantine any extended bus client and allow the PC to continue operations.

The PC does not have DIUs, so all DIU-related hardware was packaged in the lower shelf of the DIC. By deleting this shelf and the associated circuit packs, the PC did not bear this unnecessary cost burden and 80 percent of the DIC hardware is still applicable to the PC.

VI. MAINTENANCE AND RECOVERY STRATEGY

6.1 Overview

The maintenance and recovery strategy employed by No. 4 ESS is structured into five levels. These levels are phase level, interrupt level, interject level, base level maintenance, and diagnostic isolation. Of these levels, the first four are related to identifying and isolating the failing subunit and recovering the system. The fifth level attempts to isolate the failures within a frame and is used to maintain the operational soundness of the frame. As the service impact of a failure increases, the level of response to that failure is altered. Within this system, each frame can specify the initial level of recovery to be associated with a failure.

With the introduction of the DIF, an additional level of maintenance was added. The new level, routine exercise, is used to detect faulty DIUS before they have been able to adversely affect the service they provide. In the next few sections we present a description of the maintenance and recovery strategies specifically used to support the DIF. Table I shows the maintenance and recovery strategies to be covered.

6.2 Phase level

Phase level is the severest of the recovery strategies, in that service interruption may be experienced during its execution. Phases are stimulated when a failure seriously hinders the effective operation of the system or if reinitialization of part or all of the system is required for recovery. Phase level is further segmented into four levels. The most drastic, level 4 (phase 4), is the level of phase recovery we will concentrate upon.

Table I—Maintenance and recovery software hierarchy

Phase Level
System Failures Action—reinitialize system
Interrupt Level
Controller (DIC) errors Action—DIC configuration
Interject Level
Unit (DIU) errors Action—DIU configuration
Base Level
Normal operating level Action • Base level maintenance • Diagnostics

A phase 4 cleanly reinitializes the entire No. 4 ESS office, leaving failing frames out-of-service. In previous generic issues, frame initialization during level 4 executed in a strictly serial manner and resulted in simplex operation. In the case of the DIF, a new approach to level 4 initialization was implemented. The DIFs within an office are initialized to duplex operation with the initialization of all DIFs being done in parallel.

To a large degree, this change in initialization design can be credited to the incorporation of a maintenance microcomputer (MMC) within the DIF. The MMC is responsible for the bulk of the frame initialization function. The MMC accepts orders from the 1A Processor, which it expands and executes in addition to executing its internally generated maintenance actions. The internal expansion of 1A Processor orders by the MMC allows the 1A Processor to issue a command to the DIF and have an entire function executed, while the 1A Processor does something else. Thus, during phase 4, a small amount of configuration is performed on the first DIF, followed by the issuance of a frame initialization order. As the frame "init" order is processed by the first DIF's MMC, the 1A Processor directs its attention to the next DIF in the office. This sequence is repeated until all DIFs have been initialized. Upon the completion of the frame "init" order, the MMC returns a success or failure indication. The 1A Processor uses this information to determine the resultant configuration of each DIF (duplex, simplex, or duplex failed). The final result of the phase 4 is a stable operating office.

6.3 Interrupt level

In developing the DIF, it was concluded that errors within the DIC should be reported separately from those errors associated with DIUs to simplify the resolution of faults. It was also decided that DIC errors should be reported at a higher level to prevent the possible loss of all the DIUs in the alarming frame. The F-level interrupt is the mechanism used by the DIF to notify the 1A Processor of failures associated specifically with its controller circuitry. It is the initial level of recovery associated with a DIF controller failure.

Let us suppose that a No. 4 ESS office is operating stably. Further, suppose that a DIC in that office experiences a problem. This problem may be due to a hard failure, the loss of a circuit pack, or it may be caused by a transient failure condition. In any case, the operation of the DIF has been disrupted and needs to be corrected. An interrupt (F-level) is the mechanism used to inform the 1A Processor of the problem and request action to resolve the failure.

A failure detected within a DIC results in the setting of an ESR bit(s) associated with the failed circuitry. The setting of this bit(s) stimulates

the F-level interrupt. The recognition of this interrupt at the 1A Processor invokes the DIF interrupt recovery package, DIFRINTR. It is the responsibility of DIFRINTR to determine the appropriate frame recovery action based upon the error-source signature, the initial configuration of the frame, and the relative frequency of reported ESR bits from the frame.

The actions available to DIFRINTR are basically three. DIFRINTR can decide that the appropriate recovery action is the restoral of the alarming controller. This may be dictated by the transient nature of the failure or the simplex configuration of the frame. DIFRINTR can request the listen-only removal of the alarming controller followed by a diagnostic. The majority of the failures occurring in duplex frames are handled using this option. Removing a controller to listen-only allows it to be isolated from its mate, yet kept entirely up to date until the diagnostic begins. The listen-only removal request allows DIFRINTR a second chance if the wrong decision as to which controller to remove was made. The third option available to DIFRINTR is the zero-start restoral of the mate. This option is executed as a final attempt to preserve the operation of the frame. It terminates all stable and transient cells that are being handled by the alarming DIF. The alarming controller is removed from service while the mate controller is completely initialized. Failure of this option results in the duplex failure of the alarming DIF.

6.4 Interject level and base level maintenance

Both the interject and base level maintenance (BLM) reporting mechanisms are additional methods used to inform the 1A Processor of a problem and request action to resolve the failure. The particular levels of recovery in relation to the DIF are strictly reserved for DIU-associated failures. Within the DIF, these failures are monitored and reported by internal frame processes.

The maintenance of each DIU is performed internal to the DIF frame within its MMC complex. On a 32-ms cycle, data associated with the performance and health of each DIU in the frame is written into the MMC memory spectrum. The MMC real-time DIU maintenance firmware scans this data in search of errors and reports both common alarms and digroup alarms to the 1A Processor.

Common alarms indicate a malfunction affecting the operation of 120 trunks, an entire DIU. To preserve the operation of this alarming subunit, these failures need to be detected quickly. For this reason, they are scanned for every 32 ms. The discovery of a common alarm places that DIU on the hit timing list. If the alarm exists for three consecutive scans (96 ms) of the DIU data, it is classified as a hard failure and reported to the 1A Processor. Common alarms are reported

via the Autonomous Peripheral Unit Trouble bit, in the frame's primary ESR. This causes interject processing to be scheduled by the 1A Processor. Reporting common alarms via an interject is rapid enough (served within 20 ms) to allow the recovery software the option of protection switching a spare DIU for the alarming DIU. Protection-switching preserves stable calls active in the alarming DIU.

Digroup alarms are less critical with respect to trunk effect than common alarms. These failures affect the operation of only a single digroup (24 trunks) within a single DIU. These indicators are scanned once every 384 ms. The lower-scanning frequency for these failures results from the requirement that all DIUs be checked for common alarms every 32-ms cycle. The time remaining in each cycle after this checking permits the scanning of three DIUs for digroup failures. Even with the longer scan time, digroup failures are hit timed for three counts before being labeled hard. Hard failures of the digroup variety are reported via a BLM. A BLM is less system-affecting than an interject and is the result of a failure report being retrieved from the MMC maintenance buffer on base level. The BLM removes the failing digroup from operation.

6.5 Diagnostics

Once the recovery software (outlined above) has isolated an alarming DIC or DIU, the DIF diagnostic is called upon to further resolve the failures. It is the responsibility of the diagnostic to specifically determine which circuitry, if any, has failed. Toward this end, the DIF diagnostics are structured into phases. Each phase is responsible for determining the operational fitness of a specific section of the frame's circuitry. Thus, the operational fitness of the entire frame is based on an all-tests passed condition being achieved by all phases of the DIF diagnostic. A diagnostic phase should not be confused with the system phases discussed earlier.

The DIF controller diagnostic, for example, consists of 23 distinct phases. The phases of the DIF diagnostic are ordered such that the fitness of the frame is checked using an "onion peeling" philosophy. In peeling an onion you start at the outside and work your way inward. The DIF diagnostics are implemented in much the same way. The early phases of the DIC diagnostic start at the PUB and clock interfaces to the frame and progress inward. The early phases confirm the 1A Processor's ability to gain access to the frame. Later phases attempt to verify the functional integrity of the executive processor and check the accessibility and operation of the DIC's internal bus. At this point in the diagnostic, the front end and the bus to the workings of the controller have been verified. The remaining phases of the diagnostic determine the soundness of the maintenance microcomputer complex

and the signal processor complex which are accessible via the internal bus.

The end result of a diagnostic analysis of the controller or subunit circuitry will be the isolation of the circuit pack(s) that have failed. Upon replacement of the faulty equipment, the diagnostic is executed again to ensure that the problem has been corrected before returning the DIC or DIU to service.

6.6 Routine exercise

In addition to the other functions detailed previously, the inclusion of the MMC complex has afforded the DIF the capability of executing tasks (functions) on a routine basis. When the MMC has no requested tasks to execute, it sequentially executes the tasks that have been identified as routine. To date, two tasks, DIU exercising and auditing of the protection-switch state, are run routinely within the DIF.

The DIU exercise routine task uses test vectors to functionally test the operation of each DIU. If a DIU does not respond properly to the test sequence, the failure is reported via BLM. The protection-switch state audit routine task checks for the proper configuration of the spare units. A failure of the audit results in an interject.

The routine exercise capability allows the DIF the opportunity to functionally verify the operation of critical hardware segments routinely. This constant monitoring provides early error-detection and correction.

VII. CONCLUSION

The development of the DIF was the result of the concerted effort of the components, switching, and transmission organizations in Bell Laboratories and of the components and system organizations in Western Electric. A comparison of the DIF with the DT/SP2, which it replaces, indicates a significant reduction in cost, power, and space requirements, an improvement in reliability, and a reduction in the installation effort. By the end of 1980, it is expected that over 500,000 No. 4 ESS terminations on DIF will have been deployed.

REFERENCES

1. J. F. Boyle et al., "Transmission/Switching Interfaces and Toll Terminal Equipment for No. 4 ESS," B.S.T.J., *56,* No. 7 (September 1977), pp. 1057–97.
2. T. J. Cieslak et al., "Software Organization and Basic Call Handling," B.S.T.J., *56,* No. 7 (September 1977), pp. 1113–38.
3. W. L. Harrod and A. G. Lubowe, "The *BELLPAC*™ Modular Electronic Packaging System," B.S.T.J., *58,* No. 10 (December 1979), pp. 2271–88.
4. R. A. Bruce, P. K. Giloth, and E. H. Siegel, Jr., "No. 4 ESS—Evolution of a Digital Switching System," IEEE Trans. Commun., *COM-27,* No. 7 (July 1979), pp. 1001–11.
5. T. W. Anderson et al., "No. 4 ESS—Mass Announcement Subsystem," B.S.T.J., this issue.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 60, No. 6, July–August 1981
Printed in U.S.A.

No. 4 ESS:

Evolution of the Software Structure

By P. D. CARESTIA and F. S. HUDSON

(Manuscript received August 26, 1980)

This paper examines two different examples of how No. 4 ESS software has evolved through restructuring to meet the needs of the changing No. 4 ESS environment. The two software areas that underwent varying degrees of incremental restructure are Call Processing and Fault Recovery. We characterize the pre-restructure architectures, discuss the motivation and rationale which led to restructure, and present and evaluate the post-restructure architecture for each software system.

I. INTRODUCTION

No. 4 ESS, the largest-capacity electronic switching system ever developed by the Bell System, was developed to meet specific objectives of capacity and reliability.[1] To meet these objectives, the No. 4 ESS was designed using new hardware technology and a comprehensive stored program. The primary objectives of the initial No. 4 ESS program design were:

(*i*) real-time efficiency,
(*ii*) simple human interface,
(*iii*) defensive design,
(*iv*) ease of modification.

Since its initial service date, the No. 4 ESS has released a new generic software package approximately once a year incorporating major new hardware and software capabilities. Each new generic was built upon the previous generic. As the number of features provided by the No. 4 ESS grew, it became increasingly more involved in certain areas of software to accommodate new features without impacting existing features. The amount of time spent in regression testing had the potential for becoming an ever growing part of the software development interval, thus, increasing new feature development cost.

New software development methodologies that used top-down de-

sign and structured programming techniques gained wider use in the No. 4 ESS software development process. These rigorous approaches to software design effectively pointed out where certain areas of No. 4 ESS software could be improved.

A new high-level programming language, EPLX, was introduced that supported structured programming techniques and provided increased program readability, modularity, and maintainability.

Given the continuing demand for new features, the design objectives for software development had to be enhanced to place greater emphasis on ease of modification and flexibility to reduce the cost and development time of new system features. This increased emphasis along with new software development methodologies and programming languages led to a selective restructuring of areas of No. 4 ESS software, which were to be affected the most by new feature development.

The No. 4 ESS software areas which became major candidates for restructuring were call processing and fault recovery. Sections II and III give the restructuring process for these two software areas.

II. CALL PROCESSING RESTRUCTURING

To better understand the motivations and rationale for restructuring, we review the call processing architecture prior to restructure.[2] It should be made clear that the entire call processing system was not restructured. Instead, an incremental restructuring occurred which focused primarily on the task programs responsible for call handling actions. We discuss the task programs in light of their original design and their deficiencies. We give the motivation for and approach to restructure, along with a discussion and evaluation of the new architecture.

2.1 Call Processing before restructure

When the No. 4 ESS cutover in 1976, it provided the capability to interface with both local and toll switching machines, to function as a tandem and/or toll switch, and to interface with all of the trunk signaling types listed below:

(*i*) Dial Pulse (DP)
  (*a*) Delay Dial Start Dial
  (*b*) Immediate Start
  (*c*) Wink Start
(*ii*) Multifrequency (MF)
  (*a*) Wink Start
  (*b*) Delay Dial Start Dial
(*iii*) Common Channel Interoffice Signaling (CCIS)

The No. 4 ESS also provided the Centralized Automatic Message

Accounting (CAMA) function for trunks using dial pulse or MF address signaling.

The call processing programs were initially structured in a three-level hierarchy as shown in Fig. 1. The task dispensers (Level 1), which were entered directly from Executive Control, provided the interface for external stimuli received from the signaling hardware (Signal Processors and CCIS terminals) and the interface for internal stimuli received from timing and queuing programs. Executive Control provided both high- and low-priority entries to the task dispensers, which used the entries to poll the buffers in the signaling hardware for high- and low-priority reports and to determine if time-out conditions existed. If reports or time-out conditions existed, they were dispensed sequentially to the appropriate task program for processing. The task dispensers remained in control until all relevant external or internal stimuli had been processed or until an overload threshold had been reached that limited the amount of activity processed by the system during any base cycle.

The task programs (Level 2) performed the specific actions that switched calls. Task programs were entered from the task dispensers in response to a particular stimulus. The task program investigated

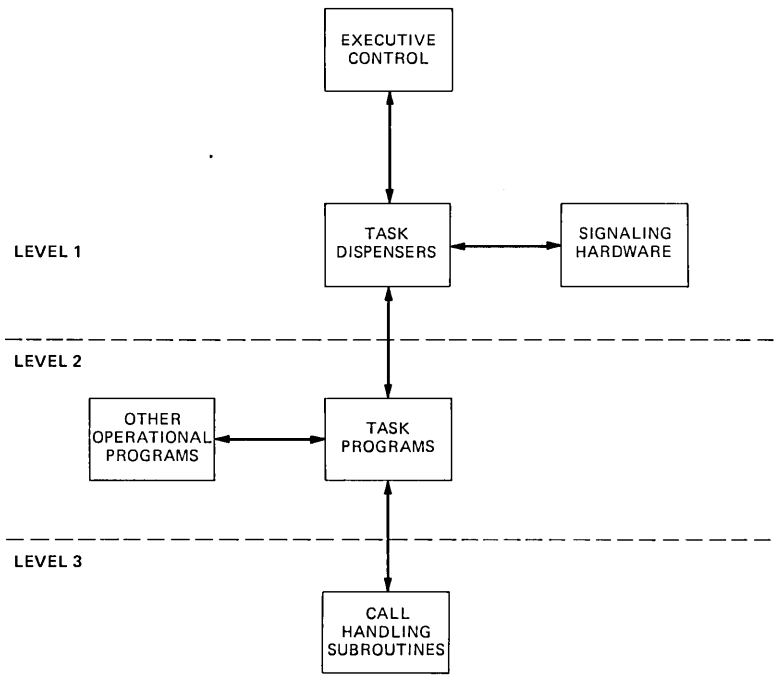


Fig. 1—Initial No. 4 ESS call processing architecture.

the present state of the call and, depending upon the present state and the stimulus, initiated the appropriate actions to advance the call to a new state. The present state of a call can be determined from the call register (CR) or trunk register (TR). The CR is a 64-word block of call store memory used for temporary storage of information during call setup. CRs are not dedicated on a per-trunk basis. Instead, there is an engineered number of CRs per office which are link-listed together. The TRs are two-word blocks of call store memory assigned on a per-trunk basis. TRs contain dynamic information about the current state of the trunk or call.

Certain repetitive or specialized call handling functions were designed as subroutines (Level 3) so they could be accessed by several task programs. Examples of call handling subroutines are seizing and initializing a CR, connecting incoming and outgoing trunks, hunting a service circuit, or pegging a traffic counter.

The task programs also interfaced with other operational programs during the processing of a call. These interfaces were established to allow independent software development of major operational functions such as audits, translations, network management, and trunk maintenance. Where these functions overlapped during the processing of a call, clearly defined interfaces were established with the task programs.

2.2 The Call Handling task programs

The Task Program block in Fig. 1 shows the set of task programs as shown in Fig. 2. The task programs were organized on a signaling-type/type-of-trunk basis and separated into incoming trunk and outgoing trunk programs. Each program was state driven and was responsible for acting on the stimuli dispensed from the task dispensers. The incoming trunk programs processed internal and external stimuli associated with the incoming trunk part of a call, and the outgoing trunk programs processed internal and external stimuli associated with the outgoing trunk part of a call. Internal stimuli were associated with events such as timing or queuing reports. External stimuli were physical trunk signals. Each incoming trunk program also had an interface with the Digit Reception and Analysis Programs, which were responsible for determining the outgoing routes for the call based upon the dialed digits. Digit sending to a large degree was part of the outgoing trunk programs.

The task program architecture arose primarily because of the means of communication with the signaling hardware. Communication with the signaling hardware was at a physical signal level (off-hook, on-hook) rather than a logical signal level (seizure, answer). Therefore, the signaling protocol for a specific trunk was required very early in

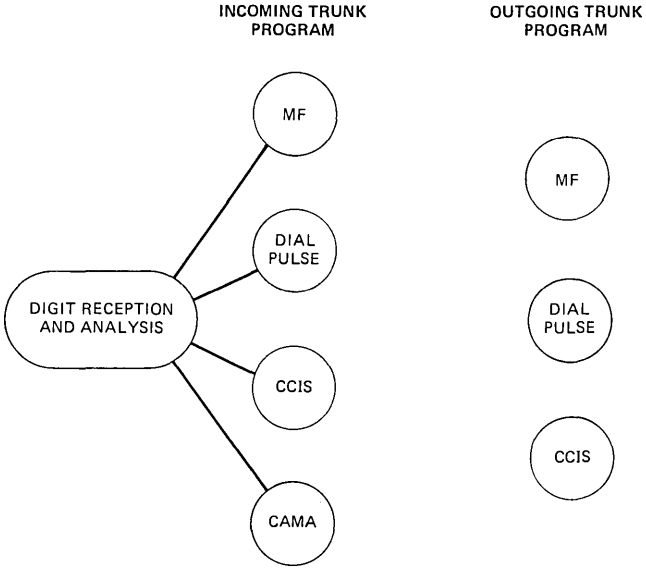


Fig. 2—Detailed view of task programs.

the processing of trunk signal reports. Rather than convert the physical signal into a logical signal prior to dispensing reports to task programs, the task programs were designed to handle the physical signals on a signaling-type/type-of-trunk basis. This approach resulted in a set of programs that were organized on an incoming and outgoing trunk basis for each signaling type/type of trunk.

Each program processed stimuli associated with its particular signaling type. However, there were always points in call setup where an incoming and outgoing trunk were involved in the call. They could be of the same or different signaling types. A stimuli at these stages of a call usually required actions by both the incoming and outgoing trunk programs. The design approach was to take one of two actions: (*i*) do whatever processing is required by the incoming trunk program, then pass control to the outgoing trunk program or vice versa, (*ii*) have the incoming or outgoing trunk program process the signal for both trunks associated with the call. The latter approach resulted in task programs that no longer contained processing logic for a single signaling type or for incoming or outgoing trunk. Incoming trunk programs made decisions based upon the type of outgoing trunk associated with the call and vice versa. For example, the CCIS task programs contained MF and DP signaling logic, etc. This approach was generally taken to save real time or to minimize program interfaces.

The drawbacks to such a task program design were: (*i*) proliferation

of decisions; (*ii*) duplication of program functions; (*iii*) dilution of program cohesion; and (*iv*) loss of independence between incoming and outgoing trunk. In some cases, task programs were call controllers and in others, single trunk controllers. The interfaces between incoming and outgoing trunk program became many and complex as shown in Fig. 3. The task program interfaces with the other operational programs further complicated the picture.

2.3 Motivation for restructure

With the development of the 4E3 generic for the No. 4 ESS, call processing was enhanced to provide the International Gateway Exchange Feature. This new feature required the addition of CCITT No. 5 and CCITT No. 6 signaling capabilities to call processing. Two new incoming and outgoing task programs were required along with the modification and retest of all existing task programs. Rather than add these signaling types to the existing architecture, thus further complicating an already complex structure, we considered restructuring the call processing task programs.

The goal of restructuring was to minimize the drawbacks of the current design, while at the same time, to minimize the effects of restructure upon the existing task programs which were known to be real-time efficient and virtually error-free. Eliminating duplication, strengthening program cohesion, and true separating incoming and

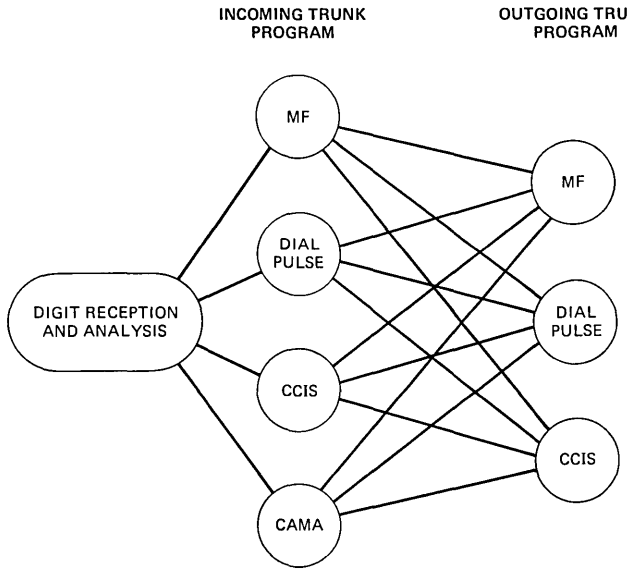


Fig. 3—Task program internal interfaces.

outgoing trunk processing were of special importance since future generics are very likely to require additional call handling task programs for new signaling types.

2.4 Approach to restructure

The key problem to be resolved with the restructure effort was how to make incoming trunk and outgoing trunk programs truly independent. The process began with formulating a universal call (shown in Fig. 4) and identifying the call events that must be processed to complete the call. At this point no effort was made to distinguish incoming trunk from outgoing trunk. The main focus was on the overall call. Seven call events were identified:

(*i*) Origination,
(*ii*) Digit reception,
(*iii*) Outgoing trunk selection,
(*iv*) Origination on the outgoing trunk,
(*v*) Digit sending,
(*vi*) Receive answer,
(*vii*) Receive disconnect.

The architecture began to materialize as a result of functionally decomposing the universal call into three sequential call stages:

(*i*) Setup—that part of the call from seizure on the incoming trunk through digit sending and connection of the incoming and outgoing trunks.

(*ii*) Post Setup—that part of the call during which time a voice path is connected, while awaiting answer and in the talking state.

(*iii*) Clearing—hardware and software trunk idling sequences after call termination.

The three sequential call stages were further decomposed into incoming trunk (ICT) and outgoing trunk (OGT) processes, resulting in the functional decomposition shown in Fig. 5.

The final phase of the process addressed the basic problem of isolating ICT and OGT processing. A new program function was created to consolidate the communication interfaces between ICT and OGT task programs and to oversee common call related functions. This program was called the Report Dispenser. Its inclusion in the new architecture made it possible to remove from the task programs any trunk signaling logic dealing with the other trunk involved in the call and to create trunk handlers.

The Report Dispenser was the single most important addition to the call processing architecture, because it introduced the use of logical signals as the means of communication between trunk handlers. Trunk handlers could now communicate with the Report Dispenser without involving another trunk handler. Incoming trunk and outgoing trunk

Fig. 4—Universal call flow diagram.

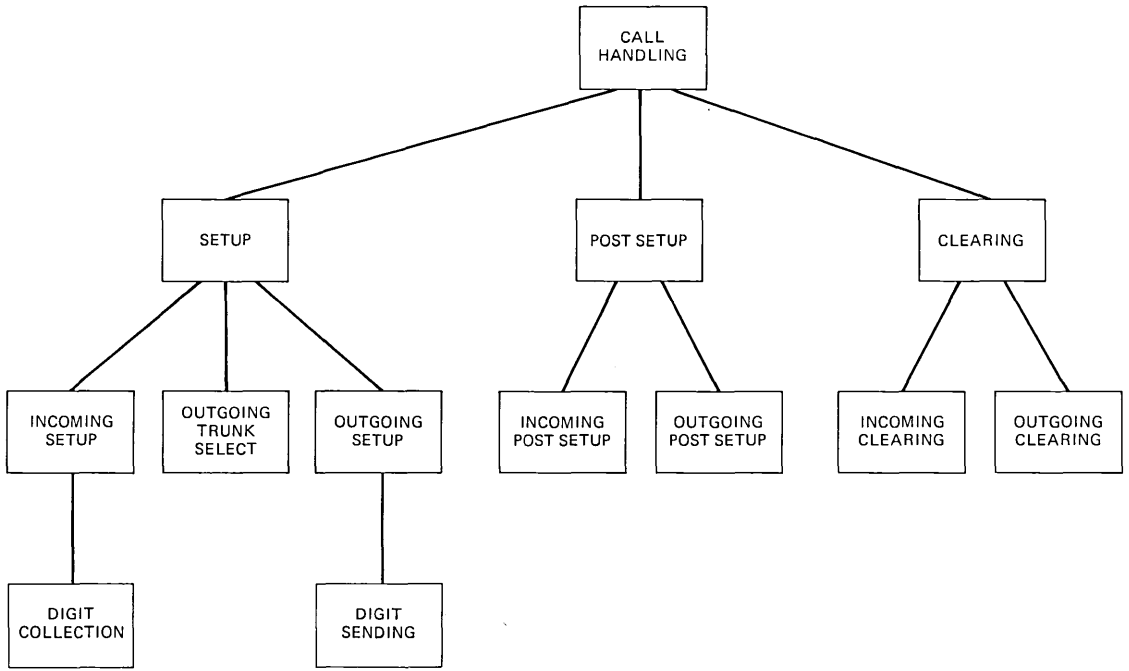


Fig. 5—Functional decomposition of a universal call.

processing became independent. The complex interface between incoming and outgoing trunk programs had been replaced with a standardized interface that used logical signals as a means of communicating with a central point of control wherein call-related decisions requiring knowledge of the other trunk were made. The new architecture became one where task programs/trunk handlers communicated with trunk circuitry via physical signals and communicated with other trunk handlers via logical signals.

2.5 New architecture overview

The primary features of the new architecture are as follows:

(*i*) The splitting of call handling into parallel real-time processes (finite state machines), which control states of the incoming trunk, the outgoing trunk selection process, and the outgoing trunk.

(*ii*) The consolidation of communication decisions, which link these finite state machines in a program called the Report Dispenser.

(*iii*) The identification of a subset of call handling functions that can be implemented as subprocesses (submachines) under control of incoming or outgoing trunk handlers. These functions were common to most calls and relatively independent of signaling type. They are digit reception and digit sending.

An architecture based upon trunk handlers is advantageous from the standpoint of minimal impact upon the existing call processing task programs. The basic logic of the task programs can be maintained; the changes are limited to separating incoming and outgoing trunk functions, eliminating redundant code, and interfacing the task programs with the Report Dispenser. Figure 6 illustrates the major modules in the new architecture and the control hierarchy.

All task dispenser reports are made on a trunk state basis. This means that report dispensing is based strictly on the state of one trunk involved in a call to the trunk handler responsible for handling that type of trunk.

All internal and external stimuli are dispensed by the task dispensers in the same manner as existed in the pre-restructure system. A new task dispenser was added as part of the restructure effort to interface with the CCITT No. 6 signaling terminal.

When a trunk handler receives a physical signal from the task dispenser it takes whatever action is appropriate and then reports a logical call event to the Report Dispenser. The Report Dispenser determines the next call action to initiate based upon the logical event and may invoke per call common functions, such as outgoing trunk selection, or invoke the other trunk handler involved with the call. When the signal has been completely processed, control is returned to the task dispenser via the Report Dispenser and the trunk handler

that initially received the stimulus. In summary, a physical signal is passed to the trunk handler, which converts the signal to a logical signal (answer, disconnect, etc.) based upon the state of the trunk. The logical signal now becomes the stimulus to the Report Dispenser to stimulate further call processing actions.

The block called Common Call Functions in Fig. 6 consolidates many common call related functions and interfaces which in the pre-restructure architecture were spread throughout the task programs. Many of the interfaces with the other operational programs are consolidated here.

The Digit Reception and Digit Sending functions appear as submachines under the incoming and outgoing trunk handlers. They are programs which are invoked by the trunk handlers and are logical rather than physical signal driven. Task dispenser reports are directed to these submachines and not to the trunk handlers. This allowed for efficient real-time execution in the processing of these reports and

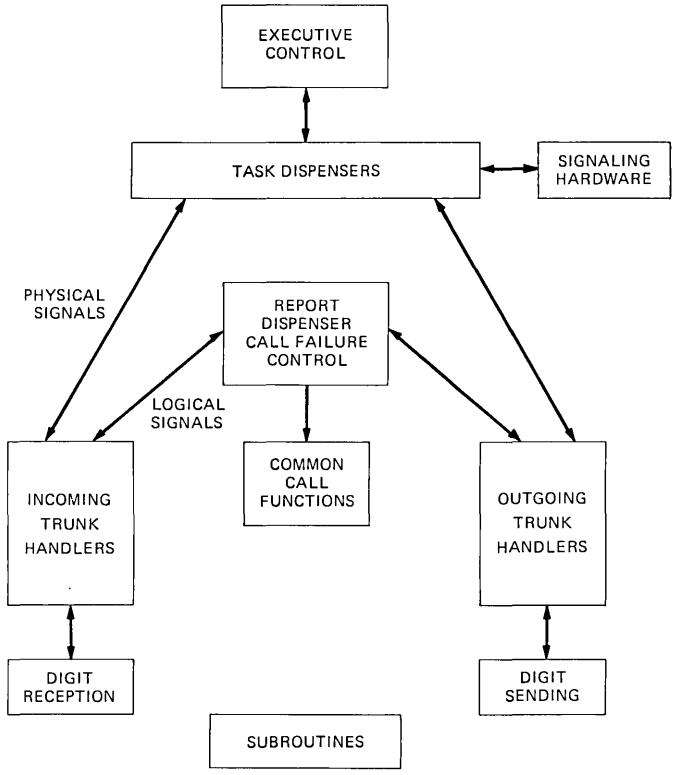


Fig. 6—No. 4 ESS call processing architecture.

does not burden the trunk handlers with detailed knowledge of how the submachine performs its function.

Call Failure Control has the same relative position in the new architecture as the Report Dispenser and is responsible for controlling the clearing of incoming and outgoing trunks as a result of Ineffective Attempts, i.e., calls that are not successfully completed.

2.5.1 A simple call

To more clearly understand the structure, interfaces, and control, we describe a simple DP-to-DP call. We incorporate only those events needed to successfully complete the call because the picture becomes more complicated when call anomalies are taken into consideration. The scenario is based upon the universal call diagram. Figure 7 represents the call flow diagram for the call. Incoming Trunk actions are represented along the top horizontal axis. Logical events are reported to the Report Dispenser, which then communicates these call events to the OGT. Outgoing trunk actions are represented along the bottom horizontal axis. The call actions progress in sequence from left to right.

The call begins with the receipt of an off-hook origination on the idle ICT. The physical off-hook signal is passed from the task dispenser to the ICT handler which prepares for digit collection. When the ICT is ready to receive digits, an integrity check signal is sent backward toward the originating office. The next ICT action is to receive digits. This action is performed by the Digit Reception Program. The Digit Reception Program also analyzes the digits to determine the outgoing trunk group for the call. The Report Dispenser is now notified that the call is ready for OGT selection. The Report Dispenser invokes the OGT selection program which is a common call function. After a successful return from the OGT selection program, the Report Dispenser invokes the OGT handler. The first OGT handler action is to seize the OGT. After seizing the trunk, the Report Dispenser is informed that seizure is complete. For this call, no ICT action is required at this point. Action is required if the ICT is CCIS or CCITT No. 6. This knowledge resides only within the Report Dispenser. The OGT handler waits for receipt of the integrity check signal from the far end office indicating readiness to receive digits. The OGT handler invokes the Digit Sending program, which deletes or prefixes digits to the dialed number and controls the outpulsing process on the OGT. When all digits have been outpulsed, control passes back to the Report Dispenser indicating the outgoing part of the call is complete. At this point, the call moves from the setup stage to the post setup stage. The CR is released and the voice path between ICT and OGT is completed. Both actions are common call functions. Each trunk handler places itself in the waiting-for-

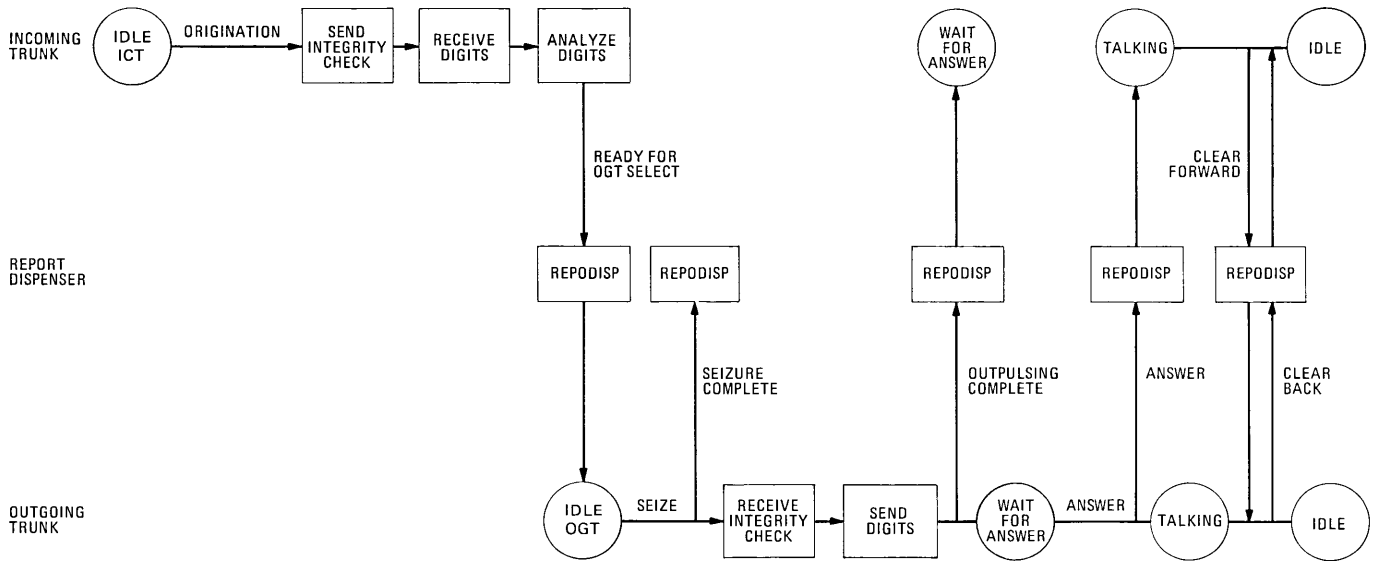


Fig. 7—Dial pulse-to-dial pulse call flow.

answer state. The next signal the No. 4 ESS expects to see is off-hook answer on the OGT or on-hook clearforward on the ICT, should the originator disconnect. In the case of the answer signal, the OGT handler processes the signal for the OGT and reports the logical event to the Report Dispenser, which in turn, passes the event to the ICT handler for processing. The next event in the call will be an on-hook disconnect on either trunk.

If the ICT receives an on-hook clearforward, the ICT handler reports the event to the Report Dispenser which invokes the ICT and OGT clearing routines to idle the trunks. If an on-hook clearback signal is received by the OGT, the OGT handler passes the event to the Report Dispenser, which invokes the ICT handler to send a clearback signal on the ICT. A clearback does not cause the call to be idled. A clearforward must be received to idle the call.

This simple example of the DP-to-DP call demonstrates how the Report Dispenser isolates the ICT and OGT from having knowledge of the other. We can then extend this example to cases where the ICT and OGT are of different signaling types and show that by communication with the Report Dispenser using logical call signals any type of ICT can interwork with any type of OGT, given the necessary logical signals have been defined.

2.5.2 The report dispenser

Communication between the trunk handlers is consolidated in the Report Dispenser. When a trunk handler detects a logical event that may be significant to the other trunk handler on that call, it reports that event to the Report Dispenser. The Report Dispenser determines the other trunk handler on the call and passes control to the appropriate trunk handler. This consolidation of what are primarily signaling type decisions about the other trunk handlers involved in the call results in an overall reduction of code and simplifies the addition of new signaling types. The addition of a new signaling type to this call processing system obviously involves the design and development of an ICT and OGT handler. However, if the new signaling type does not require the addition of any new logical signals, then the Report Dispenser only requires slight modification to include the new signaling type.

The trunk handlers communicate with the Report Dispenser by means of logical signals and pass additional data with the CR and TR. The signals are divided into two categories: setup and post setup. Setup signals are passed to the Report Dispenser, along with the CR, during the setup stage of a call. The signaling type of the ICT and OGT and the state of the call are stored in the CR. Based upon CR data and the logical signal, the Report Dispenser makes the decision on what to do next in the processing of the call. Post setup signals are passed to

the Report Dispenser in the post setup stage of the call, which is after the CR has been released. The Trunk Scanner Number (TSN), which identifies the ICT or OGT, is passed along with the logical signal. Through data translations using the TSN, the signaling type of the trunk and the TR are found. Based upon the signaling type of the trunk and the logical signal, the proper next step in the call can be taken. This mechanism then allows ICT and OGT handlers to have no knowledge about the other trunk involved in the call. That knowledge, along with the knowledge of the state of the call, resides within the Report Dispenser.

There are several functions in call processing that are call-event dependent and signaling-type independent. The Report Dispenser provides a central point for calling procedures associated with these common call functions. The common call functions include:

(*i*) Voice path setup and take down.

(*ii*) Interfacing with the No. 4 ESS Service Observing System, if it is active on a call.

(*iii*) Interfacing with the No. 4 ESS Network Management Programs.

(*iv*) Call register release.

(*v*) Interfacing with the No. 4 ESS Inward Wide Area Telecommunications Service Billing Program.

(*vi*) Interfacing with the No. 4 ESS Call Detail Recording Program for international calls.

The advantages of calling common call functions from the Report Dispenser instead of assigning that responsibility to the trunk handlers are as follows:

(*i*) minimization of errors—the functions can be called from a single module;

(*ii*) reduction in real time—such functions generally required the knowledge of both trunks, information which the Report Dispenser had available but would have to be regenerated in a trunk handler;

(*iii*) elimination of code—the functions can be called from a single module;

(*iv*) simplification of changes or additions to event-dependent functions—a significant advantage when adding features that are signaling-type independent.

2.5.3 Call failure handling

Call failure handling or final handling is the name given to the cleanup process for calls that fail to complete in a normal manner. Calls can fail due to machine error (hardware or software), customer error (misdialing, early abandon), or network conditions (congestion, network management controls).

There is a general class of events in the No. 4 ESS known as call

irregularities, which cause either a retrial attempt, or an abnormal termination of the call. An abnormal termination is called an ineffective attempt. Most ineffective attempts are because of an inability to complete the setup stage of a call. Some examples are as follows:

(*i*) ICT abandon in the setup stage.

(*ii*) Network congestion (all circuits busy, network management controls).

(*iii*) Failure on retrial attempts (glare, outpulsing errors, integrity check failures).

(*iv*) Office congestion (no CRS or service circuits, network blockage, overload controls in effect).

Some ineffective attempts occur in the post setup stage, such as loss of transmission (carrier failure).

Final handling clears ineffective attempts, allowing call processing resources (CR, trunks, service circuits) to be reused for new calls. Announcements and tones are also provided to help inform the customer of the situation.

There are numerous states that a call could be in when final handling is required. A call could be using many combinations of machine resources (i.e., CR timing lists, service circuits). Rather than determine the exact state of a call and idle only those resources and processes associated with that state, final handling checks for and idles all possible resources and processes on a call. In this way, calls can be cleared that have invalid states or invalid resources associated with valid states.

Final handling can be thought of as having two components, a call failure controller and a set of trunk clearing modules. The call failure controller holds a position in the architecture equivalent to the Report Dispenser, and like the Report Dispenser performs functions associated with common call related facilities (see Fig. 6). The trunk clearing modules are part of each trunk handler and provide a customer treatment based upon the trunk signaling type.

The call failure controller could have been made part of the Report Dispenser and final handling conditions treated via the same logical event-type interface that trunk handlers have with the Report Dispenser. However, the call failure controller already existed in the pre-restructure architecture and changing this interface would have had a major impact on the existing trunk handlers. A logical event-type interface like that of the Report Dispenser was provided in the call failure control module to accommodate the CCITT No. 5 and CCITT No. 6 trunk handlers, since they were new task programs to be developed during restructuring.

When a call requires final handling, the trunk handler interfaces with the final control module, which clears common facilities and

invokes the particular trunk clearing modules to idle remaining trunk-related facilities and to provide proper customer treatment for the call.

2.6 Evaluation

The interfaces and direction of communication between the trunk handlers, the task dispensers, and the Report Dispenser have become call processing programming standards. In some cases these standards produce a call flow which sacrifices real-time efficiency for the sake of uniformity. However, the sacrifice of real time is justified to maintain the integrity of the architecture. The analysis of call processing program errors and the changes required for program correction are a much simpler task because of easier problem isolation. The architecture makes the addition of new signaling types and design changes a more quantifiable job. The placement of new modules becomes readily apparent in the structure because the architecture directs the designer to a specific process of functional decomposition. The new signaling type is separated into ICT and OGT processes. Each process is then further decomposed into setup, post setup, and clearing functions, and new logical signals, if any, are identified. This process to a degree forces a consistent approach to the first level of task program modularity.

Since the call processing restructure was incremental, major portions of the existing code were not redesigned or rewritten. Existing task programs were not totally reorganized into distinct setup, post setup, and clearing modules. However, additional reorganization continues as new features are added to the call processing task programs. The implementation of the new CCITT No. 5 and CCITT No. 6 trunk handlers followed the call processing program standards completely. In addition, CCITT No. 6 was implemented with the use of EPLX.

As part of the 4E5 generic, the Mass Announcement System (MAS) feature was added to the No. 4 ESS. The MAS feature required a number of new types of calls to be processed by the No. 4 ESS and was a major software development undertaking in call processing. Using the structure of the new architecture as the basis for MAS feature decomposition and design, changes were made to add MAS to the call processing system. The feature addition was successful. Many of the new MAS calls executed correctly soon after introduction for testing in the system laboratory environment. At the same time, the old call types that the No. 4 ESS previously accommodated remained intact with no errors introduced as a result of the MAS feature addition.

The restructuring effort did not go without problems, the prime being increased real-time usage. After the architecture was solidified and much of the software developed, certain real-time critical parts were reviewed and optimized until real-time performance was judged to be within reason. As real-time improvements were made to the

EPLX, additional changes were made to certain areas of call processing to further improve real-time performance.

III. MAINTENANCE SOFTWARE RESTRUCTURING

Peripheral maintenance software for the No. 4 ESS also has been selectively restructured to minimize the cost of developing such software and provide the ability to continue to add new hardware features. The restructured peripheral fault recovery system incorporates operating system concepts, top-down hierarchically designed control structures, and use of a formal development methodology. This section gives a brief overview of the pre-restructured maintenance system. We also give error recovery and system recovery concepts, the motivation behind restructuring a selective part of the maintenance system, and finally a description of the restructured system, and an evaluation of the benefits of the new system.

3.1 Maintenance system overview

The stringent reliability and maintainability requirements of the No. 4 ESS affect both the hardware and software design of the system. In the software, we have developed a large program package to provide maintenance functions.[3] This maintenance software package consists of four functional areas that play an essential role in providing the maintenance capabilities of the No. 4 ESS: (*i*) fault recovery; (*ii*) diagnostics; (*iii*) system reinitialization and recovery; and (*iv*) system integrity and audits. Fault recovery is concerned with the system recovery from hardware faults. Diagnostics aid the craftperson in the identification of faults and repair of a faulty unit. System reinitialization is concerned with the overall coordination of system recovery from multiple or severe hardware and software malfunctions. The system integrity area is concerned with the detection of and recovery from memory mutilation. These software areas are designed based on specific error recovery and system recovery concepts. We give these concepts in later sections for background information.

Various types of redundancy (e.g., duplex, $n + 1$ duplication, $n + 2$ duplication, and load sharing) are used in the different subsystems to meet hardware reliability requirements. Each subsystem uses a number of error-detection techniques such as parity, matching, order acknowledgment, and self checking. These hardware characteristics place specific requirements on the maintenance software, particularly the fault recovery area, which is tightly coupled to hardware design.

3.1.1 Error recovery concepts

The maintenance software is built around several levels of execution based upon both maintenance software and hardware error detection triggers. Table I shows the maintenance program execution levels in

Table I—Maintenance program execution levels

Layer	Level	Function
System Recovery	Phase 4 Phase 3 Phase 2 Phase 1	System initialization
Maintenance Hardware Interrupts	A B C D E	Processor fault recovery
	F	— Peripheral recovery
	G	— Utility and timing
	K	— Segment timing validation
Maintanenace Software	Interject BLM	Fault recovery
Manual requests Audits Diagnostics	Base	Low-priority task

the No. 4 ESS. Only those execution levels which are applicable to peripheral fault recovery are discussed. The remaining levels of execution are presented in Ref. 4 on the 1A Processor.

Base level is the lowest and the normal level of system execution. All the call processing work described earlier, as well as audits and diagnostics, are normally executed at this level. Base level maintenance (BLM) is the next level and is triggered by defensive checks provided in software or firmware. Interject level is the next higher level and is guaranteed to be served by the 1A Processor every 10 ms. F-level interrupts report peripheral errors and are of two types: peripheral unit failure (PUF) and autonomous peripheral unit failure (APUF). The PUF interrupt is generated by the 1A Processor when a peripheral frame fails to acknowlege, or incorrectly acknowledges a directed order. The APUF interrupt is generated autonomously by a peripheral unit failure. The 1A Processor scans for APUFs every 11.2 μs. Base level maintenance, interjects, and both types of F-level interrupts invoke peripheral fault recovery. Fault recovery actions can also be stimulated by manual requests, such as input messages or power control switch requests.

3.1.2 System recovery concepts

When fault recovery succeeds in reconfiguring the system so the faulty unit is not in service, repair activity commences. However, in cases when complex or multiple faults prevent fault recovery from configuring an acceptable working system, system recovery actions are taken. Phase recovery is the highest level system recovery action and can be initiated either manually or by software. Phase recovery can

escalate through four phases, where phase 1 is the least severe and phase 4 is the most severe. Phase 4 can only be requested manually.

There are two types of Phase 1s. The first type executes a specified set of audits that correct data mutilation. The second type is a directed phase 1 and is initiated by a fault recovery action on a peripheral unit which results in a loss of service provided by the unit. The directed phase 1 initializes software structures associated with the faulty unit. A phase 2 initializes additional software structures and also performs a unit access test on the peripheral hardware when it is initiated by F-levels. Phase 3 is the highest level phase that is automatically requested. It performs additional software structure initialization and additional tests on the hardware. A phase 4 performs a total system initialization and can only be requested manually.

3.2 Motivations for a modern structure

3.2.1 Drawbacks of original implementation

Since the initial generic release (termed 4E0), each new generic has included new features, hardware cost reductions, and enhancements. Each generic must continue to meet the original design objectives of the system for capacity and reliability and, at the same time, provide new services, and take advantage of the rapidly changing technology through hardware cost reductions. By the end of the second generic (4E1), the continuing demand for new hardware features and cost-reduced hardware was evident. We, therefore, determined how the development cost could be reduced for maintenance software. Of the four maintenance software areas described earlier, we found the fault recovery area to be affected most by new hardware feature development since efficient changes or additions could not be made.

The principal reason the pre-4E2 fault recovery software exhibited this lack of flexibility was that it was functionally partitioned with a decentralized control structure. Figure 8 shows the functional partitions used in the pre-4E2 fault recovery software. They include:

(*i*) Peripheral Configuration Program,
(*ii*) Craft-Machine Interface Program,
(*iii*) Hardware Phase Recovery,
(*iv*) Error Analysis Program,
(*v*) Per unit fault recovery programs.

Each functional area contained control and unit-dependent code for many units embedded in the same programs. Each time a new unit was added, the functional area was modified, resulting in increased complexity, and requiring the entire area to be retested. Each one had decentralized control and had to provide for the following common requirements:

(*i*) Multilevel execution,
(*ii*) Unit dependent interfaces,

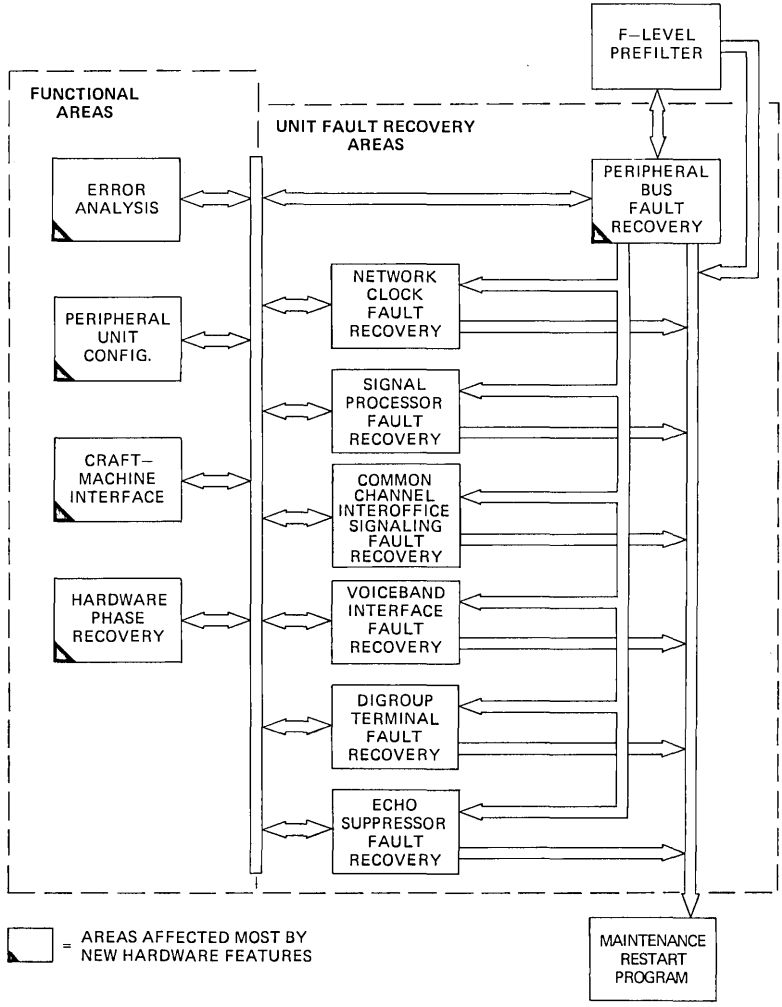


Fig. 8—Peripheral fault recovery structure (4E2 generic).

(*iii*) Hardware and software coordination,

(*iv*) Reentrant software.

Fault recovery software is required to execute on all system levels of execution, i.e., base level, BLM, interject, F-level interrupt, and phase level. In general, other maintenance software executes only on one level. In addition, fault recovery software is required to interface to all types of peripheral hardware. Hardware coordination is required because of the highly interconnected hardware. In particular, recovery from a problem in one unit generally affects other units. Software coordination is required to prevent software interaction due to time-

shared execution on base level. Reentrant software is required due to the multilevel execution of fault recovery software. As an example, the same fault recovery software may be started and successively restarted by escalating recovery actions. These requirements resulted in excessive interfaces and interaction between the different functional areas. Consequently, much fault recovery code was duplicated in an effort to reduce the number of interfaces. However, duplication made the job of maintaining the software much more difficult.

Each unit fault recovery program was responsible for many functions common to fault recovery of several units. The functions were being performed by several different programs in numerous ways. For example, each unit fault recovery program provided interfaces to each functional area, provided software to collect common recovery data, provided software to output recovery messages, etc.

Since most fault recovery software executes on interrupt level, it was designed with emphasis on real-time efficiency to minimize the interruption of base level due to a faulty unit. Techniques such as "tricky code" and private interfaces, as examples, were used for real-time efficiency. This also contributed to a structure that was difficult to change.

3.2.2 Development of improved structure

In response to these shortcomings in the pre-restructured fault recovery structure, an improved fault recovery structure was developed to incorporate the following: (*i*) a peripheral maintenance operating system; (*ii*) new hierarchically designed fault recovery control structures; (*iii*) a higher-level language; and (*iv*) a more formal development methodology.

The operating system would remove some complexity from the software by handling multilevel execution, memory allocation, and software coordination, and provide a truly standard interface between functional areas of fault recovery software.

The hierarchically designed control structure would provide complete separation between control and unit-dependent code. This would remove much of the unnecessary complexity in the control areas and limit the testing mainly to the new feature software being added. A hierarchical structure would lend itself more easily to changes and additions. It would improve readability and maintainability of the product.

A high-level language would improve programming productivity, readability, and maintainability. Programming productivity would be improved by allowing the programmer to concentrate on programming the function and not on initializing and saving registers, implementing loops, etc. Removing this level of detail from the source code would also improve the readability and maintainability of a program.

A formal development methodology would provide uniform and up-to-date documentation. The more rigorous steps in a methodology that insist on requirement reviews, design reviews, code walk-throughs, and test plan reviews help ensure that more software bugs are found early in the development. Other benefits of this formal development methodology, which uses development teams, are better project visibility and a larger group of people with knowledge of the software.

The development cost of the operating system and new control structures could be spread over several generics with little additional development cost beyond that required to add new units. Once the operating system and control structure were in place, the development costs for a new hardware-related feature would be reduced. In addition, program maintenance cost would be reduced.

Each of the above techniques, to some degree, has the drawback of less real-time efficiency and greater program size. The advantages stated above were judged to outweigh these considerations. It is also common practice when using a structured design approach to optimize after the design is working. Time should be scheduled for optimization when it can be determined which areas require real-time and program-size optimization. Note that optimization is generally easier in a structured design that is written in a high-level language. In many cases, large improvements in real-time and program store usage can be accomplished by small changes in a structure and/or compiler. Also, the increased program size is partially offset by reduced temporary memory requirements. This reduction can be attributed to more efficient use of temporary memory by the new operating system.

The fault recovery software, thus, evolved to a set of centralized control structures executing under a maintenance operating system. These control structures are designed with complete separation between control and unit-dependent code. Both maintenance control and unit-dependent software are written in EPLX. To add units to this new system, unit-dependent modules are added to each control structure as illustrated in Fig. 9 by the dashed blocks. In general, no modification is needed to the control structures. This results in testing the new unit software and little regression testing. In the pre-restructured fault recovery system each functional area required modifications to add the new control code and unit code. In Fig. 8 the blocks with the blacked-in corners are the functional areas that required extensive changes and additions. Each functional area required testing of the new unit fault recovery software and extensive regression testing of existing unit fault recovery software.

The new fault recovery system was planned to be evolved over several generics and to operate in parallel with the pre-restructure fault recovery system. The pre-restructure fault recovery system con-

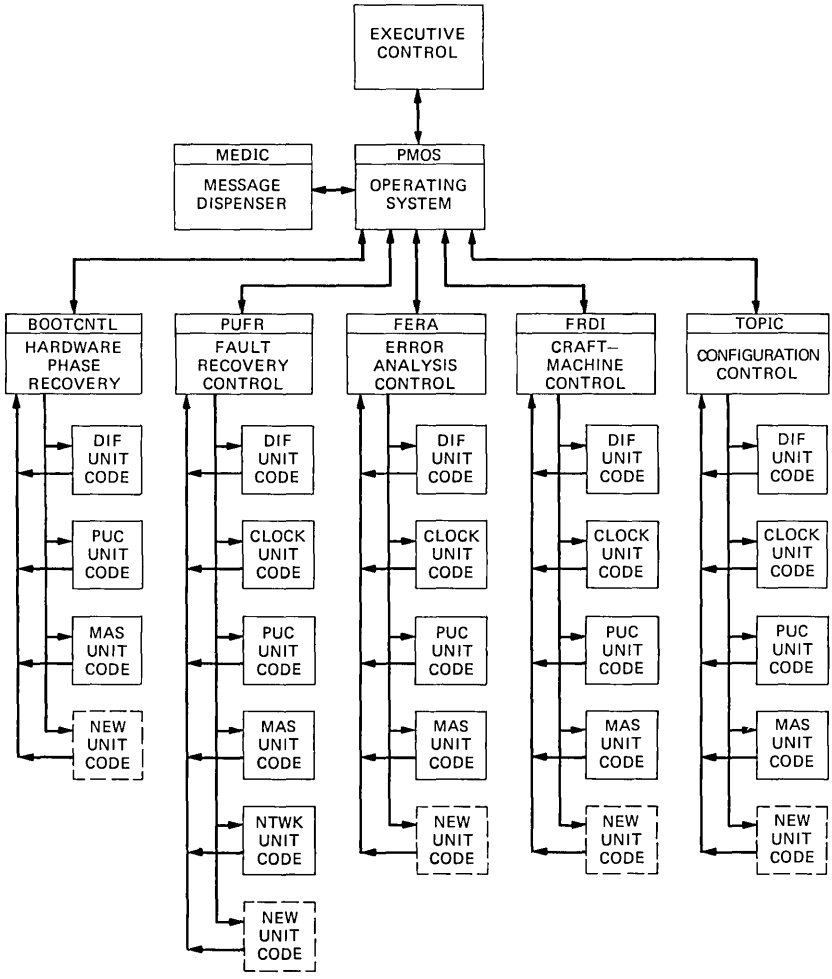


Fig. 9—Peripheral fault recovery structure (4E5 generic).

tinues to handle the units it was designed to accommodate. New units are being implemented under the new fault recovery system in the EPLX language.

3.3 Characterization of new fault recovery control architecture

The development of the new fault recovery control architecture was a multigeneric development. This new architecture was developed as a parallel system without disturbing the existing fault recovery software system, which supports the existing peripheral hardware architecture. The initial introduction was in the 4E3 generic with the development of the Peripheral Maintenance Operating System (PMOS)

and the Peripheral Unit Fault Recovery (PUFR) control structure. The first units supported by this system were the network frames (Time Slot Interchange and Time Multiplexed Switch). In the 4E5 generic, five new control structures were added, plus unit-dependent code for four new hardware frames. The five control structures were: (*i*) Toll Peripheral Configuration (TOPIC); (*ii*) Frame Request and Diagnostic Interface (FRDI); (*iii*) Failure Error Analysis (FERA); (*iv*) Message Dispenser and Coordinator (MEDIC); and (*v*) Bootstrap Control (BOOTCNTL) Program.

Each of the control structures, with the exception of MEDIC, was recommended as part of the original plan. Message Dispenser and Coordinator is a control structure which resulted from the introduction of intelligent (micropressor based) peripherals into the No. 4 ESS. These new peripherals execute macro-level orders which return multiword responses on a deferred basis after control is released. This required a new structure to control sending, receiving, and dispensing responses from these units. All four of the new hardware frames developed for 4E5 were of this type.

The resulting fault recovery software structure after the 4E5 generic is shown in Fig. 9. Each of the control structures was designed to meet the following objectives:

(*i*) Remove complex system dependencies by making use of the PMOS.

(*ii*) Make use of a more formal development methodology.

(*iii*) Use EPLX.

(*iv*) Provide complete separation between control and unit-dependent code.

(*v*) Provide standard unit-dependent interfaces.

(*vi*) Remove limitations (structure sizes, number of units, etc.) which exist in the present fault recovery software system.

(*vii*) Provide new capabilities.

Section 3.3.1 briefly describes each functional area of the new control architecture.

3.3.1 Peripheral maintenance operating system

The PMOS is the heart of the new fault recovery control architecture. This operating system centralizes peripheral maintenance control and coordination while reducing the complexity of system interaction. The operating system provides a standardized interface between PMOS tasks and the remainder of the system software. A PMOS task is a software process or function defined to the operating system. This interface allows an operating system task to be requested with several options specifying levels of execution, request mode, and priority. For example, simultaneous tasks can be scheduled on the same level or different levels, requested in a schedule and hold mode, parallel sched-

ule, or run-immediate mode. The operating system provides for task execution on base level, BLM, interject level, F-level, and phase level. Schedule and hold mode allows a task to schedule other tasks and be suspended until the other task is completed. Parallel schedule allows a task to schedule several other tasks and be suspended until all are completed. Run-immediate mode allows a task to request other tasks to be executed immediately. The main features of the operating system are: (*i*) task coordination; (*ii*) multilevel execution; (*iii*) administration of segment breaks; and (*iv*) reentry.

Peripheral fault recovery code for several units tends to be tightly coupled due to highly interconnected hardware units. The operating system removes from a task many of the concerns of task interaction by providing several task coordination functions. The operating system exercises blocking rules as defined in a central blocking table. Blocking prevents time-shared execution of specific tasks which otherwise would interact. Control of abort conditions and the execution of abort procedures are also provided. Control of execution and the determination of associated priorities are also included in a task coordination function.

Multilevel execution is a characteristic which in the past required numerous redundancies in many fault recovery programs. For example, each fault recovery program was required to check for the execution level and perform the necessary function to segment on that level. The operating system consolidates the necessary checks and functions to execute on different levels in one place. In general, tasks need not know what execution level they are on.

Segment breaks* required by base level processing add complexity and substantial development cost without a unified control architecture. Peripheral Maintenance Operating System provides segmentation routines for the new control structures. These routines preserve task memory when segment breaks are taken and ignore segment breaks on interrupt, interject, and phase level. The operating system also provides routines for timing breaks. Timing breaks at any execution level releases the operating system for execution of other tasks until the time specified at the break has expired. The task environment is preserved on segment breaks or timing breaks and reestablished upon return to the task.

Reentry is a condition that causes numerous problems for multilevel maintenance software. This problem arises, for example, when a multilevel program is interrupted on base level and the same program is entered on the interrupt. This can result in variables, initialized on base level, being overwritten on the interrupt level. This problem has

* Segment break is a convention in No. 4 ESS whereby all base level processing programs are required to return control to the Executive Control program every 3 ms.

forced fault recovery software to be exceedingly defensive during execution, adding system integrity checks and numerous other controls to avoid problems. The operating system resolves each case of reentry to the interrupted program. The integrity of each task is maintained either by aborting a task or allocating a different memory block to the task.

3.3.2 Message dispenser and coordinator

The MEDIC is a control structure developed to satisfy new requirements introduced with microprocessor-based frames in the peripheral system of the No. 4 ESS. Prior to 4E5, all peripheral frames on the peripheral unit bus returned responses to orders in the peripheral unit bus reply window. This window is 32 1A Processor cycles or 22.4 μs in duration. With the introduction of microprocessor-based frames, their macro-level orders required much longer times to complete because of the higher-level function being performed. These frames were designed to return an initial response in the reply window, indicating the order was accepted. A "task complete" response was returned when the macro work was completed within the frame.

Message Dispenser and Coordinator was developed as a control structure, having special interaction with the operating system. The basic functions of MEDIC are to (*i*) coordinate sending macro orders to microprocessor-based frames; (*ii*) poll these frames for responses on a deferred basis; and (*iii*) dispense those results to the appropriate client. The message dispenser, in conjunction with the operating system, provides primitives (low-level function calls) which allow a PMOS task to be suspended while waiting for a macro response. The task is automatically reactivated when the macro response is received. MEDIC provides a macro timeout notification. If a response is not received in a predefined maximum allowed time, the task is notified. The message dispenser also provides appropriate handling of unsolicited frame reports and autonomously generated reports. The unsolicited and autonomously generated reports are processed on BLM. The fault recovery program resolves the cause of the report and takes the appropriate recovery action to clear the problem.

3.3.3 Peripheral unit fault recovery

The PUFR control structure was the first control structure developed. It was developed in the 4E3 generic and supported the cost-reduced TSI frames. The Peripheral Unit Fault Recovery is a common control structure that handles all levels of peripheral error recovery (BLM, interject, and F-level). It consolidates common error-recovery functions under one control program by providing the following common functions: (*i*) initialization of data structures; (*ii*) collection of critical data required to isolate the source of a fault; (*iii*) an interface to unit-

dependent tasks to isolate the fault; (*iv*) an interface to error analysis programs to acknowledge the recovery actions; (*v*) execution of the necessary actions to recover the system; (*vi*) scheduling of any deferred maintenance actions, e.g., diagnostic, audits, etc.; and (*vii*) printing of reports containing critical data and the recovery action taken at the time of the fault.

The Peripheral Unit Fault Recovery controls the execution of fault recovery by calling both common routines and special unit dependent procedures. It satisfies the requirements of complete separation of control and unit-dependent software by providing standard interfaces to unit-dependent procedures. It calls the appropriate unit procedure by indexing a table based on unit identity. In addition to consolidating the common functions, PUFR also provides several enhancements. Some of these enhancements are: (*i*) multiple isolation attempts; (*ii*) multiple unit interface to error analysis; and (*iii*) enhanced report messages. Multiple isolation attempts allow a unit isolate program to request an isolation attempt on a different unit, for the same interrupt, when the problem cannot be resolved to the original unit. The subsequent isolation attempt is usually on a connecting frame. Multiple unit interface to error analysis allows the unit isolation program to pass a list of suspect units to error analysis when the problem cannot be resolved to a single unit. Enhanced report messages provide the craft with additional information concerning the source of the interrupt and the corrective action taken.

3.3.4 Failure error analysis

The FERA program provides the centralized control structure for carrying out the fault recovery error analysis function. The main role of error analysis in the No. 4 ESS is listed below:

(*i*) Complement fault recovery by adding the element of interrupt history,

(ii) Resolve intermittent and transient hardware faults,

(*iii*) Resolve faults in interconnected hardware,

(*iv*) Isolate persistent or intermittent system troubles in highly interconnected hardware subsystems,

(*v*) Record and analyze error history information,

(*vi*) Provide graceful degradation by removing units, which causes the minimum service effect, to correct a system problem,

(*vii*) Monitor deferred maintenance activities to guard against removing the redundant part of an intermittent failing piece of equipment.

Failure Error Analysis provides these functions by determining recovery actions with strategy tables. Strategy tables are a collection of decision schemes which make different decisions on successive occurrences of an error. A strategy table is selected by fault recovery

based on the type of fault occurring. The Analysis can acknowledge and accept the action recommended by PUFR or recommend an alternate action. The decision schemes and the selection of a strategy table use several factors to reach a decision:

(*i*) environment of the configurable portion of the system (simplex, duplex),

(*ii*) number of times the fault has occurred,

(*iii*) type of fault (unique, nonunique),

(*iv*) characterisic of the fault (transient, hard failure, illegal system action).

The Analysis also provides control for alternate recovery strategies. This control provides better analysis functions to be performed. Another new feature is a parallel analysis capability, which allows a strategy table and analysis function to execute in parallel. If the analysis function reaches a conclusion, it can override the action recommended by the strategy table. These new strategies allow FERA to resolve intermittent faults, transient faults in interconnected hardware, and persistent troubles more efficiently than the existing error analysis program. These enhancements were provided in addition to meeting the common objectives of all the new control structures. Secondary functions provided by FERA are (*i*) monitoring manual configuration requests; (*ii*) monitoring deferred maintenance actions (diagnostics, routine exercises); and (*iii*) manual input/output for control and display of FERA functions and data.

3.3.5 Craft-machine control program

The craft-machine interface functions are provided by the FRDI control structure. It provides the basic interface for manual configuration requests of the No. 4 ESS peripherals from either the TTY or Power Control Switch (PCS) located on the frame. Requests from the TTY may be for removal, restoral with diagnostic, restoral without diagnostics (unconditional), or for a switch of an active unit. Requests from the PCS may be for removal of a unit or restoral with a diagnostic. When any manual configuration requests are initiated, FRDI validates the request, interfaces with the TOPIC program to perform the configuration, interfaces with the FERA program to monitor the request, and prints the appropriate message to indicate whether the action was completed or denied. In addition to printing a message, if the request was initiated from a PCS, lights at the frame are lighted or extinguished to acknowledge the request.

The Frame Request and Diagnostic Interface is also the primary interface to the Diagnostic Control program for peripheral configuration before and after diagnostic requests. All diagnostic requests, independent of the source, are validated by FRDI. After the request is validated, the appropriate configuration function is requested. After

the diagnostic, FRDI controls the disposition of the unit by either restoring it or leaving it out of service. This decision depends upon a variety of conditions, such as the termination condition of the diagnostic, the results of the diagnostic, the type of request, and the state of the PCS.

3.3.6 Configuration control

The configuration control in the new peripheral fault recovery control architecture is provided by the TOPIC program. The Toll Peripheral Configuration program is responsible for establishing the configuration of the new peripherals introduced in the 4E5 generic. It is also responsible for the configuration of the Network Clock (NCLK) and System Clock (SYSCLK). These latter units existed in the initial release of the No. 4 ESS generic, 4E0. However, with the addition of the Network Clock Synchronization Unit (NCSU) in 4E5,[5] a major portion of the configuration software had to be modified and was moved into the new architecture.

Beyond the primary function of accepting requests from all sources and directing configuration requests to the specific unit-dependent program, TOPIC will determine if there are any connecting unit considerations, for example, clock or voice data path dependencies. If there are, TOPIC will take appropriate action on the connecting units. It also attempts to leave the resultant peripheral system configuration in a state that minimizes service degrading conditions. This is also based on connecting unit status.

3.3.7 Bootstrap control

Bootstrap is a function executed during phases of recovery which include hardware configuration. The bootstrap function for the new units in the 4E5 generic is controlled by the BOOTCNTL program. The function of bootstrap is to initialize the hardware and execute access tests. The degree of initialization and access testing varies depending on the phase of recovery (phases 1 to 4). With the introduction of microprocessor based units in 4E5, a large portion of the initialization of these frames is performed by firmware resident within the frame. During this initialization process, the 1A Processor is free to perform other functions. Bootstrap Control, making use of features provided by the operating system, is free to start the bootstrapping of other units. This technique is referred to as parallel bootstrap. This process results in less total time required to bootstrap several frames than would be required if the function were executed in a serial fashion.

Bootstrap Control also provides output containing the results of access tests performed during the phase. This information is useful to

the craft in understanding the final configuration after a phase, and in troubleshooting the peripherals removed from service during a phase.

3.4 A fault recovery example

This section presents a simple example of recovery from a hardware fault by the restructured fault recovery system. This example is provided to give a clear understanding of the function of each control structure. Figure 10 illustrates the actions taken in this example. A single hard (nonintermittent) fault in a duplicated unit is assumed for this example. The FRDI and BOOTCNTL structures are not involved in this example since they primarily execute on base and phase level, respectively.

A hardware error triggers an F-level interrupt and results in a PUFR F-level task being scheduled in PMOS. Peripheral Unit Fault Recovery performs initialization of internal data structures and collects data at the time of the interrupt which reflects the state of the system and the interrupting unit. It then schedules (schedule and hold mode) the unit's fault recovery task, passing the data it has gathered as input. Peripheral Unit Fault Recovery is suspended until the unit fault recovery task completes.

The unit fault recovery task will attempt to isolate the source of the interrupt to a configurable piece of hardware (half of a duplicated unit, etc.). The unit fault recovery task will analyze the input data, perform access tests, and reconfigure the unit and retry peripheral orders to isolate the source of the error. The suspect unit half, fault class (hard fault, software fault, intermittent fault, etc.), resolution class (resolved, unresolved), and recommended actions are returned to PUFR in a data block passed as input. Peripheral Unit Fault Recovery is reactivated when the unit fault recovery task is completed.

It then schedules (schedule and hold mode) the FERA task passing the suspect unit, fault class, and resolution class as inputs. The Analysis then determines if this is the first interrupt for this unit by consulting history data files. A new history data file is allocated, if it is the first interrupt. The Analysis updates a history data file, if a previous interrupt has been recorded for the suspect unit. It then selects a primary strategy based on interrupting unit, number of interrupts, fault class, resolution class, and configuration of the unit at the time of the error (simplex or duplex). The primary strategy acknowledges the action recommended by the unit fault recovery task or specifies an alternate action. The primary strategy is not used if the number of interrupts which has occurred on this unit is greater than the designed limits of a strategy table. The Analysis control selects a secondary strategy if this occurs. Otherwise, the secondary strategy is not used if a recovery action is specified by the primary strategy. The Analysis

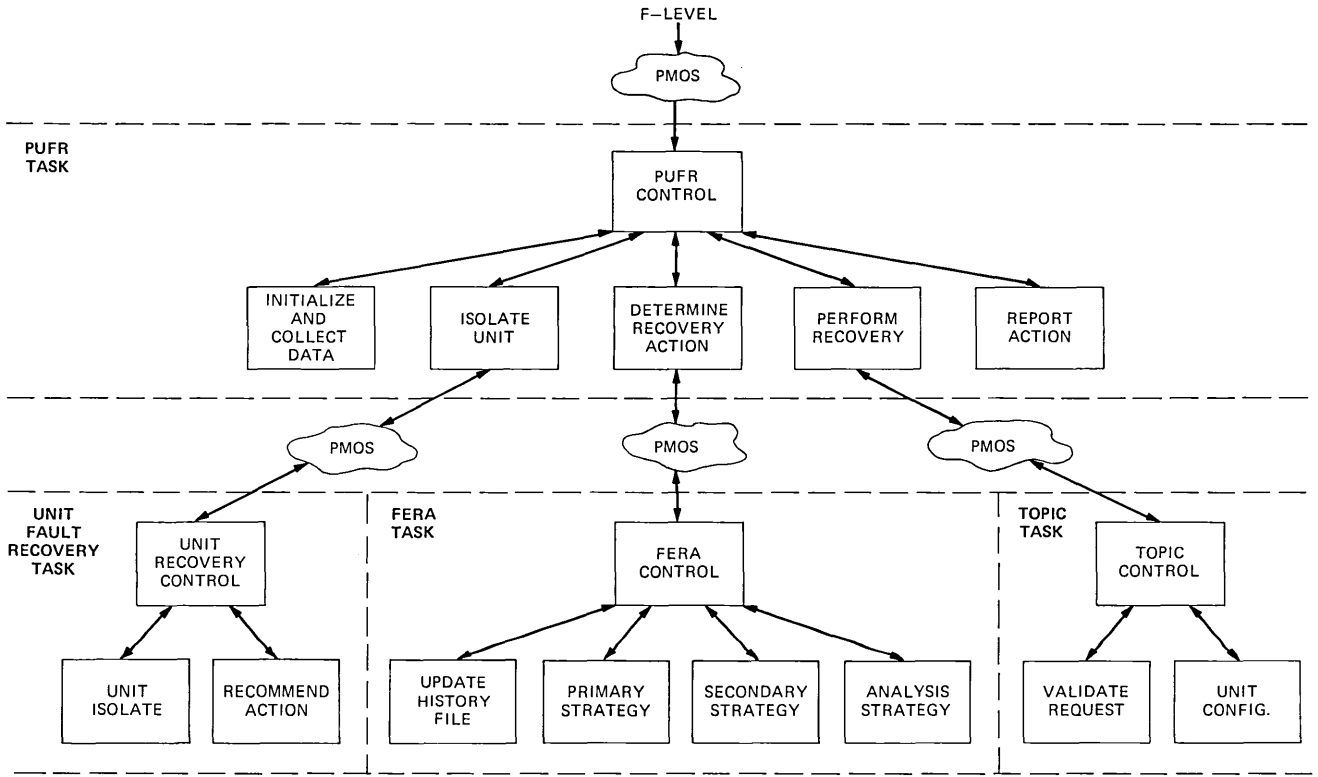


Fig. 10—Example fault recovery actions.

control always executes the analysis strategy independent of the action specified in the primary or secondary strategy. The analysis strategy views the interrupt in terms of a potential multi-unit hardware interconnection fault. The analysis strategy can overide the action specified by the primary or secondary strategy if an interconnection fault is suspected. The Analysis returns the recommended recovery action to PUFR in a data block passed as input.

The Peripheral Unit Fault Recovery is again reactivated when FERA completes. The recovery actions specified by FERA are then performed. The recovery actions may be an immediate action (configuration, etc.) or a deferred action (diagnostic, audits, etc.). Deferred actions are scheduled for base level execution. Immediate actions are scheduled for F-level execution. In the case of an immediate configuration action, PUFR schedules a TOPIC task in a schedule and hold mode. Toll Peripheral Configuration validates the configuration request and interfaces to the specified unit software for the function requested (remove, restore, switch, etc.). PUFR is reactivated after the TOPIC task is completed.

At this point the recovery is completed. The remaining function of the Peripheral Unit Fault Recovery is to format and output the data collected at the time of the interrupt and the recovery actions taken. It then returns to PMOS, completing the F-level processing.

The Peripheral Maintenance Operations System returns to base level processing after it determines that no other F-level tasks are scheduled. The deferred actions scheduled for base level are then executed.

3.5 Evaluation

With the release of the 4E5 generic, the multigeneric plan to develop a centralized peripheral fault-recovery-control architecture and a maintenance operating system is complete. This new control structure provides the fault recovery capability for four new peripheral unit types (DIF, PUC, MAS, NCSU).[5-7] Four other peripheral unit types (TSI, TMS, NCLK, SYSCLK) are partially supported under this system.

The development cost of the new fault recovery control architecture and the unit dependent code for the units listed above has been slightly larger than the original estimates. This difference can be partially attributed to the introduction of microprocessor technology with these units. It can also be said that introducing this new technology in the pre-restructured maintenance system would have resulted in even larger software development costs for these units.

The new fault-recovery-control architecture provides a well-documented, flexible-control architecture with well-defined interfaces to unit-dependent code. With this control architecture in place, new

features can be developed with an estimated 30 to 50 percent savings in fault recovery software effort. A portion of this savings can be attributed to the development methodology and the use of a high-level language. It is difficult to estimate the savings contributed by either factor. Also, it is not clear that the total benefit of either factor can be attained without the other also being present.

There are also many side benefits in addition to a decrease in development costs. Most of these benefits stem from the use of a modern development methodology. Improved, up-to-date documentation is one benefit already mentioned. Others are (*i*) better project visibility through the use of development teams and walkthroughs; (*ii*) a larger base of people with knowledge of specific software modules through the use of development teams; and (*iii*) more software bugs found early in the development prior to laboratory testing and field release.

Disadvantages of the new fault recovery control architecture and structure design methodology are increased program size and real-time usage. Real-time usage in error recovery, even though critical, does not significantly affect the system call handling capability as in call processing programs. These disadvantages were anticipated, but it was unclear what increase could be expected. Initial data indicate that a specific function like error recovery, which is real-time critical, has the following distribution of real-time usage:

Operating system—10 percent,
Control structures—2 percent,
Macro waits—35 percent,
Unit code—53 percent.

Some portion of the operating system, control structures, and unit code time can be attributed to the use of a high-level language. However, without recoding specific procedures it is difficult to determine what percentage is due to the language. Experiments have been performed comparing EPLX with the previously used language (EPL). In general, EPLX used more real time and program store than EPL. However, with some optimization the EPLX program was nearly as efficient as the EPL program. This indicates that there is little inherent inefficiency in the language. With proper knowledge of the language, programs can be optimized for both real time and program store usage.

IV. SUMMARY

This paper has described two specific examples which show how the No. 4 ESS has evolved through software restructure to better accommodate the addition of new hardware and software features. Call Processing and Fault Recovery software underwent varying degrees of incremental restructure. These software areas were considered for

restructure because many new features were to be added to the No. 4 ESS which directly affected Call Processing and Fault Recovery. The restructuring efforts focused on improving the deficiencies of the pre-restructure software and made use of modern development methodologies and a high-level programming language to accomplish the objectives. The resultant architectures are heirarchical, much more modular, and more easily modified and maintained. We acknowledge the effort of those designers too numerous to mention, who contributed to the successful Call Processing and Fault Recovery restructuring effort.

REFERENCES

1. E. A. Davis and P. K. Giloth, "No. 4 ESS Performance Objectives and Service Experience," B.S.T.J., this issue.
2. T. J. Cieslak et al., "Software Organization and Basic Call Handling," B.S.T.J., *56*, No. 7 (September 1977), pp. 1113–1138.
3. M. N. Meyers, W. A. Routt, and K. W. Yoder, "No. 4 ESS Maintenance Software," B.S.T.J., *56*, No. 7 (September 1977), pp. 1139–67.
4. "1A Processor," B.S.T.J., *56*, No. 2 (February 1977), pp. 119–327.
5. R. Metz and D. F. Winchell, "No. 4 ESS Network Clock Synchronization," B.S.T.J., this issue.
6. T. W. Anderson et al., "No. 4 ESS Mass Announcement Subsystem," B.S.T.J., this issue.
7. K. M. Hoppner et al., "No. 4 ESS Digital Interface," B.S.T.J., this issue.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 60, No. 6, July–August 1981
Printed in U.S.A.

No. 4 ESS:

Performance Objectives and Service Experience

By E. A. DAVIS and P. K. GILOTH

(Manuscript received July 16, 1980)

The development plan of the No. 4 ESS included provisions for measuring the effectiveness of the design, operation, maintenance, and administration of the total system. This paper reviews system performance from 1976 to 1980, describes principal factors affecting system performance, and presents the service experience measured for the No. 4 ESS. Steady improvement has been measured in the number of service-affecting incidents experienced per office each month. This improvement is also reflected in the rate of cutoff and denied calls, as well as in system "no call processing" time. We discuss some of the factors influencing this performance record, e.g., a sound initial design, reliable hardware, effective maintenance and repair tools, continuing analysis and resolution of causes of service-affecting incidents, and continuing development of new features for performance improvement.

I. INTRODUCTION

The No. 4 ESS is a digital time-division toll and tandem switching system first placed in service in Chicago. It was described in the *Bell System Technical Journal* in 1976.[1] Since then, 51 offices with over 1,000,000 terminations have been put into service. The deployment progress is shown in Fig. 1. The average size of the No. 4 ESS is 22,000 terminations with current office sizes ranging from 6,000 to over 60,000 terminations. Detailed statistics demonstrate that the No. 4 ESS provides high-quality service to its customers and that its performance continues to improve as the system matures despite office growth, new generic programs, and evolving hardware.

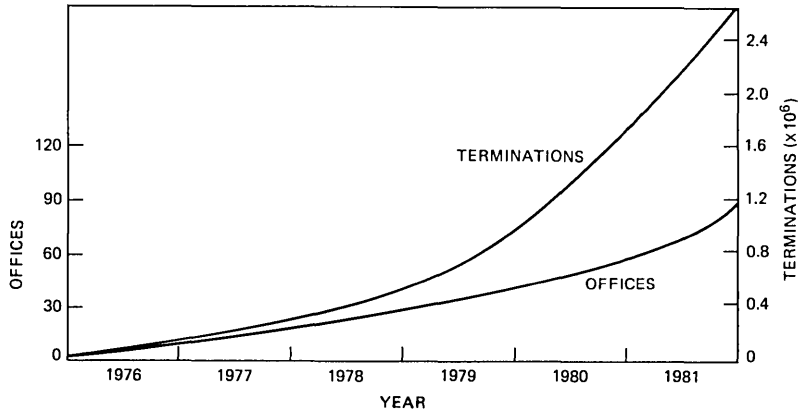


Fig. 1—No. 4 ESS deployment.

Substantial effort has been applied to developing methods and procedures for evaluating the performance of hardware and software in the No. 4 ESS. Data collection on performance parameters was built into the initial design so that performance data from many No. 4 ESS systems could be obtained easily and accurately. New performance criteria have been developed to measure the effect on the customer and to provide data for maintaining the hardware and software.

A typical No. 4 ESS has intertoll and toll connecting trunks to about 200 other switching entities. Therefore, because of its size and position in the network, its continuous availability for service is needed since any malfunction can affect communication in many areas of the country. All No. 4 ESS machines are staffed 24 hours a day, 7 days a week, and all service-affecting incidents are reported and analyzed. Special attention is given to correct the causes of service-affecting incidents.

This paper describes some of the major system objectives, specific reliability and maintainability objectives, operational factors affecting performance, service experience, and methods used to manage performance. References 1 through 8 provide additional information on system performance.

II. SYSTEM OBJECTIVES

The traditional measure of telephone switching system reliability and performance is the amount of "no call processing" time in 40 system years. This measure is a useful design objective, but it does not include all of the effects of complete and partial system failures which can lead to unsatisfactory performance from a customer viewpoint.

The primary objective is to minimize the impact on the customer of

all types of system failures. Consequently, cutoff calls and denied calls are among the most important performance indicators measured in the No. 4 ESS. Many other performance indicators are also measured to determine the effectiveness of maintenance and operation so that procedures and design problems can be corrected promptly.

As an example, the derivation of the cutoff call objective for a toll call is shown in Fig. 2. Calls are assumed to pass through two local offices, two toll offices, and interconnecting transmission facilities. As shown, the overall call cutoff objective is less than or equal to 15 calls per 10,000, with an allocation to each switching entity of less than or equal to 1.25 calls per 10,000.

Special performance criteria were set for the cutover of the first No. 4 ESS in Chicago in 1976 (referred to as Chicago 7). They were expressed both as objectives and concern thresholds.[1] Table I lists the objectives.

Performance objectives have also been set for other performance indicators where supporting information is available. However, some performance measures are new, and the present, self-imposed, objectives are based on data obtained from typical No. 4 ESS offices and were not part of the original design objectives. The new objectives are described later in this paper.

The design of reliable telephone switching systems involves built-in tools to measure performance, as well as reliable hardware, software, and equipment configurations. Objectives must be set that are stringent, yet attainable at a reasonable cost. Objectives for the No. 4 ESS performance are based on a reliability model and field data from the existing network. Advances in technology and the expectations of the public are also considered in setting objectives.

The ultimate performance of telephone switching systems depends on design, as well as installation, operation, and maintenance. Conse-

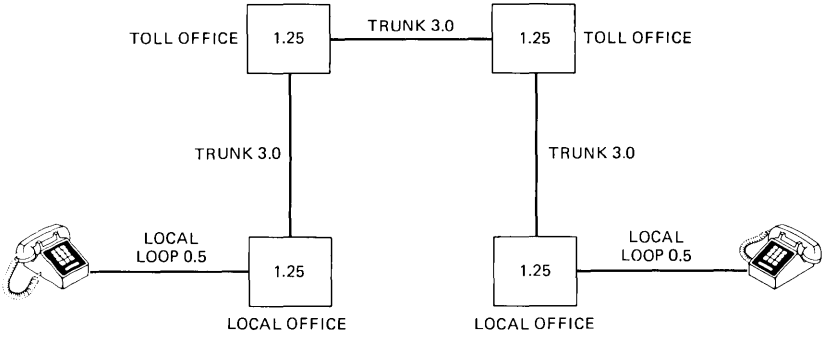


Fig. 2—Allocation of cutoff calls objective in calls per 10,000. The total cutoff call objective is less than or equal to 15 calls per 10,000.

Table I—Chicago 7 cutover objectives

Description	Objectives
Ineffective Attempts	<1.25 percent
Plug-in Replacements	<2 per day
Interrupts	<50 per day
Phases (2 or higher)	<1/2 per month

quently, standards have been developed for final installation acceptance tests, daily equipment performance, and routine maintenance procedures.

III. RELIABILITY AND MAINTAINABILITY OBJECTIVES

A primary architectural feature of the No. 4 ESS is the system organization and design which provides dependable hardware and a software structure that can be operated and maintained by craft personnel. These objectives have been accomplished by using reliable circuitry and hardware redundancy with extensive supporting software.

The software design provides centralized maintenance control from the 1A Processor. The processor and the peripheral equipment have configurable redundancy, which is accomplished automatically without affecting service. An automatic backup for the processor semiconductor memory is provided by the disk system, which in turn has a magnetic tape system backup. A detailed description of circuit reliability and system redundancy can be found in Ref. 2.

3.1 Reliability

The basic element of a reliable system is well-designed hardware that includes trouble-detection features and ease of component replacement. The development of the No. 4 ESS is based on a gold metal system for semiconductors and their interconnection. The connector contacts are also gold plated. The basic design features include open-frame convection cooling (rather than fan cooling) and the ability to operate in a temperature range of 30°F to 120°F. The hardware is designed to make per-frame checks and depends on a centralized software maintenance system to automatically reconfigure the hardware in case of trouble, to diagnose the frame reporting irregularities, and to locate the faulty component so it can be replaced by maintenance personnel.

A reliability model was developed for the No. 4 ESS to help translate service objectives into a redundancy plan and to predict long-term performance. The No. 4 ESS reliability model specifies a number of hardware failure modes, determines their impact on performance, and

predicts their likelihood of occurrence.[2] The model was derived principally through analysis of predicted hardware failure rates, system hardware configurations, and predicted repair times. However, the model did not attempt to directly account for the following factors:

(*i*) procedural errors,
(*ii*) change activity,
(*iii*) growth,
(*iv*) retrofits,
(*v*) routine exercise,
(*vi*) software deficiencies, and
(*vii*) hardware design deficiencies.

Instead, the hardware failure rates predicted by the model were scaled to account for procedural errors and software errors based on experience gained from previous systems. No provision was made for generic program retrofits since their frequency is determined by the rate of new feature introduction in each office, which was unknown at that time.

The hardware reliability of the overall system is a function of its size, hardware failure rates, redundancy plan, and mean repair times. Data taken over a 4-year period show that the predicted hardware failure rates essentially have been met. Special repair studies have been conducted which show that the mean time to repair solid faults is 1.25 hours while the mean time to repair intermittent faults is 20.5 hours. As shown in Fig. 8, component failures cause only 11 percent of the service-affecting incidents.

3.2 Maintainability

The No. 4 ESS is designed to perform extensive maintenance functions automatically so that, problems are rapidly corrected and personnel costs are minimized. The initial design provided work centers at each office for maintenance and administration. Experience has shown that centralized maintenance and administration for up to six No. 4 ESSs is possible.

Switching Management Control Centers (SMCCs) have been implemented to centralize the maintenance functions. This has led to the centralization of expertise, reduction of total maintenance personnel, and improved system performance. Additional centralization of Machine Administration Centers and Trunk Operations Centers is planned for the future.[3]

Current field experience demonstrates that the basic system design is highly reliable and that craft-level personnel can maintain the system. Hardware displays, software support tools, and new maintenance documentation (task-oriented practices) have contributed significantly to the performance of the maintenance personnel.

IV. OPERATIONAL FACTORS AFFECTING PERFORMANCE

The principal operational factors affecting performance of a No. 4 ESS are the change and repair activities and some of the environmental factors that can affect No. 4 ESS service. Taken together, they represent a high level of activity in many offices. Section V presents performance statistics which include the service impact of these activities.

4.1 Variety of system configurations

One significant factor is the variety of configurations of the No. 4 ESS. Each installation is engineered to match the service requirements of a particular location; therefore, each office is different. This implies that fault recognition and system recovery programs must be able to operate with any of the possible equipment configurations.

4.2 Evolution of equipment

As mentioned earlier, the equipment comprising the No. 4 ESS has evolved rapidly and many early offices have added each new type of equipment as it became available. The result is a mixture of vintages of equipment, complicating the environment in which system integrity and fault recognition programs must operate. An example is the first No. 4 ESS office, Chicago 7. It has a mixture of core, small (64K) semiconductor and large (256K) semiconductor memory frames. Similarly, in its time-division network, Chicago 7 has original vintage Time Slot Interchange (TSI) frames, a cost-reduced vintage of TSI frames, and the present version called the TSI-B.

Virtually every type of equipment has evolved to incorporate new technology since initial introduction: the Digroup Terminal (DT) has been cost reduced and replaced with the Digital Interface Frame (DIF), the Signal Processor was replaced with the Signal Processor 2 and eventually its signal processing function was incorporated into the DIF, Common-Channel Interoffice Signaling (CCIS) terminals have been improved, and the common control echo suppressor was added and will be superceded by per-trunk echo cancelers. Figure 3 gives a more complete picture of the evolution of No. 4 ESS equipment.

4.3 Growth activity

The rate of growth additions to existing No. 4 ESS systems has increased steadily. Figure 4 shows the number of major growth jobs in progress and the number of new No. 4 ESS offices placed in service during each year since 1976. Nearly two-thirds of the operational No. 4 ESS systems have been expanded with growth jobs. Through the end of 1979, growth activity had added over 900 frames of ESS equipment and provided over 350,000 new terminations, or nearly one-third of all installed No. 4 ESS terminations. Several offices have been expanded

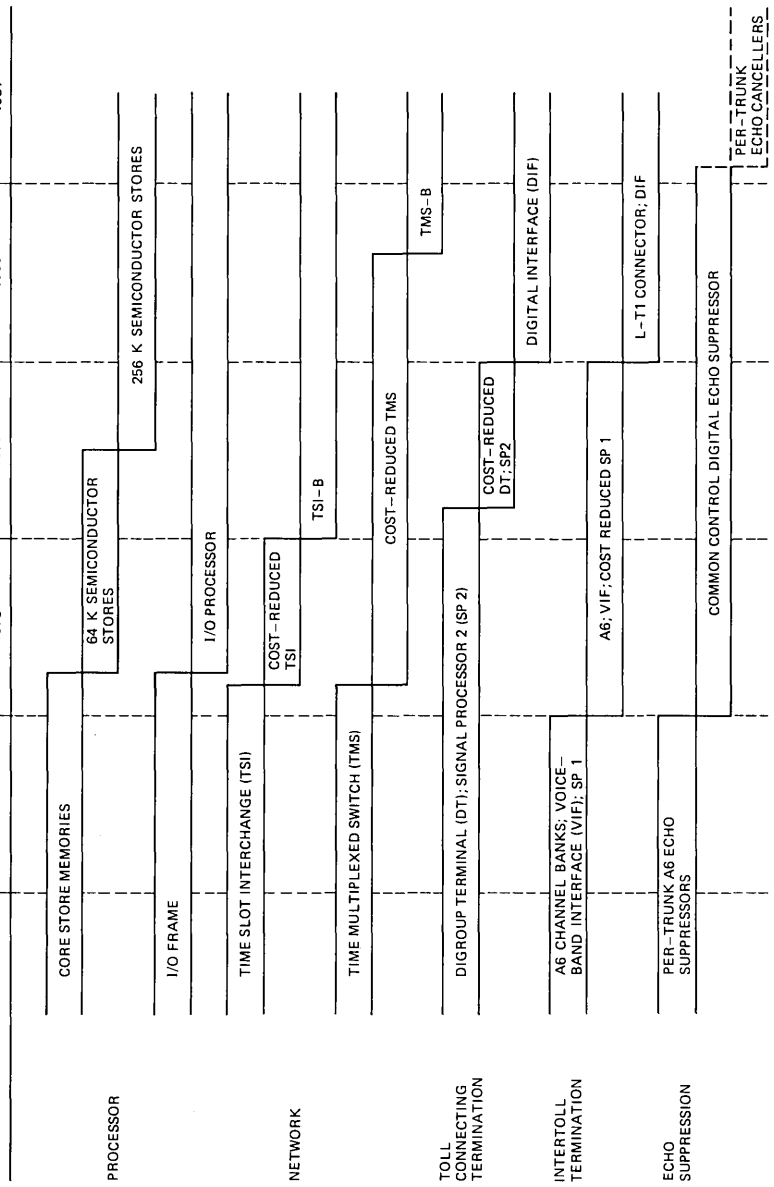


Fig. 3—Evolution of No. 4 ESS equipment.

several times, sometimes with additions providing 30,000 terminations.[4]

The growth process has been designed so equipment can be added without affecting service. However, growth and related activities have been responsible for approximately 5 percent of the service-affecting incidents (see Section 5.2) in the No. 4 ESS. The principal causes of

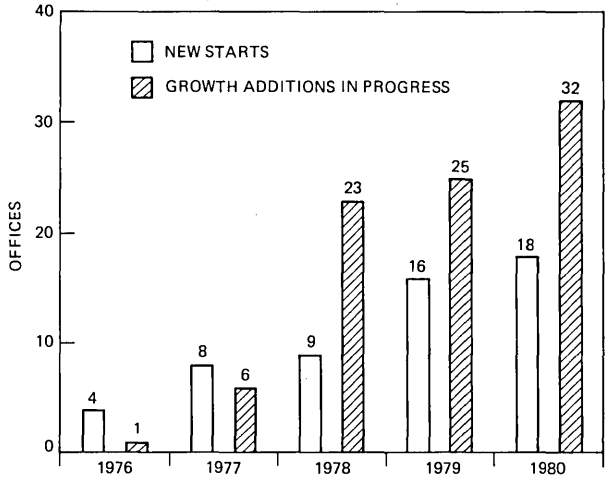


Fig. 4—Trend in No. 4 ESS growth activity.

these incidents have been human error, system software problems, and equipment failure in some of the new equipment shortly after it was made operational. Some of the improvements made in the growth process have been to incorporate temperature stress tests and extra network transmission path checks into selected growth procedures to improve the reliability of the new equipment once it is made operational.

4.4 Hardware change activity

Over 400 Change Notices (CNs) have been prepared by Western Electric to implement hardware changes in No. 4 ESS equipment. The scope of CNs includes wiring changes, circuit pack changes (including firmware updates), documentation, and addition of new types of equipment to existing frames. CNs may be stimulated by design changes initiated by Bell Laboratories or by the discovery of manufacturability problems discovered by Western Electric. All hardware changes are authorized and monitored by the No. 4 ESS hardware change committee. The Western Electric Product Engineering Control Center (PECC) tracks application of CNs in the field.

4.5 Software change activity

Software problems account for 25 percent of all No. 4 ESS service-affecting incidents. These are problems not detected in laboratory system tests or in first application office field tests. Such problems may go undetected until the generic program is introduced into an office with a particular configuration. Some software problems are

caused by incomplete defensive checks and are only stimulated through combinations of failures; others are simply design errors.

Table II shows the size of the No. 4 ESS program with the introduction of each new version. The numbers of problems corrected after the generic was placed in service are also shown. Although the quality of each generic issue is improving, as demonstrated by the decreasing number of service affecting incidents per office (Fig. 7), the number of field problems fixed has increased for each generic. This is a result of greater exposure to different office configurations and the contribution of undiscovered problems carried forward from previous generics. Generally, these software changes are of two types: the relatively few urgent fixes are called out to all offices or transmitted by the Software Change Administration and Notification System and installed with generic utility overwrites; the remainder are installed only when a partial update is distributed to each office. A partial update is a technique for introducing large numbers of program corrections without affecting service. Figure 5 shows a plot of the problems identified, fixes under test, and overwrites delivered to the field for the 4E4 generic.

One of the major reasons the No. 4 ESS has provided excellent service, despite the existence of software problems, is its basic system architecture and software integrity design. It is not technically or economically feasible to detect and fix all software problems in a system as large as the No. 4 ESS. Consequently, a strong emphasis has been placed on making it sufficiently tolerant of software errors to provide successful operation and fault recovery in an environment containing software problems.

Another type of software change activity involves the office data base which includes translations, parameters, trunking, and routing information. Occasionally, corrections and changes are made to the office data base with standard recent change methods and also with generic utility system overwrites.

4.6 Retrofits

A major type of software change activity is a generic retrofit in

Table II—Field problems

Generic	Service Date	Total Size (words)	No. of Field Problems Fixed
4E0	1/76	1160*K*	—
4E1	7/76	1312	249
4E2	6/77	1405	352
4E3	4/78	1663	434
4E4	2/79	1736	411*
4E5	2/80	2162	184*

* Problems fixed in field as of August 26, 1980.

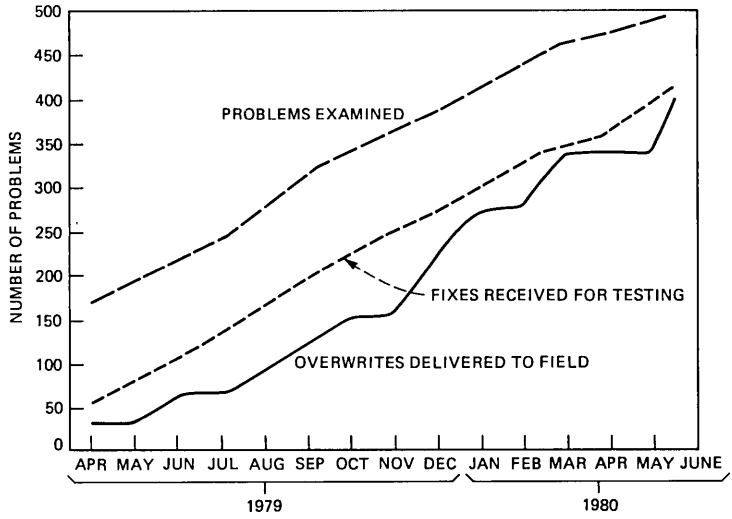


Fig. 5—4E4 generic problem status.

which each No. 4 ESS replaces its current generic program with the latest generic. Current plans call for each office to receive the new generic within a year of its official release. Figure 6 shows the number of retrofits each year since 1976, indicating a large increase as new offices have been added.

A new office data base is compiled for each retrofit. The data base is expanded in anticipation of future growth and also includes a recompiled description of the current office data. Other types of

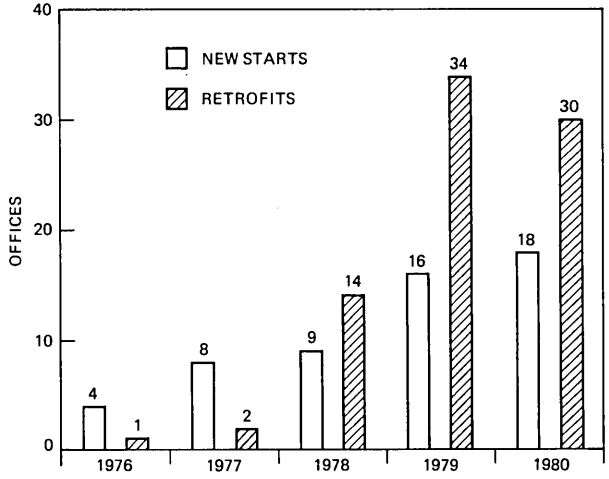


Fig. 6—Trend in No. 4 ESS retrofit activity.

software changes generally made during retrofits, and also once between them (as "midgeneric releases"), are the introduction of new network management software, new trouble-locating procedure tapes that help office maintenance personnel locate faulty circuit packs when diagnostic tests indicate trouble, and new library programs that contain infrequently used test and administrative programs.

4.7 Rearrangements

In addition to hardware changes, software changes, growth and retrofit activity, office performance can also be affected by major rearrangements. Three principal kinds of rearrangements have occurred. Common-Channel Interoffice Signaling terminals in the first 28 offices are being rearranged to improve system reliability. This involved growth of new terminals and execution of a special library program to modify 12 translators in the office data base to effect a new terminal pairing arrangement. The second major rearrangement was a series of activities to allow one office to serve as a gateway office, a function normally planned when an office is first installed. The third type of rearrangement performed was to change the pulse point control for large numbers of frames in one office to increase its reliability.

4.8 Repair

Equipment fails and requires repair on an ongoing basis in No. 4 ESS offices. The average circuit-pack replacement rate for the first quarter of 1980 was 1.7 per day per office. This is half the rate experienced during the first 122 days of Chicago 7 operation in 1976, and it meets the short-term objective of less than two per day per office.[1] To place this number in perspective, a typical No. 4 ESS contains 50,000 circuit packs. In a small fraction of cases, office technicians must use oscilloscopes and probe communication buses and backplane wiring to isolate equipment faults. Such routine repair of equipment often involves several steps, and human error in performing them accounts for 18 percent of the service-affecting incidents.

4.9 Other factors

Although No. 4 ESS offices are well-protected from most external factors, some have had an impact on service. In particular, some offices have been affected by air-conditioning problems, power-distribution failures, failure of non-No. 4 ESS equipment, and static discharge.

V. SERVICE EXPERIENCE

5.1 Service-affecting incidents

To track the performance of No. 4 ESS, the notion of a service-affecting incident (or simply, incident) has been defined as those

equipment failures and major system recovery actions with a significant effect on the customer. Specifically, they include:

(*i*) Hardware failures affecting more than 360 trunks.

(*ii*) System recovery directed Phase 1 and Phases 2, 3, and 4.

(*iii*) System reinitializations.

5.1.1 Hardware failures

Hardware failures affecting more than 360 trunks are called Multiple Unit Failures (MUFs). Originally, MUFs represented half the trunks served by a Voiceband Interface Frame (VIF). With the addition of frames, such as the DIF serving up to 3840 trunks, a MUF is now defined as an outage affecting more than 360 trunks. In duplicated equipment, duplex failures and/or restoral from them, also cause the recovery actions described in Section 5.1.2.

5.1.2 System recovery

When the No. 4 ESS must halt call processing to recover from problems, the result is called a system recovery phase. In a directed Phase 1, all calls associated with a duplex-failed peripheral frame are lost; however, the other stable calls in the system are saved. A directed Phase 1 can have a duration from 1 to 15 seconds.

A Phase 2 is used to recover from memory mutilation or peripheral configuration problems. It checks the integrity of fixed data, such as program store with a hashing algorithm, reconfigures the peripheral complex with peripheral bootstrap (when F-level interrupts implicate the periphery), and initializes most of the call store memory spectrum that is not related to stable calls. A Phase 2 saves stable calls and requires less than 30 seconds if peripheral recovery is not required, and less than 60 seconds if it is. Calls in the dialing state are lost during a Phase 2.

A Phase 3 is used when a complete processor or peripheral reconfiguration is required. It lasts from 1 to 4 minutes, depending on office size, and saves stable calls. Calls in the dialing state are lost, as in a Phase 2.

A Phase 4 is similar to a Phase 3, but it is initiated manually and disconnects all calls.

5.1.3 System reinitialization

A System Reinitialization is a complete reload of the generic program from magnetic tape. It is required only under the most severe cases in which data in program store and both file stores are mutilated. It can take up to 20 minutes and it disconnects all calls.

5.1.4 Number of incidents

When several recovery phases or MUFs are stimulated by the same

event, or follow in succession, they are considered a single incident. All No. 4 ESS service-affecting incidents are recorded and analyzed. The record of these incidents provides an extremely valuable method for evaluating system performance and for guiding efforts to improve it.

Figure 7 is a graph of the number of service-affecting incidents per office per month. The trend indicated is a reduction in the average number experienced by an office to 1.4 per month during the first quarter of 1980. It is significant that a high fraction of service-affecting incidents occur in low traffic periods. Over 55 percent of the "no call processing" time (see Section 5.3.1) has occurred between midnight and 8:00 a.m. to a great extent because routine exercise, complicated repairs, change installation, growth activity, retrofits, and other activities with high risk are generally scheduled during the periods of lowest traffic.

Each service-affecting incident is classified into one or more of the categories shown in Fig. 8. Software design problems account for 25 percent of the total causes. These problems form the basis of an investigation list that is used to guide software current engineering effort. The expected category comprises 16 percent of the incidents. These are cases in which the system reacted as expected, such as planned retrofits, intentional test phases, or when it is impossible to resolve a problem to the proper unit of a duplicated pair and the system must randomly choose the unit to be removed. Duplex frame failures are incidents that occur because a frame is simplex for repair and a fault develops in the active controller. They comprise 11 percent of the total. Unresolved incidents are 13 percent for which sufficient

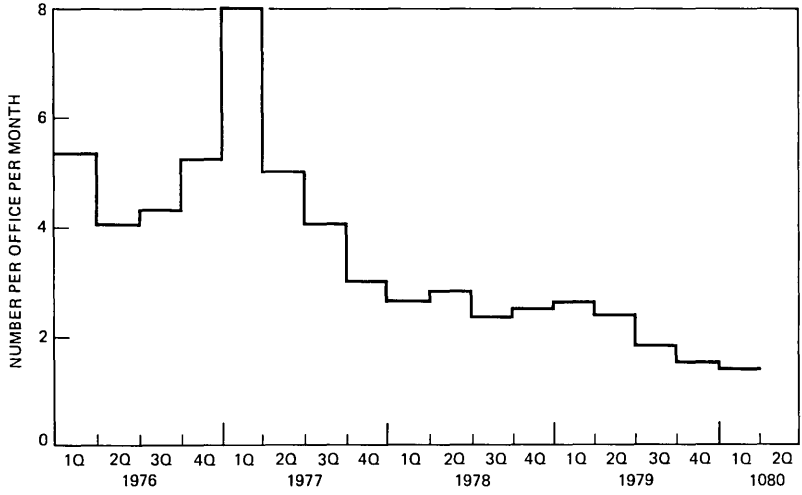


Fig. 7—Service-affecting incidents. The average for the first quarter of 1980 was 1.4 incidents per office per month.

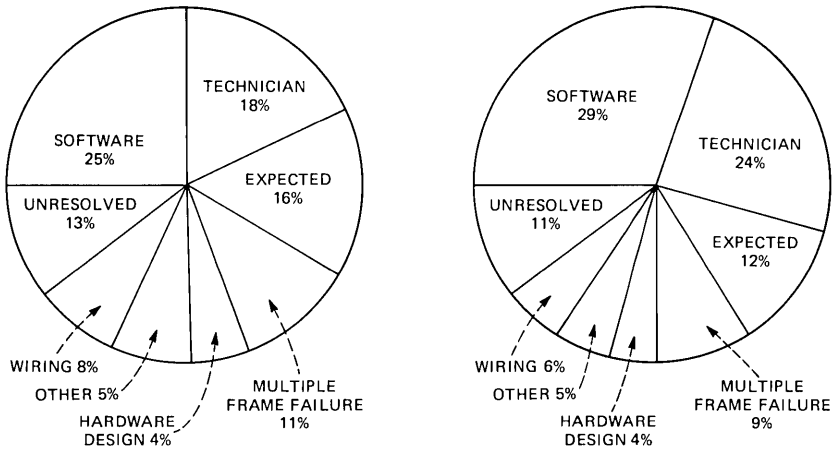


Fig. 8—Causes of service-affecting incidents, cumulative through March 31, 1980. (a) Percent of incidents. (b) Percent of no call processing time.

data to thoroughly analyze the source of the incident is unavailable. Hardware design incidents are the 4 percent caused by the hardware design of a particular frame or subunit. Hardware design problems are considered by the No. 4 ESS hardware change committee and fixes are scheduled as appropriate. Wiring errors account for 8 percent of the incidents and include wiring breaks or loss of insulation integrity as well as errors or wire clippings inadvertently left in equipment when it was being repaired or modified. The technician error category includes operating telephone company craft and Western Electric installer errors, and comprise 18 percent of the total. Figure 8 also shows the causes for service-affecting incidents by their contribution to system no call processing time.

5.2 Customer impact

The principal No. 4 ESS performance measures are those that show the impact of service-affecting incidents on the customer: cutoff calls and denied calls.

Figure 9 shows the rate of calls cutoff by the No. 4 ESS. The first quarter, 1980, rate was 0.18 per 10,000 calls, well below the objective of 1.25 per 10,000. Denied calls are the measure of the No. 4 ESS contribution to the customer's ability to complete calls on demand due to no call processing time. During the first quarter, 1980, the rate of denied calls was 0.28 per 10,000. The trend in the number of calls denied by the No. 4 ESS is shown in Fig. 10. The effect on the customer of denied calls is difficult to measure, since alternate routing strategies

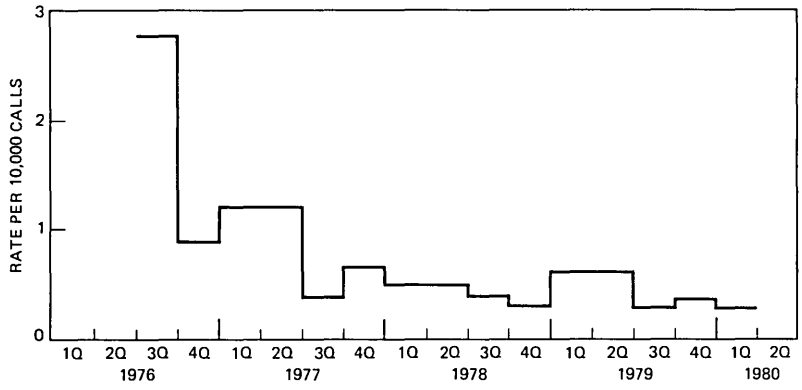


Fig. 9—Cutoff calls. The first quarter 1980 rate was 0.18 per 10,000 calls, well below the objective, which was 1.25 per 10,000 calls.

elsewhere in the network can compensate for some No. 4 ESS denied calls, often allowing the customer to complete the intended call. Both measures show substantial improvement over the period of time the No. 4 ESS has been deployed.

5.3 System performance

In addition to cutoff and denied calls, other performance factors are

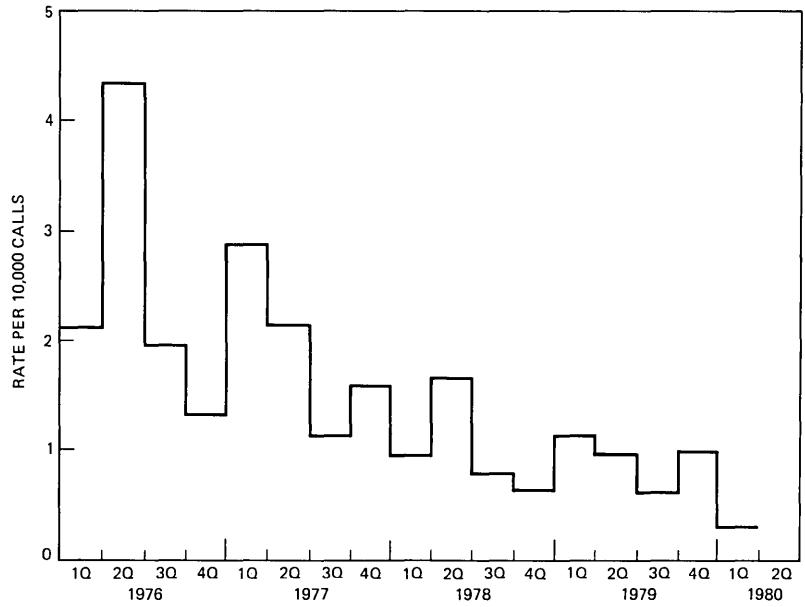


Fig. 10—Denied calls. In the first quarter of 1980, the rate of denied calls was 0.28 per 10,000 calls.

also used to give a more comprehensive measure of system performance. They are system- rather than customer-related measures of system performance and include:

(*i*) no call processing time,
(*ii*) trunk outage time, and
(*iii*) Ineffective Machine Attempts (IMA).

5.3.1 "No call processing" time

No call processing time is often expressed in terms of hours of time in 40 years. It includes outage time required for system reinitialization such as Phases 2, 3, and 4 and directed Phase 1 recovery actions. Note that during the No. 4 ESS no call processing time caused by Phase 2 and Phase 3, all stable calls continue, unless there is also a duplex failure of network or network interface equipment. Figure 11 illustrates that the long-term trend has been an improvement in "no call processing" time to a first quarter, 1980, rate of 9.9 hours in 40 years. Since generic retrofits and data base updates require use of an intentional Phase 3 during the lowest traffic periods, there is a built-in requirement that approximately 1 hour in 40 years of this total be used for this purpose. Customer impact is minimal because network management controls applied as part of the retrofit procedure virtually eliminate any customer impact. The rate of 9.9 hours in 40 years is comprised of

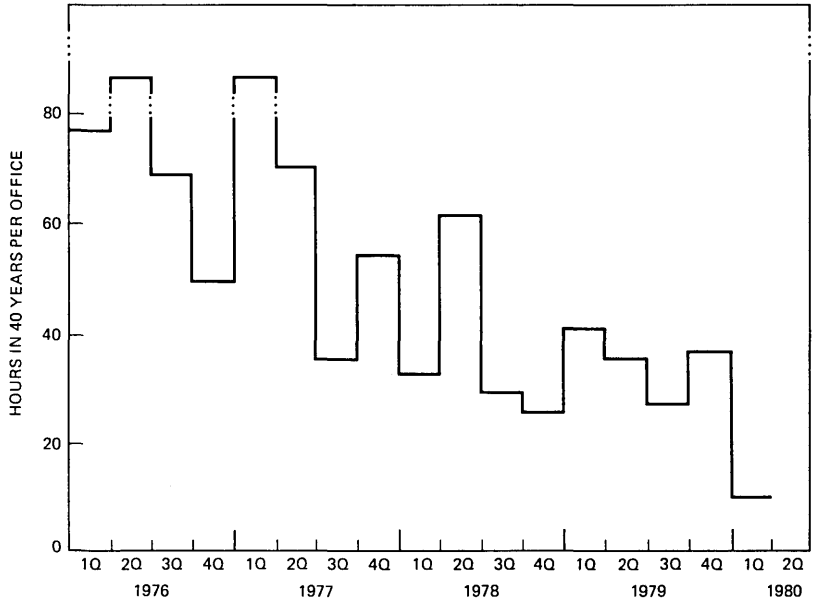


Fig. 11—No call processing. The first-quarter 1980 rate was 9.9 hours in 40 years. The objective was 2.0 hours in 40 years.

all of the factors shown in Figure 8. It is significant that factors, such as procedural errors and software deficiencies, that could not be specifically modeled (see Section III), account for nearly two-thirds of all downtime. Consequently, the internal objective of 2 hours in 40 years of total system unavailability is under review. Nevertheless, no call processing time has steadily improved as maintenance and reliability enhancements have been added to the system.

Figure 12 shows the effect of two recent enhancements. It presents overlapping histograms showing the distribution of no call processing incidents for two 6-month periods, one ending on March 31, 1979, and another ending 1 year later. The significance of the first histogram is that is represents No. 4 ESS performance before the directed Phase 1 feature was available. The directed Phase 1 was introduced in the 4E4 generic program and has been deployed both in new offices and through generic program retrofits. By March 31, 1980, all offices had the directed Phase 1 feature. Normally, the directed Phase 1 takes about 2 seconds to initialize a duplex-failed TSI frame. Prior to the directed Phase 1, a 1- to 3-minute Phase 3 was required. The significance of the second histogram is that the directed Phase 1 shifted the distribution so that 34 percent of all no call processing incidents require less than 30 seconds as compared with 2 percent prior to directed Phase 1. An additional enhancement, introduced late in 1979, was a shortened Phase 2 when no peripheral equipment was suspected by system integrity programs. This also reduces the no call processing time.

5.3.2 Trunk outage time

Trunk outage time is the measure of hardware failures, such as duplex-failed equipment or MUFs. Note that no call processing time is not included in trunk outage time measurements. Figure 13 shows a graph of No. 4 ESS trunk outage time. During the first quarter of 1980, the system performance was 38.0 minutes of outage per trunk per year compared with an objective of 28.0. Several maintenance enhancements are planned to help bring No. 4 ESS performance closer to this objective.

5.3.3 Ineffective machine attempts

Some customer attempts to originate calls result in noncompleted calls. The No. 4 ESS has a large and precise ineffective-attempt reporting system that measures call failure statistics and allows an analysis of chronic problems. Over 300 call-failure modes are defined, including customer errors, failure of switching machines or transmission media connected in an incoming mode to the No. 4 ESS, failure of the No. 4 ESS to establish a cross-office connection, and a failure of the switching

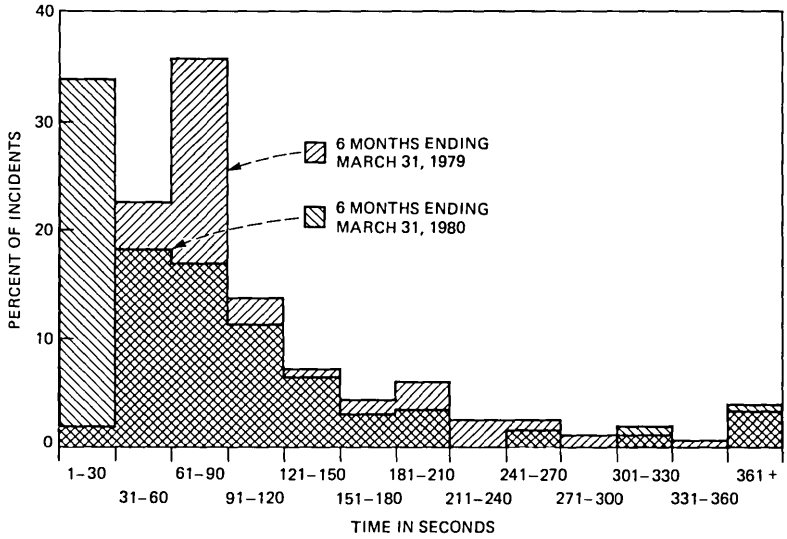


Fig. 12—Incident duration for two six-month intervals that show the impact of the directed Phase 1 and shortened Phase 2.

machine or transmission media connected in an outgoing mode to the No. 4 ESS. A subset of the total ineffective attempts is classified as an IMA. These include calls that must be terminated with incoming, connecting or outgoing reorder tone, vacant code announcements, or no-circuit tone.

Figure 14 shows that the average adjusted domestic IMA performance has remained relatively constant at a little over 1 percent of all attempts. The rate during the first quarter of 1980 was 1.02 percent,

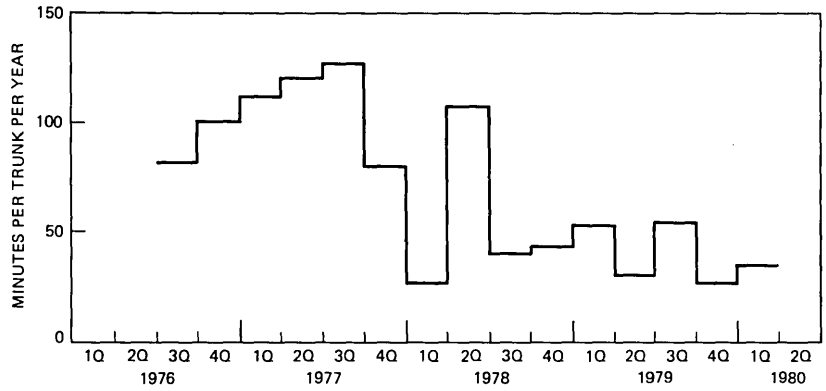


Fig. 13—Trunk minutes out of service. For the first quarter of 1980 the system performance was 38.0 minutes of outage per trunk per year. The objective was 28.0.

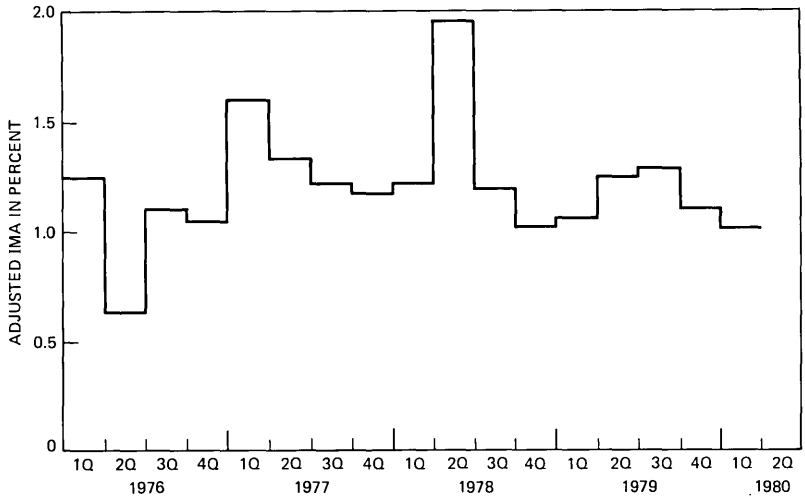


Fig. 14—Ineffective machine attempts. The first quarter of 1980 had a rate of 1.02 percent. The original objective was 1.25 percent.

meeting the original objective of 1.25 percent. The rate for calls to other countries is higher. A study of the specific failures shows that the No. 4 ESS and outgoing trunks contribute to less than 0.01 percent of the total number of IMAs. Most failures originate from irregularities in the incoming network. Further analysis shows that in large metropolitan systems, such as those in Chicago and New York City where common control Class 5 offices with multifrequency signaling or CCIS are used, the reorder component of IMA for domestic calls ranges between only 0.2 to 0.3 percent. However, where step-by-step or early vintage crossbar switches are used, the reorder IMA ranges between 2 and 3 percent even though the equipment is properly maintained. This can frequently be attributed largely to outside plant problems not screened by these systems. The IMA data are effective in identifying network problems, and also serve as a continuous check on network performance.

5.4 Interrupts

One of the most closely watched system maintenance indicators in No. 4 ESS is the level of system interrupts. They generally indicate an unexpected response from a system action. For example, an equipment failure that affects a path through the time-division network may cause interrupts. (For a more complete description of system interrupts, see Refs. 2 and 5.)

Although interrupts do not directly affect the customer, an objective has been set to help manage system maintenance activity. When the

interrupt level rises, more attention needs to be spent on maintenance. The original empirical interrupt objective of less than 50 per day has been tightened to an average of less than 40 per day. Some small offices have an objective that is more stringent since they have less equipment. The average number of interrupts per office during the first quarter of 1980 was less than 25 per day, meeting the objective.

VI. MANAGING PERFORMANCE

6.1 Ongoing development

The original design and implementation of the No. 4 ESS are key factors in allowing the system to provide the current level of service. However, another key ingredient has been the management of No. 4 ESS performance.

Each service-affecting incident is recorded in a data base and analysis is performed monthly to track the overall performance. When analysis has shown that specific improvements can help improve system performance, they become candidates for features to be developed as part of the next generic program release. Committees review each new feature candidate for its impact on system resources, the development effort required, and the feature's value relative to other candidates. The directed Phase 1 was such a feature; it was proposed when analysis showed it could reduce system no call processing time.

6.2 Current engineering

In addition to new features aimed at improving performance, an ongoing effort also exists to identify problems in existing systems and to deliver fixes. Specific responsibility for carrying out this effort is assigned to a group that works closely with developers to generate the necessary fixes. Much of this effort is directed toward the large generic program. However, with the rapid introduction of new equipment, all modifications to existing hardware designs are also tracked by the hardware change committee.

6.3 Acceptance tests

In addition to its basic design, No. 4 ESS performance is affected by how well each new system is installed and how in-service systems are operated. New systems must meet rigorous operational readiness tests and final verification acceptance tests before they are turned over from the installer to the operating telephone company (OTC). Before the OTC places the system in service, it must meet another set of performance criteria, of which the 7-day sliding interrupt average is the most visible. These performance criteria are specified in Bell System Practices and Western Electric Installation Handbooks expressly to help the OTCs manage the quality of initial service they offer. After initial service,

extensive service results performance measurements or indices are used to help judge the effectiveness of the team operating each No. 4 ESS.

6.4 Managing deployment

Besides the performance of each individual No. 4 ESS, performance management has been extended to help govern the rate at which new systems are deployed with new software and hardware. Specific recommendations have been published in cooperation with AT&T that establish intervals after the first application office for subsequent new offices and for the beginning of the generic retrofit program. These recommendations limit the initial exposure of new software and hardware until sufficient experience is gained under actual operating conditions to allow rapid deployment with confidence that service performance standards will be maintained.

The recommendations also specify the composition and duties of steering and cutover committees for each new system and major growth job. Recent experience indicates that these committees can be very effective and are key ingredients in the smooth transition from an earlier system to a new No. 4 ESS.

As indicated in Section III, there are many demands for changes, rearrangements, and additions to existing systems. To help manage this high level of activity, as well as arbitrating schedule conflicts for new systems, retrofits, and data base updates, an Implementation Review Committee was formed with representatives knowledgeable in OTC needs, Western Electric production and installation capacities, and Bell Labs development capabilities and schedules. One of its tasks is to help manage peak demands, such as the high fraction of systems requesting spring service dates to help meet busy season traffic demands.

VII. SUMMARY

The No. 4 ESS has been incorporated successfully into the Bell System and international telecommunications network. Since the cutover of the first system in Chicago in January 1976, 51 systems terminating over 1,000,000 trunks have been put into service. During this period, the hardware and software have evolved to include the latest technology which has made possible additional equipment cost savings and a reduction in space and power requirements.

Experience with the No. 4 ESS has confirmed the original design criteria for improved reliability and maintainability in stored programmed control systems as follows:

(*i*) Reliability, maintainability, and administrative features must be included in the original architecture of the entire system.

(*ii*) Software integrity features are necessary to allow large systems to perform successfully in an environment containing software problems.

(*iii*) Automatic and semiautomatic maintenance aids are mandatory for maintaining modern systems.

(*iv*) Many factors other than component failures cause reliability problems and must be considered in basic design decisions.

(*v*) Built-in facilities for continually measuring performance parameters are needed to make sure that performance criteria are met and to identify where improvements are required.

(*vi*) Performance criteria should be based on customer impact.

Inclusion of these concepts in the No. 4 ESS has been a major factor in its excellent performance and rapid deployment.

VIII. ACKNOWLEDGMENTS

Many of the performance statistics presented in this article were compiled by K. J. Rice. The authors would especially like to thank her for this contribution. Also, the steadily improving performance trends are attributed to the dedicated effort by No. 4 ESS designers, the diligence of the people operating No. 4 ESS systems, and the excellent support they receive from the SMCCs, the No. 4 ESS National Electronic Switching Assistance Center, and the Western Electric PECC.

REFERENCES

1. H. B. Compton et al., "System Integration and Early Office Planning," B.S.T.J., *56,* No. 7 (September 1977), pp. 1279–96.
2. J. J. Kulzer, "Systems Reliability—A Case Study of No. 4 ESS," INFOTECH State of the Art Conference on Computer System Reliability, London, England, June 1977.
3. P. K. Giloth and J. A. Giunta, "No. 4 ESS Operations and Performance," Int. Switching Symp., Paris, France, May 1979, pp. 1071–8.
4. E. A. Davis, "No. 4 ESS Growth: Serving Increased Toll Switching Needs," Bell Labs Record, *58,* No. 5 (May 1980), pp. 146–51.
5. M. N. Meyers, W. A. Routt, and K. W. Yoder, "Maintenance Software," B.S.T.J., *56,* No. 7 (September 1977), pp. 1139–67.
6. P. K. Giloth and H. E. Vaughan, "Early No. 4 ESS Field Experience," Int. Switching Symp., Kyoto, Japan, October 1976.
7. R. A. Bruce, P. K. Giloth, and E. H. Siegel, Jr., "No. 4 ESS—Evolution of a Digital Switching System," IEEE Trans. Commun., *COM-27,* No. 7 (July 1979), pp. 1001–11.
8. P. K. Giloth, "No. 4 ESS Reliability and Maintainability Experience," 1980 Proc. Annu. Reliability and Maintainability Symp, pp. 388–392.

ACRONYMS AND ABBREVIATIONS

ADF	arranged with data features
ADS	announcement distribution system
ALU	arithmetic logic unit
APUF	autonomous peripheral unit failure
ASW	all seems well
AWG	American Wire Gauge
AUTOVON	automatic voice network
BLM	base level maintenance
BOOTCNTL	bootstrap control program
BSRF	Bell System reference frequency
CAMA	centralized automatic message accounting
CAROT	centralized automatic reporting on trunks
CC	control clock
CCIS	common channel interoffice signaling
CCITT	International Telegraph and Telephone Consultative Committee (Comite Consultatif International Telegraphique et Telephonique)
CESR	controller error source register
CI	control interface
CLF	clear forward
CMOS	complementary MOS
CMP	complete
CMS	circuit maintenance system
CPU	central processing unit
CR	call register
DCON	diagnostic control
DDD	direct distance dialing
DDI	direct data insert
DIC	digital interface controller
DIF	digital interface
DIFRINTR	digital interface frame interrupt recovery package
DIPs	dual in-line packages
DIU	digital interface unit
DMA	direct memory access
DP	dial pulse
DT	digroup terminal
DTCRs	dedicated time-slot interchange connection registers
ECL	emitter-coupled logic
EIA	Electronic Industries Association
EIB	extended internal buses
ENSIG	enable signaling

EPL	electronic programming languages
EPLX	electronic programming language extended
ESR	error source register
ESS	electronic switching system
EXEC	executive controller
FERA	failure error analysis
F/R	framing and receiving
FRDI	frame request and diagnostic interface
IB	internal bus
IBMUX	internal bus multiplexer
IC	integrated circuit
ICT	incoming trunk
IGFET	insulated gate field-effect transistor
IMA	ineffective machine attempts
INWATS	inward wide area telecommunications service
IOP	input/output processor
IPUB	interface to peripheral unit bus
LSI	large-scale integration
MAC	machine administration center
MAPT	mass announcement phasing table
MAR	mass announcement register
MAS	mass announcement system
MAUs	mass announcement units
MCC	master control center
MEDIC	message dispenser and coordinator
MF	multifrequency
MMC	maintenance microcomputer
MOC	maintenance operations center
MOS	metal oxide semiconductor
MP	maintenance processor
MSC	media stimulated calling
MSS	mass announcement support system
MSTAT	mass announcement system status table
MUDS	machine updatable data system
MUF	multiple unit failures
NCLK	network clock
NCSU	network clock synchronization unit
NMOS	N-channel metal oxide semiconductor
OGT	outgoing trunk
OGTS	outgoing trunk select
ONAC	operations network administration center
OOF	out of frame
OP	operation
OSC	oscillator

OTC	operating telephone company
PAS	public announcement service
PC	peripheral controller
PCM	pulse-code modulated
PCS	power control switch
PCSINH	per-channel signaling inhibit
PECC	product engineering control center
PIC	peripheral interface controller
PIDENT	program identification
PMOS	peripheral maintenance operating system
PROM	programmable read only memory
PU	peripheral unit
PUB	peripheral unit bus
PUC	peripheral unit control (controller)
PUCB	peripheral unit control bus
PUEAB	peripheral unit enable/address bus
PUF	peripheral unit failure
PUFR	peripheral unit fault recovery
PURB	peripheral unit reply bus
PUWB	peripheral unit write bus
PWB	printed wiring board
RAM	random access memory
RCRRT2	remove recent change
RCV	receive
REF	reference
REG	register
REPODISP	report dispenser
ROM	read only memory
RPT	report scan
RPY	reply
SFPD	superframe pattern detector
SMCC	switching management control centers
S/P	serial to parallel
SP1	signal processor type 1
SYSCLK	system clock
TDN	time-division network
TEC	terminal equipment center
TMR	trunk maintenance register
TMS	time multiplexed switch
TMT	transmit
TOC	trunk operations center
TOPIC	toll peripheral configuration
TR	trunk register
TS	time slot

TSI	time-slot interchange
TSI SPC	time-slot-interchange switching and permuting circuits
TSN	trunk scanner number
TSPS	traffic service position system
TTL	transistor-transistor logic
TTY	teletypewriter
TU	time of unit
UART	universal asynchronous receiver transmitter
UMB	unit maintenance
UTE	unitized terminal equipment
VIF	voiceband interface frame

CONTRIBUTORS TO THIS ISSUE

Thomas W. Anderson, B.S.E.E., 1974, Michigan Technological University; M.S., 1977, University of Minnesota; Bell Laboratories, 1977—. Mr. Anderson initially was involved in both the hardware design and firmware development for the No. 4 ESS Mass Announcement System. He is currently working in the area of microprocessor applications for new network services in the No. 4 ESS.

John H. Bobsin, B.S.E.E., 1967, M.S.E.E., 1968, and Ph.D., 1975, Polytechnic Institute of New York; Bell Laboratories, 1969—. Mr. Bobsin has been engaged in development of wire pair and fiber optic digital transmission systems, and electronic switching system peripheral equipment. He is presently involved in the development of systems for new telecommunications services.

Paul D. Carestia, B.S.E.E., 1969, Colorado State University; M.S.E.E., 1971, Northwestern University; M.B.A., 1975, University of Chicago; Bell Laboratories, 1969—. At Bell Laboratories, Mr. Carestia initially worked on the No. 4 ESS in the areas of service circuit diagnostics and control and trunk maintenance. He supervised the development of domestic and international call processing features for the No. 4 ESS and currently supervises the development of maintenance and network management capabilities for export and domestic applications of the No. 4 ESS.

Richard F. Cook, B.S.E.E., 1953, Northeastern University; Bell Laboratories, 1953—. Mr. Cook initially worked on circuit design for a naval air-to-ground data communications system and UNICOM, a military communications system. He later worked on diagnostic programs, trouble location procedures, and microprocessor programs for No. 1 ESS Arranged with Data Features and No. 4 ESS. He is currently engaged in maintenance software designs for new teleconferencing equipment. Member, Eta Kappa Nu, Tau Beta Pi.

Edward A. Davis, B.S.E.E., Michigan State University; M.S.E.E., Northwestern University; Bell Laboratories, 1968—. Upon joining Bell Laboratories, Mr. Davis began designing an automatic billing circuit for the No. 1 ESS. He later worked on an experimental wideband network for *PICTUREPHONE*® service signals. In 1972, he worked on the design of the Input/Output circuit for the 1A Processor. Two years later, he became involved in the development of the No. 4 ESS

growth procedures. In 1976, he was promoted to assistant engineering manager on a rotational assignment with the AT&T Technical Policy Studies Group. In 1978, he returned to Bell Labs, supervising the No. 4 ESS System Growth and Project Coordination Group. In 1980, Mr. Davis became supervisor of the No. 4 ESS Field System Evaluation Group, which is responsible for analyzing the performance of the No. 4 ESS, solving field problems, and participating in the design of features aimed at improving its performance. He is currently Supervisor of the No. 4 ESS System Test and Planning Group which is responsible for introducing new system software. Member, IEEE, Tau Beta Pi, Eta Kappa Nu, Phi Eta Sigma, Tau Sigma.

Rudolph J. Frank, B.S. (Electrical Engineering), 1966, Seattle University; M.S. (Electrical Engineering), 1968, Ph.D. (Electrical Engineering), 1971, Oregon State University; Pacific Northwest Bell, 1964–1966; Bell Laboratories, 1971—. At Pacific Northwest Bell, Mr. Frank was an electronics data processing supervisor in the comptrollers division. At Bell Laboratories he worked in the Traffic Service Position System laboratory. In 1975, he was designated Bell Laboratories Visiting Professor to Southern University (Baton Rouge, La.). Mr. Frank became supervisor of the No. 4 ESS Network Management Control Group in 1976 and has worked on several large software development projects. He was recently awarded a Sloan fellowship to Stanford University.

Paul K. Giloth, B.A., 1942, Beloit College; B.S.E.E., 1947, M.S.E.E. Northwestern University; Illinois Bell Telephone Company, 1947–1950; Bell Laboratories, 1951—. Mr. Giloth worked initially on analog computer simulators for military applications. Following this, he supervised development of TRADIC, a transistorized bombing and navigation system, and the guidance computers for the NIKE-ZEUS ABM system. In 1961, he was appointed head of the UNICOM Test Model Department and was responsible for digital terminal equipment and the store-and-forward message portion of the UNICOM system. In 1963, he became head of the Data Switching Systems Department and was responsible for development of the No. 1 ESS arranged with data features (adf) Data Switching System. From 1970 to 1972 he was responsible for automatic voice network (AUTOVON) development. As head of the No. 4 ESS Coordination and Evaluation Department, he was responsible for developing software tools for system evaluation and the planning, engineering, and integration of early No. 4 ESS offices. Mr. Giloth is now head of the No. 4 ESS Project Management and Applications Department. He has been associated with the No. 4

ESS since 1969. He is a registered engineer in Illinois. Senior Member, IEEE; Member, Sigma Xi.

Lando Gingerich, Jr., B.S.E.E., 1959, University of Iowa; Western Electric Company, 1959-63; Bell Laboratories, 1963—. Mr. Gingerich has been associated with the development of control systems for the Nike-Zeus, *TELSTAR®*, and Safeguard projects. Since 1971, he has been involved in the physical design of the No. 4 ESS.

Kathryn M. Andersen Hoppner, B.S.E.E., 1976, University of Notre Dame; M.S.C.S., 1978, Northwestern University; Bell Laboratories, 1976-1981. As a member of the Digital Terminal Recovery Program Group, Ms. Hoppner was responsible for the development of maintenance software to support the introduction of the Digital Interface frame into the No. 4 ESS system. In addition to this responsibility, she developed portions of the microprocessor software which operates the Digital Interface frame. Member, IEEE.

Fred S. Hudson, A.A.S., 1968, DeVry Institute of Technology; Bell Laboratories, 1968—. Mr. Hudson's initial assignment was in the maintenance software area for the ADF system. In 1969, he joined the No. 4 ESS Maintenance Planning and Design Department. Since then he has been involved in the design and development of diagnostics, trouble location procedures and data bases, diagnostic control programs, routine exercise programs, craft-machine interface programs, fault recovery programs, error analysis programs, and system recovery programs. He currently is responsible for the 4E7 generic planning for the No. 4 ESS. He is also involved in defining the maintenance system architecture for the Advanced Mobile Phone Service.

Robert J. Keevers, B.S., 1950, U. S. Naval Academy; M.E.E., 1957, Rensselaer Polytechnic Institute; M.S.E.E., 1962, New York University; Bell Laboratories, 1957—. Mr. Keevers is supervisor of the Network Switching Requirements, Numbering Plan, and CCITT Support Group. His prior toll planning work has included IDDD, 800 Service, and the Traffic Service Position System. His present responsibilities with the No. 4 ESS generic 5 have focused on the Mass Announcement System. Mr. Keevers has been a long time delegate to CCITT Study Group XI and is a professional engineer in the state of New York.

H. Mann, B.A. (Mathematics), 1950, Brooklyn College; M.S. (Electrical Engineering), Columbia University; Bell Laboratories, 1954—.

Mr. Mann worked on an experimental pulse-code modulated (PCM) system and transmission of PCM over short-haul microwave systems. He designed the command decoder for Telstar. He supervised a group responsible for the design of improved single-frequency signaling systems and signaling test sets, and more recently supervised the design of the Digroup Terminal and Digital Interface Unit, the digital transmission interfaces for the No. 4 ESS. Since 1979, he has been in charge of a custom large-scale integration department. Member, IEEE, Pi Mu Epsilon.

Mohamed A. Marouf, B.Sc. (Electrical Engineering), 1970, Alexandria University, Egypt; M.S. (Electrical Engineering), 1974, Stanford University; Ph.D. (Electrical Engineering), 1977, University of Southern California; Bell Laboratories, 1977—. At the University of Southern California, Mr. Marouf did research in the design of totally self-checking digital circuitry including contributions to the design of self-checking checkers for m-out-of-n codes and Berger codes. Since joining Bell Laboratories, he has been working on the No. 4 ESS development. Member, IEEE.

Karl E. Martersteck, Jr., B.S. (Physics), 1956, University of Notre Dame; M.S. (Electrical Engineering), 1961, New York University; Bell Laboratories, 1959—. Mr. Martersteck initially worked in the Silicon Transistor and Integrated Circuit Department. In 1964 he transferred to Bellcomm Inc., where he was involved with systems engineering and analysis in support of the Apollo Lunar Landing Program and Skylab Program. Mr. Martersteck returned to Bell Laboratories in 1972, where he is currently executive director of the Network Switching Services Development Division. This position includes responsibility for stored program controlled systems that provide digital toll switching and operator services functions. Member, IEEE.

Reinhard Metz, B.S.E.E., 1972, Illinois Institute of Technology; M.S.E.E., 1973, University of Illinois; Bell Laboratories, 1972—. Mr. Metz has been involved in No. 4 ESS hardware design, including service circuits and studies of digital filter applications. Recently responsible for the Network Clock Synchronization hardware development, he is currently supervising a group designing network data features.

Richard J. Milczarek, B.S.E.E., 1973, Illinois Institute of Technology; M.S.E.E., 1977, Northwestern University; Bell Laboratories, 1977—. Mr. Milczarek has been engaged in the development of hard-

ware for the peripheral circuitry of the No. 4 ESS. Member, IEEE, Tau Beta Pi, Eta Kappa Nu, Phi Eta Sigma, Pi Delta Epsilon, International Solar Energy Society.

Steve Panyko, B.E.E., 1969, City College of New York; M.S.E.E., 1971, Columbia University; Bell Laboratories, 1969—. During his first nine years at Bell Laboratories, Mr. Panyko was involved with exploratory systems in in-band signalling systems and No. 4 ESS transmission terminal equipment. He worked on hardware design and participated in the generic integration of the echo suppressor terminal. Mr. Panyko supervised a group with hardware, firmware, and diagnostic software responsibility for the Digital Interface Controller, as well as a group with systems and software design roles. Currently he is head of the Advanced Processor Design Department. Member, Tau Beta Pi, Eta Kappa Nu, Sigma Alpha.

E. L. Reible, A.M.E., 1973, North Central Technical Institute; Bell Laboratories, 1973—. Mr. Reible has been working in No. 4 ESS physical design, including the Network Clock Synchronization Unit project. He is currently participating in the physical design of Network Services Frame hardware.

Frances B. Strebendt, B.S., 1965, Eastern Illinois University; Bell Laboratories, 1965—. Ms. Strebendt has worked on operational software for the No. 1 ESS Arranged with Data Features (ADF) and for No. 4 ESS. Ms. Strebendt was responsible for the design of Announcement Handling programs for the Mass Announcement System feature in No. 4 ESS. At present Ms. Strebendt is designing administrative software for the No. 4 ESS Basic Export feature.

John E. Waninski, B.S.E.E., 1966; M.S.E.E., 1967; Ph.D., 1971, Illinois Institute of Technology; Bell Laboratories, 1971—. Mr. Waninski initially worked on performance and capacity analysis of No. 4 ESS. He was later responsible for traffic engineering and for various No. 4 ESS feature planning studies. Currently he supervises a group responsible for network operations planning for new telecommunications systems. Member, IEEE, Eta Kappa Nu, Tau Beta Pi, Sigma Xi.

David F. Winchell, B.S.M.E., 1969, Tufts University; S.M.M.E., 1970, M.I.T.; Bell Laboratories, 1969—. At Bell Laboratories, Mr. Winchell began working on the physical design of No. 4 ESS network

frames. Since 1977, he worked on the firmware for the NCSU *BELLMAC®-8*. He is now responsible for the call-processing software for the teleconferencing feature in the Network Services Frame. Member, Tau Beta Pi.

John Van Zweden, B.S.E.E., 1968, University of Michigan; M.S.E.E., 1970, Purdue University; Bell Laboratories, 1969—. Mr. VanZweden has worked on the physical design of digital lines, digital multiplexers, and several No. 4 ESS transmission interfaces including the Digital Interface. He presently supervises a group responsible for the physical design of digital terminals. Member, Eta Kappa Nu, Tau Beta Pi.

THE BELL SYSTEM TECHNICAL JOURNAL is abstracted or indexed by *Abstract Journal in Earthquake Engineering*, *Applied Mechanics Review*, *Applied Science & Technology Index*, *Chemical Abstracts*, *Computer Abstracts*, *Current Contents/ Engineering, Technology & Applied Sciences*, *Current Index to Statistics*, *Current Papers in Electrical & Electronic Engineering*, *Current Papers on Computers & Control*, *Electronics & Communications Abstracts Journal*, *The Engineering Index*, *International Aerospace Abstracts*, *Journal of Current Laser Abstracts*, *Language and Language Behavior Abstracts*, *Mathematical Reviews*, *Science Abstracts* (*Series A, Physics Abstracts*; *Series B, Electrical and Electronic Abstracts*; *and Series C, Computer & Control Abstracts*), *Science Citation Index*, *Sociological Abstracts*, *Social Welfare*, *Social Planning and Social Development*, and *Solid State Abstracts Journal*. Reproductions of the Journal by years are available in microform from University Microfilms, 300 N. Zeeb Road, Ann Arbor, Michigan 48106.


Bell System