No. 1 ESS Maintenance Plan

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

(Manuscript received January 28, 1964)

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

I. OBJECTIVES

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

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

II. OVER-ALL PLAN

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

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

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

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

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

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

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

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

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

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

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

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

III. DUPLICATION AND SWITCHING

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

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

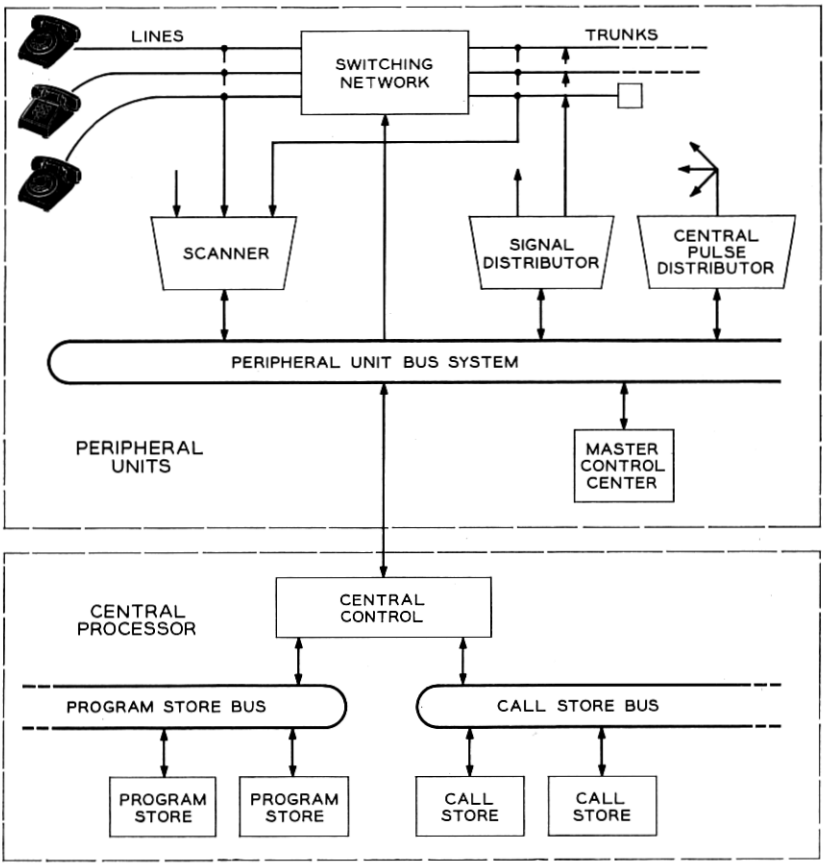

Fig. 1 — No. 1 ESS block diagram.

3.1 *Reliability Predictions*

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

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

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

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

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

---

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

† Component failure rates assumed, in FITs:

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

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

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

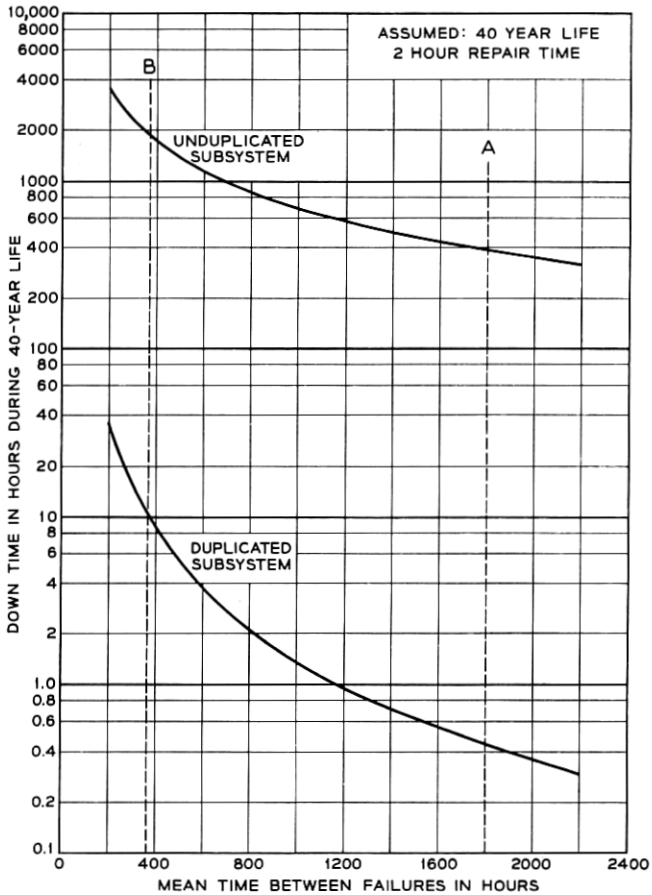

Fig. 2 — Down time vs mean time between failures for system with assumed 40-year life, two-hour repair time.

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

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

3.2 *Duplication and Switching in the Central Processor*

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

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

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

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

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

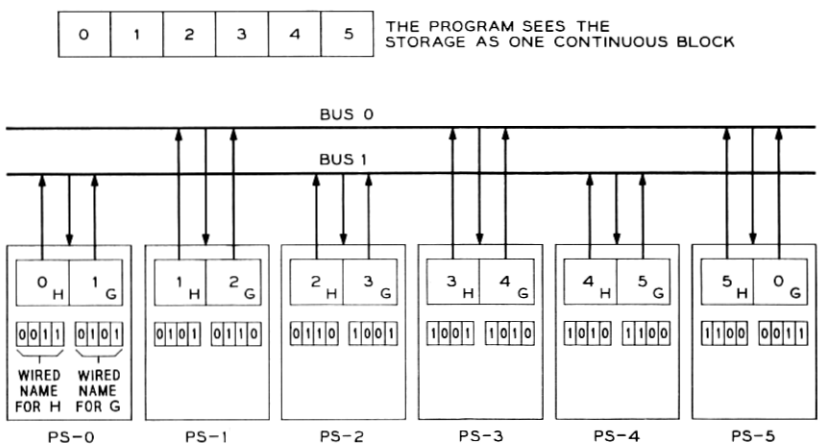

Fig. 3 — Program store duplication, six-store office.

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

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

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

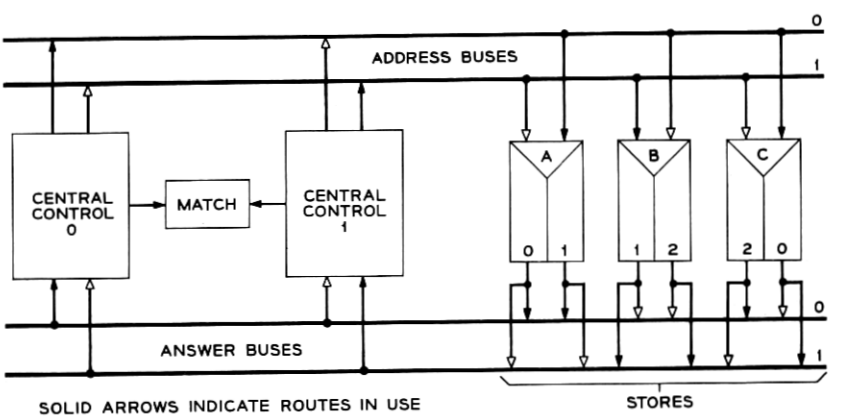

Fig. 4 — Three program store system — normal configuration.

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

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

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

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

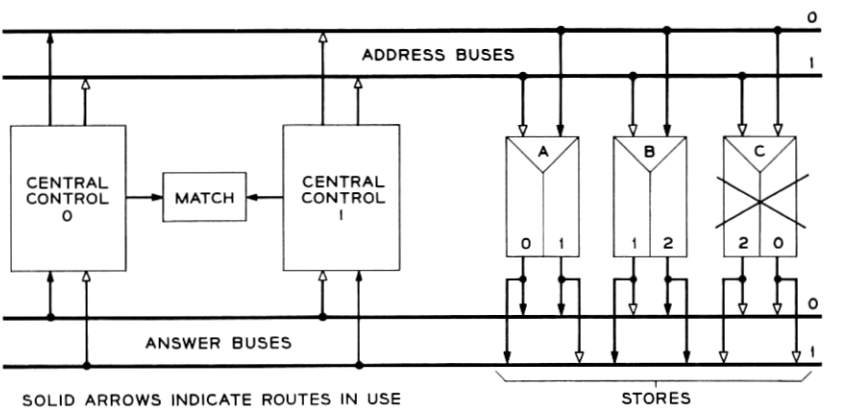

Fig. 5 — Three program store system — store C out of service.

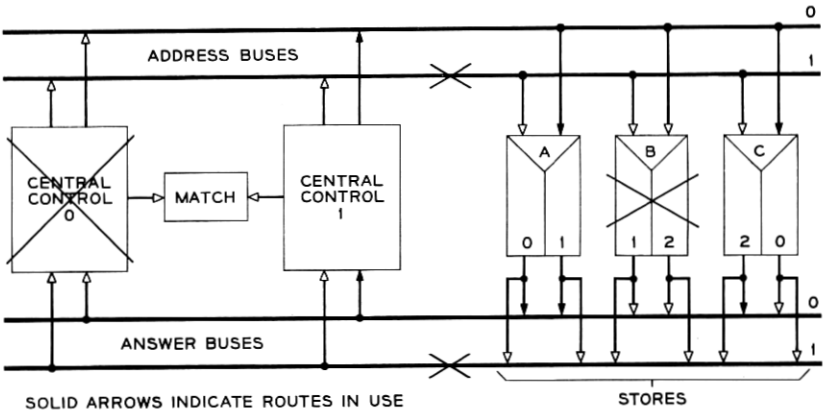

Fig. 6 — Three program store system — CC, PS, and bus out of service.

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

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

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

3.3 *Duplication and Switching of Peripheral Units*

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

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

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

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

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

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

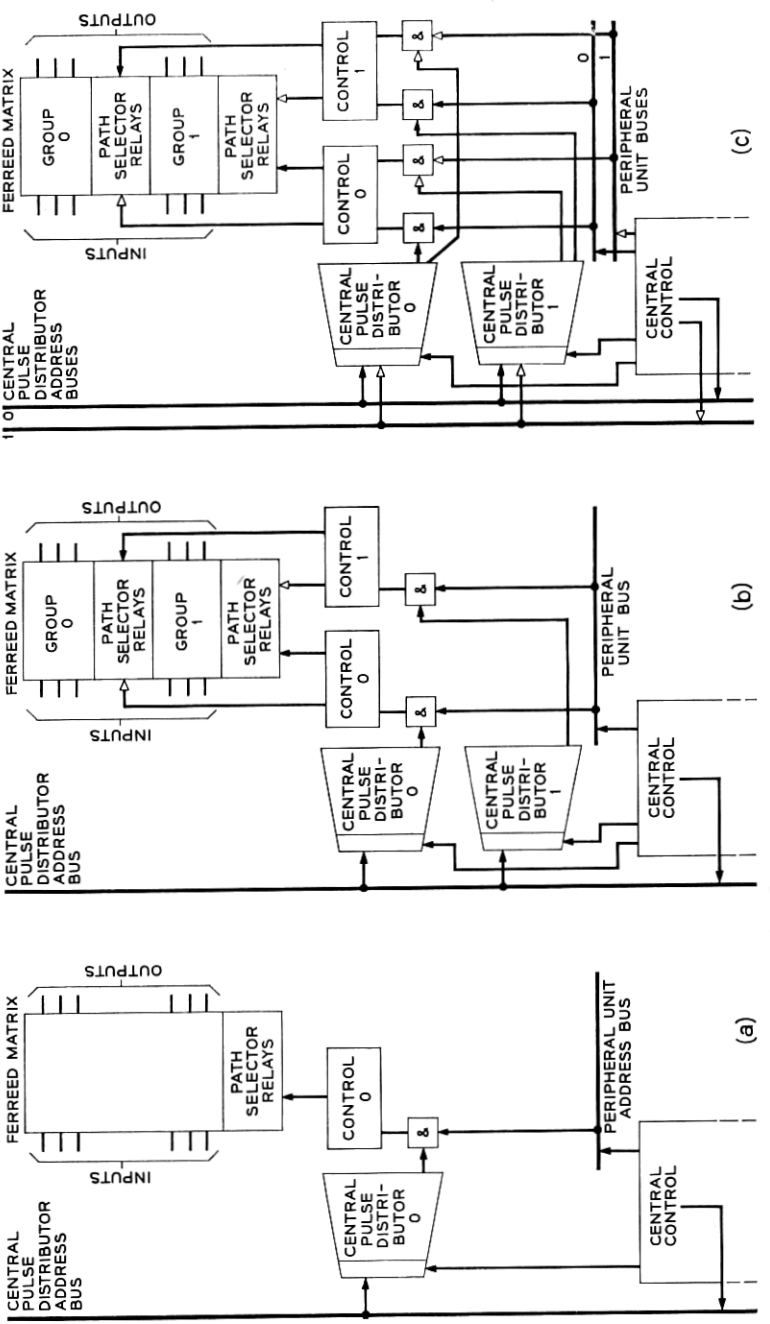

Fig. 7 — Network duplication.

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

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

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

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

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

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

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

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

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

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

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

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

IV. MAINTENANCE CIRCUITS

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

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

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

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

4.1 *Central Control*[2,3]

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

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

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

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

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

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

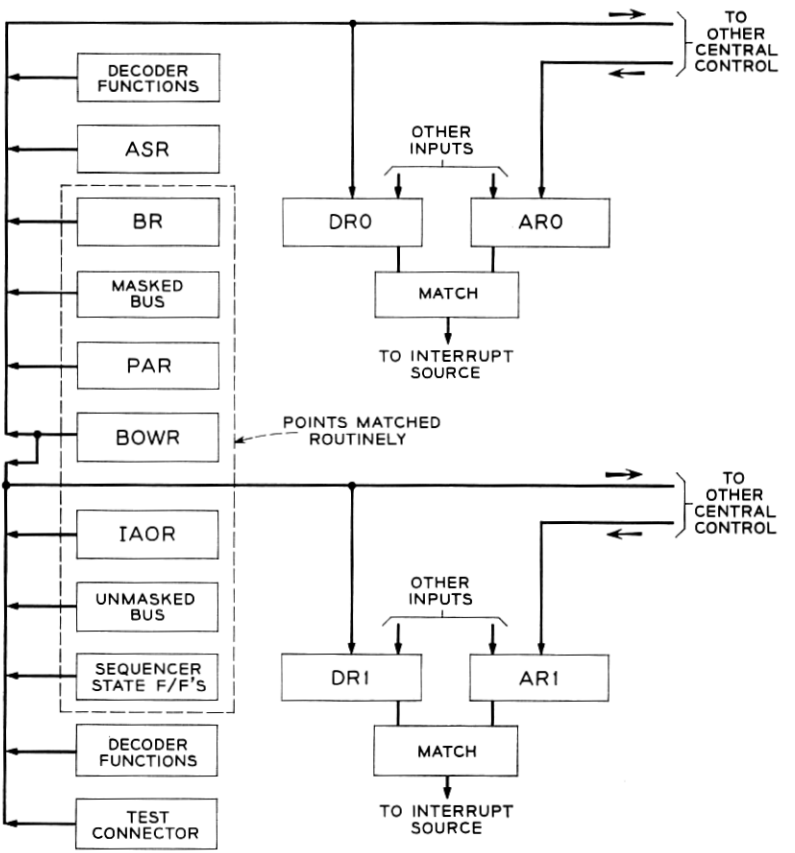

Fig. 8 — Central control matching access.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

4.2 *Program Store*[10]

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

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

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

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

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

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

4.3 *Call Store*[12]

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

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

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

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

4.4 *Central Pulse Distributor*[7]

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

4.6 *Scanner*[7]

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

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

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

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

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

(a) an office alarm system,

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

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

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

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

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

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

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

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

V. MAINTENANCE PROGRAMS

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

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

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

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

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

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

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

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

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

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

(3) simple — for easy design and understanding

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

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

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

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

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

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

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

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

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

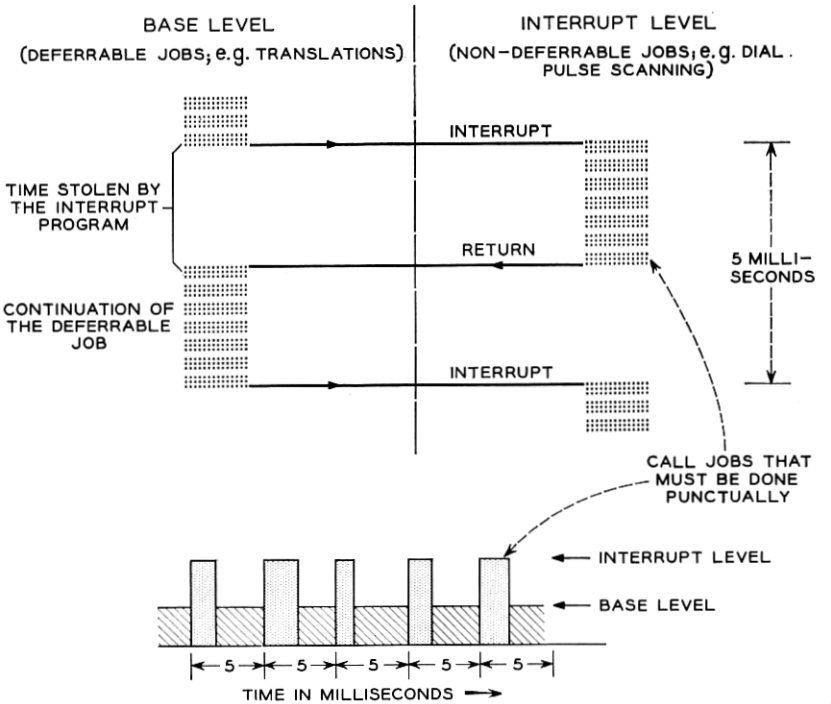

Fig. 9 — Call program operation (1).

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

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

5.2 *Fault-Recognition Programs*

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

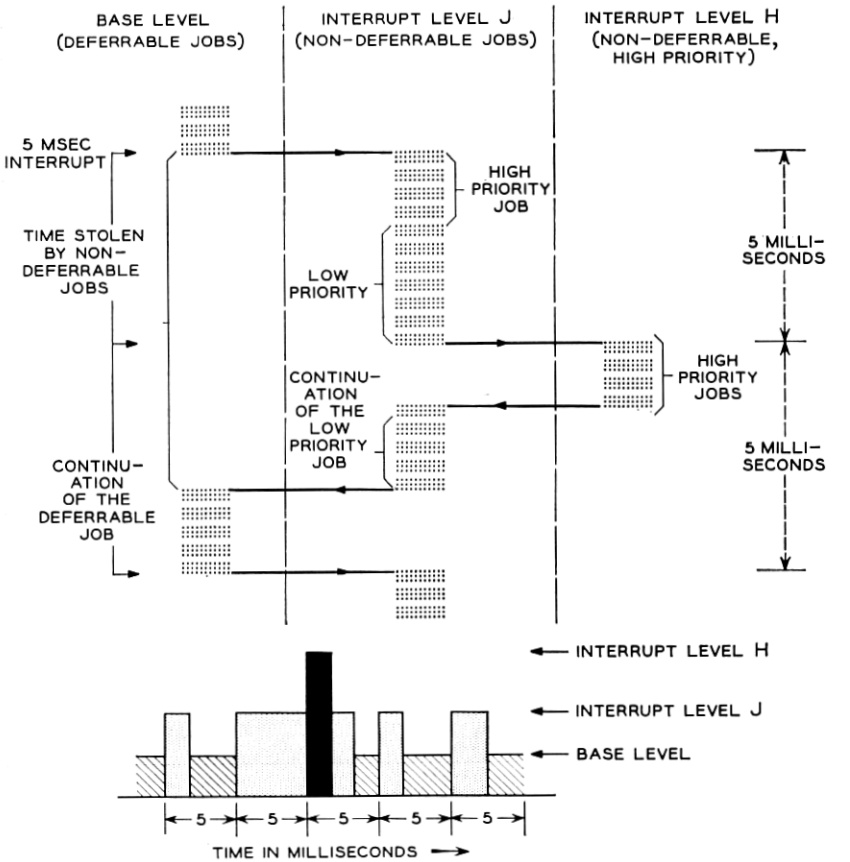

Fig. 10 — Call program operation (2).

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

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

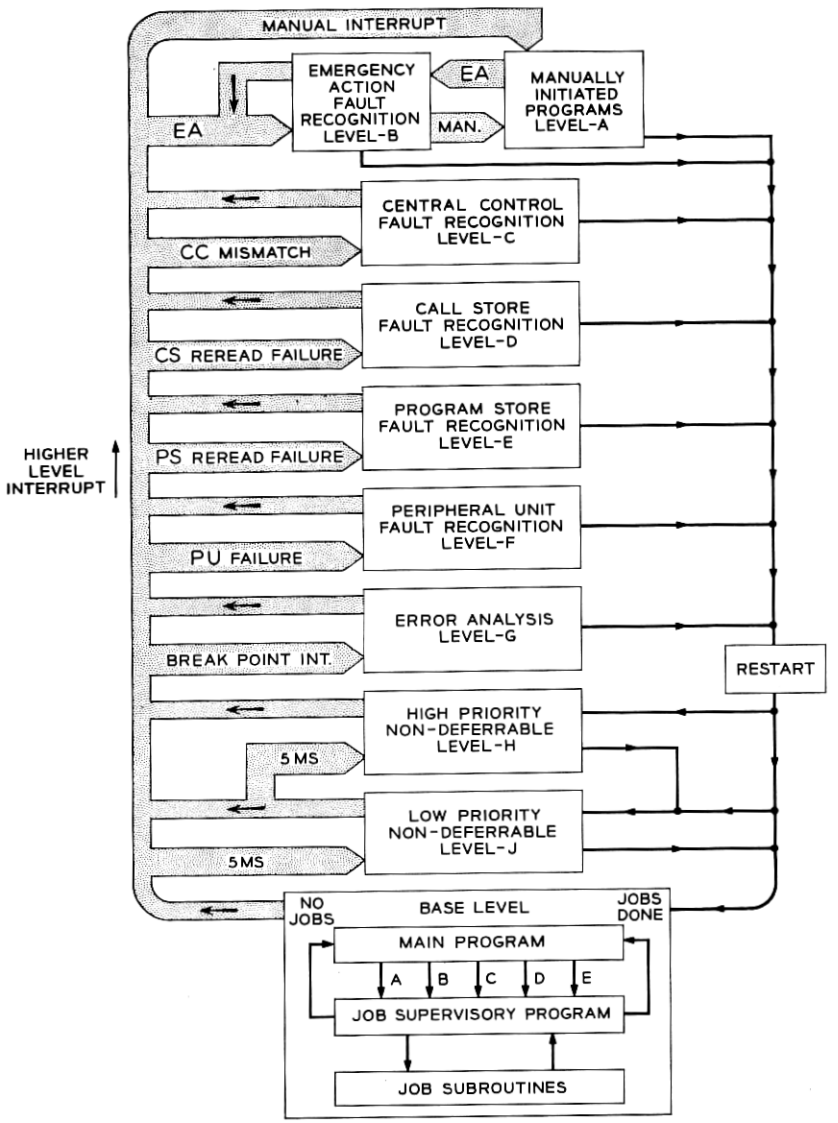

Fig. 11 — Interrupt levels and priorities.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

5.3 *Diagnostic Programs*

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

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

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

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

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

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

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

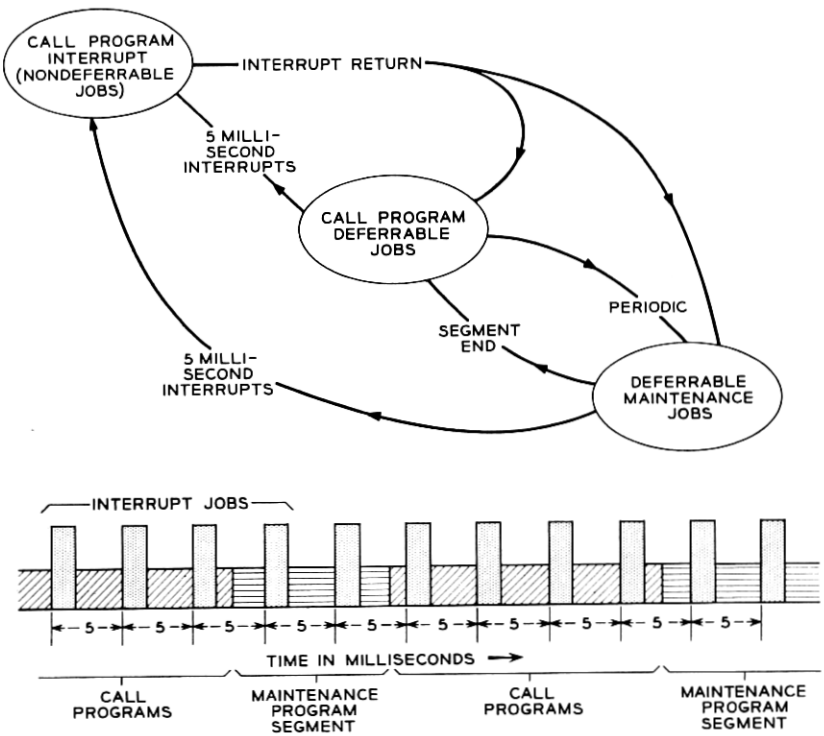

Fig. 12 — Maintenance-call processing interleaving.

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

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

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

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

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

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

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

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

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

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

5.4 *Exercise Programs*

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

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

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

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

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

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

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

5.5 *Implementation of Fault-Recognition Programs*

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

5.5.1 *Central Control Fault Recognition*

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

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

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

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

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

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

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

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

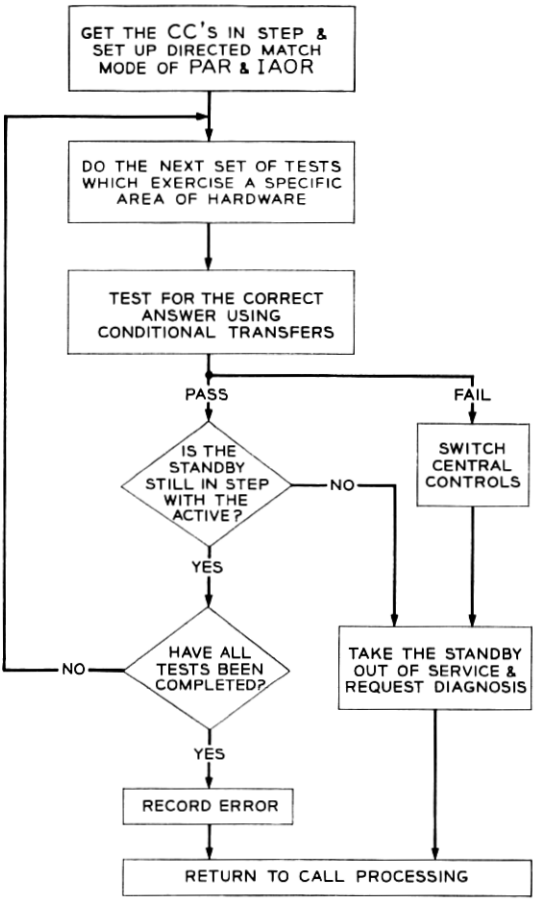

Fig. 13 — Central control fault-recognition test mode.

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

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

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

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

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

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

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

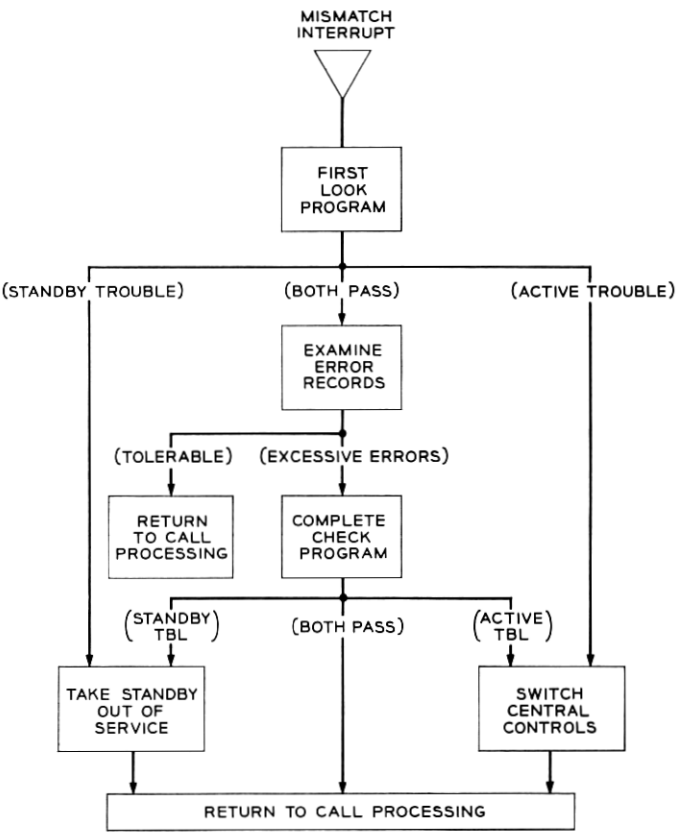

Fig. 14 — Central control fault-recognition program.

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

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

5.5.2 *Program Store Fault Recognition*

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

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

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

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

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

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

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

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

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

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

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

5.5.3 *Call Store Fault Recognition*

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

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

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

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

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

5.6 *Emergency-Action Functions*

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

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

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

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

Since the emergency-action facility is provided as a back-up to the fault-recognition programs, this facility must be made as independent of normal central processor operation as possible. For example, the emergency-action facility is not dependent on the system clock. The emergency-action hardware generates its own pulses that sequence the emergency-action circuit through its actions.

The initial activation of the emergency-action circuit begins a cycle. Subsequent input signals produce state changes advancing a four-stage state counter. This counter records the successive enable signals within a cycle and directs the specific actions which are to be carried out for each timeout.

As the emergency-action sequencer advances through its various states it switches through all possible combinations of central controls, base program stores, and buses. The configurations established by the emergency-action sequencer during each state are summarized in Table III. Note that during the initial state 0000 and during states 0111, 1000, and 1111 no switching is performed and only an interrupt is generated.

TABLE III — CONFIGURATIONS ESTABLISHED BY EMERGENCY-ACTION SWITCHING

Status of Units after a Switch Performed by the Indicated State

EA State	CC0	CC1	PS0	PS1	Bus0	Bus1	Other Stores
X000	U	U	U	U	U	U	U
X001	C	C	U	U	U	U	U
X010	U	U	U	U	C	C	U
X011	U	U	A	S	A	S	T
X100	U	U	A	S	S	A	T
X101	U	U	S	A	S	A	T
X110	U	U	S	A	A	S	T
X111	U	U	S	A	A	S	T

X: don't care
A: this unit is switched active
S: this unit is switched standby
U: the status of this unit is unchanged
C: the status of this unit is complemented
T: this unit is marked in trouble.

With each activation of the emergency-action sequencer, a B-level interrupt is initiated. The configuration changes and the actions listed above are completed before the interrupt program is started. The maze program, if successful, is followed by an emergency-action recovery program which is designed to recover the call processing ability of the system.

The emergency-action interrupt program is divided into five phases. These are as follows:

(1) basic sanity maze program,
(2) operational checks of the central pulse distributor,
(3) operational checks of the call stores,
(4) bootstrap recovery of program stores, and
(5) emergency-action evaluation.

The basic sanity testing is not designed to isolate trouble, but instead to test the ability of the active central processor to process program instructions properly. The program with the aid of a sanity timer determines if the active central processor is operable (sane).

The check of the central pulse distributor is made to ascertain that the central pulse distributor can be used successfully in the program store recovery program. The operational check of the call stores is made for similar reasons. A bootstrap recovery of the program stores is carried out only after the emergency-action sequencer has advanced to or beyond the state where the program store complex was forced into a special configuration.

The emergency-action evaluation program determines the rate at which emergency actions are occurring and attempts to determine the cause of any recurring cycles. It also requests subsequent maintenance program actions to isolate faulty units and to return the fault-free units to service.

The emergency-action circuit is activated initially by one of the following conditions (see Fig. 15):

(a) a clock check circuit detects trouble in the microsecond clock,
(b) a sequencer check circuit finds a locked up central control sequencing circuit, or
(c) a "real-time" check determines that the program is out of step with a time reference provided by the emergency action.

The real-time check is a test of the normalcy of call processing by both circuits and programs. This check assures that call processing has not been limited by the exclusion of some program functions. In addition, this check compares the passage of "real time" as counted by the program to that counted by the hardware. A record of time is kept by the program on the basis of the 5-millisecond interrupts generated by a

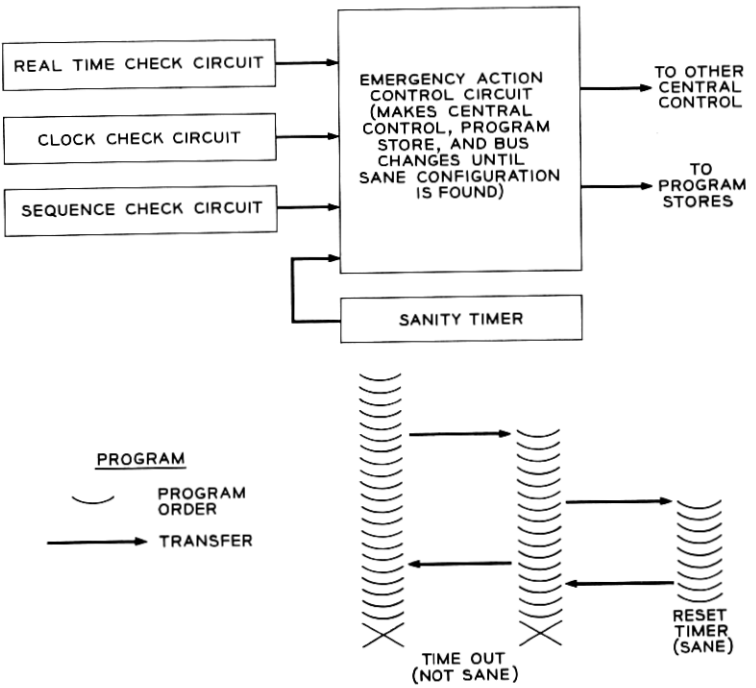

Fig. 15 — Emergency action.

5-millisecond clock. The same clock circuit supplies a signal once every 10 milliseconds to the emergency action counter. This counter, which can count up to 640 milliseconds (see Fig. 16), has two other inputs (enable and reset) which are program controlled. If the program progresses properly through its various tasks and if it stays in step with the emergency action counter, it will first reach a point (T2) where it must generate an enable signal and later a point (T3) where it must generate a reset signal. If the program goes astray and fails to generate either or both signals, the emergency-action counter will time out and activate the emergency-action sequencer within 640 msec.

During the 640-millisecond period, call programs operate on base level and on interrupt levels H and J. The proper recurrence of the interrupt level is reflected in the program count of time and in proper operation of enable and reset inputs to the emergency-action real-time counter.

Associated with each priority level (A through E) of the base level is a reserved location within the call store which is set to the "1" state after the priority level has been visited and each of the jobs at that

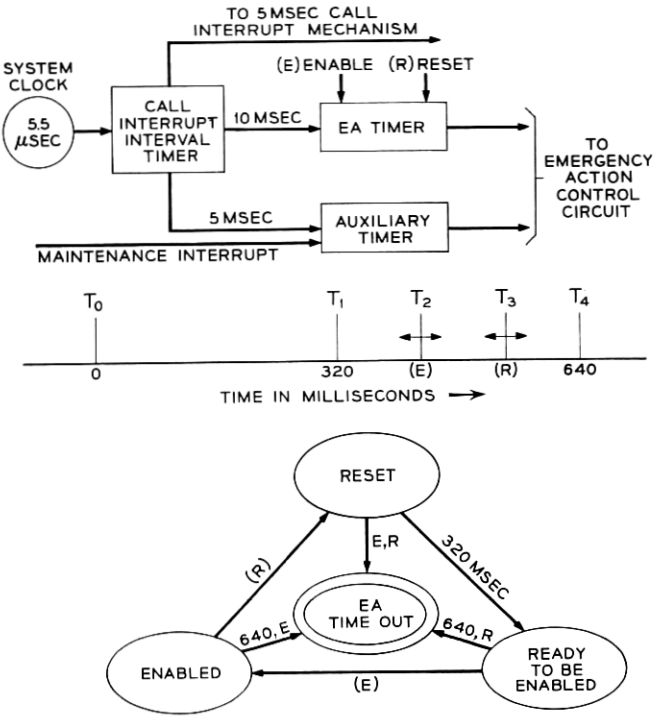

Fig. 16 — Emergency-action timers.

level has been performed. The real-time check program examines these call store locations to determine whether the base-level work is proceeding normally. If these locations indicate the failure to complete the normal amount of work, the real-time control program initiates an overload control. The over-load control slows down the acceptance of new work to allow the assumed backlog of work to be completed. If the overload control fails to recover the normal visits to the base-level jobs, the real-time check assumes that the failure to cycle through base level is due to mutilated data in call store. A reinitialization of vital call store constants is made. If the trouble condition continues, further reinitialization of data is carried out.

5.7 *Implementation of Diagnostic Programs*

5.7.1 *Central Control Diagnosis*

The object of the central control diagnosis is to isolate faults in the standby central control to a small number of replaceable circuit packs.

This is accomplished by performing a series of tests on the standby central control, recording the results of these tests and printing them out on the teletypewriter. The faulty packages are identified by locating the printout in the maintenance dictionary.

In the fault-recognition program, each central control in effect tests itself by performing logical operations and conditional transfers. This technique was employed since it was not known which central control, if either, would be faulty. However, in the diagnosis it is known that the standby central control is faulty and that the active central control is fault-free (assuming only one central control is faulty). Knowing this, a different (more reliable) testing technique can be employed. In this environment the active system can be used to test the standby central control.

To perform a test, one wishes to apply certain inputs to the circuit under test, observe the outputs, compare them with expected outputs, and record results indicating which tests passed and which failed. The facts that the central controls are complete duplicates, are capable of running in synchronism, and that we have a way of comparing their operations (the match circuits) can be used to advantage in testing the standby central control.

The principal testing technique employed in the central control diagnosis is to force the two central controls to execute the same test program and to compare certain critical points using the match circuits. A test result or results are then recorded for each match operation. A mismatch indicates a failure for the operation being tested, and a match indicates that the test passed. In this testing mode, the mismatch sampling order is used to sample the desired information at the desired time. Thus the most frequent type of test in the central control diagnosis is of the following type:

(1) the central controls are forced into step;

(2) a test program is executed which completely exercises a specific hardware function of the central control;

(3) at a number of times during this program, the output (or the nearest available output) of the circuit being tested is sampled using the match circuits and the mismatch sampling order;*

(4) as the sampled matches are executed, the results of these matches are recorded; and

(5) at various points, the central controls are forced back into step to ensure meaningful match results.

* This can be done without depending upon the standby central control to execute the mismatch sampling order properly by operating the standby central control in the directed match mode and the active central control in the mismatch sampling mode.

The programs which actually test the standby central control are divided into subroutines labeled "test phases." A test phase exercises a specific hardware area of central control. For the central control there are 20 test phases. The test phases, defined by the circuits which they test, are:

(1) power and clock circuits,
(2) start, stop and control word reception,
(3) alternate routes to hardware in phase 2,
(4) buffer order word register — error detection and correction,
(5) index adder and index adder registers,
(6) index registers and bus circuits,
(7) decoders,
(8) homogeneity and transfer logic,
(9) program address incrementing,
(10) arithmetic and logic circuitry of the accumulator register,
(11) memory operations,
(12) sequencing circuits,
(13) buffer bus registers,
(14) parity generators and checkers,
(15) maintenance circuits (matchers, etc.),
(16) call store address circuits,
(17) program store address circuits,
(18) scanner answer circuits,
(19) enable control, and
(20) peripheral address circuits.

The phases are performed in the order indicated, except that at certain points the diagnosis is terminated if failures have been detected. The first phase consists of scanner operations which check the power state of the standby central control to determine whether all voltage regulators are functioning normally and whether the microsecond clock appears normal. The above circuits have dc trouble detectors associated with them which are in turn connected to scan points. If a failure is detected in phase 1 no further tests are performed. Phase 2 tests the ability of the standby central control to receive control words from the active central control. The control write facility is used throughout the diagnosis to set up the match control circuits of the standby central control and to get the two central controls in step by control writing into the program address register (PAR). This phase is performed using both call store buses if they are available. If a failure is detected in phase 2, phase 3 is performed and the diagnosis is terminated, with the remaining test results being recorded as all-tests-pass.

Phase 4 tests the buffer order word register (BOWR) and the error

detection circuits. This phase is performed from both program store answer buses if they are available. If a failure is detected, the diagnosis is terminated.

Phase 5 tests the index adder circuits, and phase 6 tests the index registers and bus logic circuitry. If a failure is detected in either of these phases, phase 7 (which tests the decoders) is performed, and the diagnosis is terminated. These phases test registers which must be used in testing all circuits which follow; the diagnosis is terminated to avoid inconsistencies created by initial conditions described earlier.

If the above phases find no trouble, phases 8 through 15 are performed. These phases test all internal circuits which are not associated with external buses. If any failures are detected in these phases, the diagnosis is terminated after phase 15.

If the fault has not been located by any of the preceding phases, phases 16 through 20 are executed. These phases are again concerned with circuits associated with external buses. Each of these phases is performed from both of the duplicate buses associated with the circuits being tested, unless one is not available. If a failure is detected in a phase, the diagnosis is terminated at the completion of that phase.

It is estimated that the central control diagnostic program requires approximately 6,000 program words, and generates approximately 2,000 bits of test results. The 2,000 bits of information will normally not be printed out on the teletypewriter. Instead, a number generation program (to be described later) will operate upon this data and print out a much smaller, easier to handle, number.

5.7.2 *Program Store and Call Store Diagnostic Programs*

The program store and call store diagnostic programs are similar in function and in design. Both of these programs attempt to locate a fault within a store which has previously been found faulty. This objective is accomplished by performing a series of exercises on the faulty store and recording the results of these exercises.

Since it is necessary to be able to diagnose a store with one bus out of service and with one central control out of service, the exercises are performed using the active central control and active bus. Most of the testing is performed using special maintenance orders provided for this purpose. Using the maintenance orders, only the store specified by the order responds, and it sends its answers back on the bus from which it receives, regardless of the state of the answer routing flip-flops. When the faulty store is being tested it is normally connected to receive from the active bus but to send on neither bus for normal orders. Thus, for the

case of program stores, the active central control receives its instructions from a fault-free store connected to the active bus; yet it can also interrogate the faulty store by reading data words using the maintenance orders. Similarly, for the call stores the active central control uses a good set of call stores connected to the active bus for storing and reading temporary information, and it gains access to the faulty store for test purposes by using the maintenance orders.

As in the central control tests, the store tests are divided into meaningful blocks called "phases" which test various blocks of the store circuits. If both buses are available, all of these phases are performed from both buses.

5.8 *Routine Exercise Programs*

The routine exercise programs are basically of two types: those which check to see that various memory items (both call store and flip-flop) are updated, and those which exercise hardware which is not used in normal system operation.

The automatic programs have periodic schedules. There are three classes of automatic programs, where the class is determined by the scheduling technique. Class I programs are rigorously scheduled at a relatively high frequency. These programs are entered from the high-priority main program regardless of the office traffic. They must of necessity be fairly short, since they are performed religiously, even during the busy hour. An example of a program assigned to this class is that program which interrogates and resets the store error counters. This must be done on a strict schedule to ensure meaningful interpretation of the contents of the counters.

Class II programs are also rigorously scheduled, but at a much lower frequency. Programs which must be performed every hour, or at some specific time during the day, are assigned to this class. An example of a program in this class is a program for testing the emergency action circuit. This test should be performed only when traffic is low and consequently would be scheduled daily at 2 a.m. or some other nonbusy hour.

Class III routine exercise programs are the lowest-priority programs in the system. These exercises are performed in system spare time when no other jobs are waiting. These exercises are ordered in a circular list, so that when one is completed, the next one in the list is initiated. Most of the routine exercises are assigned to this class. Some examples of routine exercises in this class are:

(1) a program to exercise the match circuits of the central control to

insure that they are capable of detecting a mismatch and that they will operate correctly in all available modes

(2) a program to exercise the error detection and correction circuits of the central control

(3) programs to check that the stores will respond properly to all maintenance orders

(4) programs to change the interconnection configurations of duplicate units

(5) programs to verify and update status words in temporary memory to insure that they agree with the actual system status.

The demand exercise programs are initiated upon request. The request can be initiated by the teletypewriter or by some other program. All automatic programs can be requested as demand programs. Some examples of demand programs are:

(1) a program to remove a unit from service

(2) a program to restore a unit to service

(3) a program to print out the status of a particular unit

(4) a program to print out the contents of a specified call store location, etc., and

(5) programs that audit the network memory.

All of the maintenance programs are normally executed by the active data processor or by both the active and standby systems. Tools in the form of demand exercise programs will also be provided to allow the execution of almost all of these programs on a repeated basis by the standby system. The need for this ability may arise if a marginal trouble develops which is not detected by the diagnostic programs. For troubles of this type it may be desirable to execute some maintenance programs continuously in the standby system while maintenance personnel make observations with an oscilloscope or some other manual tool.

Circuit tools are available, and program tools will be provided, for this purpose. To run the standby central processor independently of the active central processor (i.e., off-line), all that is required is to interconnect the central controls, call stores, and program stores so they operate as two independent systems. As described in Section III, with complete duplication this ability exists. The start-stop control, control write facility, and the breakpoint match mode also provide circuit tools which allow the active system to start the standby system in any program and to stop it at any desired address.

For example, if it is desired to execute some program continuously, starting at address A and ending at address B, this could be accomplished in the following manner. An input message would request that the

standby system execute the program from address A to address B. This demand exercise would modify call store and program store configurations to set up two independent systems. The active central control would stop the standby central control and control write the start address A into the standby program address register. It would then set up the breakpoint match mode to monitor address B with the interrupt and stop standby options specified. Next, it would start the standby central control and return to call processing.

The standby system would then begin executing program at address A while the active system ran its normal call programs. When the standby system reached address B, the active matchers would be alerted, stop the standby system and interrupt the active system with a G-level interrupt. This interrupt program could restart the standby system at A and return to call processing, etc.

VI. MAN-MACHINE RELATIONSHIP

6.1 *Reaction to Trouble*

Let us now review briefly the facilities through which the maintenance man normally communicates with the system. Most of these facilities are included in the four frames which are called the "master control center." When a failure occurs in the system it is most frequently detected by circuits. Programs are immediately brought in to remove the faulty unit from the system and to establish a working system configuration. At a later time, diagnostic programs are run on the faulty unit. For example, assume that a scanner controller has failed, has been switched out of service, and has been diagnosed. At this point, the office alarm system will sound an audible minor alarm and light lamps that direct the maintenance man to the master control center. At the master control center a maintenance teletypewriter prints out the identity of the faulty controller and also a code which the maintenance man, with the aid of a dictionary, translates to the location and identity of the faulty package. The man, using this information, will locate the frame containing the faulty controller. At the frame a red trouble lamp will indicate which controller within the frame is out of service. By pushbutton he will remove the power from the controller and replace the faulty package. With another pushbutton he reapplies power. This action signals the system to start a diagnosis on the controller. If the diagnosis passes, the system will extinguish the trouble light at the controller as a signal that the controller is fault-free. Fig. 17 illustrates this flow of actions.

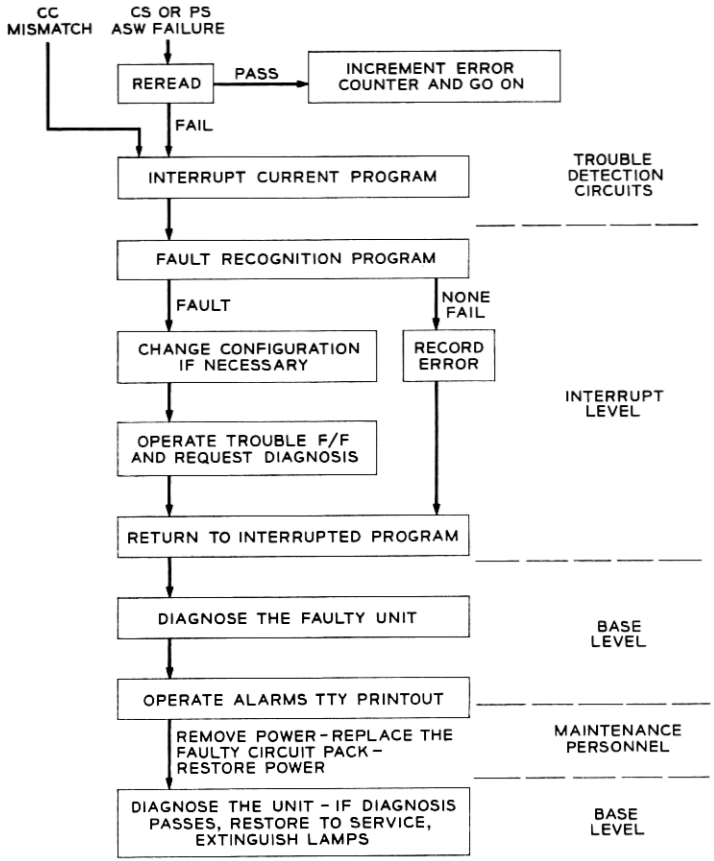

Fig. 17 — Maintenance reaction to trouble.

6.2 *Maintenance Dictionary Production*

The maintenance dictionaries are used for translating diagnostic program output, as printed out on maintenance teletypewriters, to specific package locations. Because of the complexity of No. 1 ESS and because of the number of diagnostic tests, it is considered infeasible to produce complete and accurate dictionaries by logical reasoning alone, i.e., to predict the reaction of each test to every possible fault. The method that will be used to produce dictionaries is: each plug-in circuit pack will be replaced by a fault simulator which will introduce every possible type of single fault on the replaced package one at a time and

then record the system reaction on a high-speed output tape.* From this record, the dictionaries can be produced by sorting and ordering the test results using auxiliary data processing equipment. In addition to the relatively complete and accurate dictionaries which will result, this method also provides early feedback for evaluation of the maintenance plan — feedback which otherwise would take years to collect from operational experience. At the time diagnostic data for dictionaries are collected, additional data will be collected on key program reactions and decisions. The kinds of data which can be collected are: the system configuration changes, the length of time required by the various phases of the program, program interactions, and interrupts generated. These data can be evaluated to find any weaknesses of the maintenance programs and also to find redundant or troublesome programs.

The data generated by diagnostic programs consist of a binary word of n bits. Each bit may represent the result of a test (pass or fail), or it may take several bits of information to represent the result of a test. A simple way of organizing the dictionaries is to convert these binary words to decimal numbers, place them in numerical order, and list next to each number the faulty packages that generated that number. This is essentially what was done for the Morris ESS central control dictionary.[16] This dictionary, although somewhat bulky (it consisted of some 1200 double-thickness pages), was quite satisfactory when the printout in the field exactly matched a dictionary entry. When faults were introduced into the Morris central control, it was found that approximately 60 per cent produced test results that exactly matched the dictionary entries. Another 16 per cent of the faults produced entries that could be located by using some relatively awkward interpretative rules. There are many factors that can cause the diagnostic results for a given fault to differ from one machine to another or from one occurrence to another. Test results may depend on the memory state of the machine at the time the trouble occurs,† and variations in component and voltage values in two systems may be sufficient to produce different results. It is expected that a similar situation may exist in No. 1 ESS, although to a lesser degree. Consequently, it is desirable to present diagnostic data in the dictionaries in a form such that if an exact match cannot be found, the dictionary entry (or entries) most nearly resembling the diagnostic printout can be easily located.

* This process has to be carried out only once unless changes are made in the maintenance program or in the circuits.

† As stated previously, in designing diagnostic tests extensive efforts are being made to avoid this problem.

The diagnostic data can be given a geometric interpretation by considering the given test result as a point in a multidimensional space where the binary word that represents the test results gives the coordinates of the point. The method that will be used in No. 1 ESS to generate diagnostic dictionaries takes an advantage of the fact that (1) of the many possible test patterns relatively few will occur* (the space is sparsely populated) and (2) that the significance of individual tests varies. The dictionary entries will express the weighted (according to the significance of the tests) distance of each point from preselected reference points, rather than specifying the exact coordinate of the test result. This method will yield (1) a more compact dictionary, since the distances can be expressed more concisely than the actual coordinates, and (2) a dictionary in which similar patterns can be easily located when an exact match cannot be found. The methods of diagnostic data collection and dictionary generation will be subjects of a future article.

VII. CONCLUSIONS

The design of maintenance programs has been based on the logical analysis of circuit diagrams rather than on actual experience with the system. Because of this and the complexity of the system, it is expected that the first design will not completely meet all of the objectives. The data produced for the dictionaries will provide early feedback on the strengths and weaknesses of the maintenance plan. This feedback should enable us to evolve a design which will meet all of the design objectives. It is also hoped that with adequate feedback the 47,000-word maintenance program can be substantially reduced.

VIII. ACKNOWLEDGMENTS

To design a central office which is both reliable and maintainable has required the complete cooperation and awareness of everyone connected with the design of No. 1 ESS. The authors would like to pay special tribute to their coworkers in the No. 1 ESS maintenance planning department whose work is summarized herein.

REFERENCES

1. Keister, W., Ketchledge, R. W., and Vaughan, H. E., No. 1 ESS System Organization and Objectives, B.S.T.J., this issue, p. 1831.

* For example, in the No. 1 ESS central control approximately 2×10^5 faults will be introduced; the diagnostic program consists of approximately 2000 tests. Thus, assuming all tests are independent, there are 2^{2000} possible test patterns, of which only 2×10^5 will occur.

2. Harr, J. A., Taylor, F. F., and Ulrich, W., Organization of No. 1 ESS Central Processor, B.S.T.J., this issue, p. 1845.
3. Cagle, W. B., Menne, R. S., Skinner, R. S., Staehler, R. E., and Underwood, M. D., No. 1 ESS Logic Circuits and Their Application to the Design of the Central Control, B.S.T.J., this issue, p. 2055.
4. Ferguson, J. G., Grutzner, W. E., Koehler, D. C., Skinner, R. S., Skubiak, M. T., and Wetherell, D. H., No. 1 ESS Apparatus and Equipment, B.S.T.J., this issue, Part 2.
5. Chevalier, J. G., and Eisenhart, R. K., No. 1 ESS Circuit Packs and Connectors, B.S.T.J., this issue, Part 2.
6. Danielson, D., Dunlap, K. S., and Hofmann, H. R., No. 1 ESS Switching Network Frames and Circuits, B.S.T.J., this issue, Part 2.
7. Freimanis, L., Guercio, A. M., and May, H. F., No. 1 ESS Scanner, Signal Distributor, and Central Pulse Distributor, B.S.T.J., this issue, Part 2.
8. Dougherty, H. J., Raag, H., Ridinger, P. G., and Stockert, A. A., No. 1 ESS Master Control Center, B.S.T.J., this issue, Part 2.
9. Connell, J. B., Hussey, L. W., and Ketchledge, R. W., No. 1 ESS Bus System, B.S.T.J., this issue, p. 2021.
10. Ault, C. F., Gallaher, L. E., Greenwood, T. S., and Koehler, D. C., No. 1 ESS Program Store, B.S.T.J., this issue, p. 2097.
11. Feiner, A., and Hayward, W. S., No. 1 ESS Switching Network Plan, B.S.T.J., this issue, Part 2.
12. Genke, R. M., Harding, P. A., and Staehler, R. E., No. 1 ESS Program Store — A 0.2-Megabit Ferrite Sheet Memory, B.S.T.J., this issue, p. 2147.
13. Biddulph, R., Budlong, A. H., Casterline, R. C., Funk, D. L., and Goeller, L. F., Line, Trunk, Junctor and Service Circuits for No. 1 ESS, B.S.T.J., this issue, Part 2.
14. Harr, J. H., Hoover, Mrs. E. S., and Smith, R. B., Organization of the No. 1 ESS Stored Program, B.S.T.J., this issue, 1923.
15. Carbaugh, D. H., Drew, G. G., Ghiron, H., and Hoover, Mrs. E. S., No. 1 ESS Call Processing, B.S.T.J., this issue, Part 2.
16. Tsiang, S. H., and Ulrich, W., Automatic Trouble Diagnosis of Complex Logic Circuits, B.S.T.J., **41,** July, 1962, p. 1177.